



CAMAL ŞAHVERDİYEV  
**FreeBSD**  
PRAKTİK HƏLLƏR

CAMAL ŞAHVERDİYEV

# FreeBSD

PRAKTİK HƏLLƏR

Müəllif: Camal Şahverdiyev

Layihə koordinatoru: Samirə Qafarova

Korrektor: Rəna Kərimova

Art dizayner: Murad Əsədov

Dizayner: İman Hüseynov

Oxucuya müraciət:

*Bu sahə üzrə Azərbaycan dilində kitabı ilk dəfə nəşr olunduğundan istifadə edilən termin və sözlər məlumatın daha anlaşıla bilən olması üçün tətbiq edilmişdir.*

*Kitabın daxilində səhv aşkar etsəniz, xahiş edirik, sərt şəkildə tənqid etməyəsiniz. Yanlış söz və ya sintaksis səhvini gördükünüz halda, [bookcorrector@gmail.com](mailto:bookcorrector@gmail.com) mail ünvanınıza yazmağınız xahiş olunur. Bununla növbəti nəşrlərimizi daha mükəmməl etməyə yardımçı olarsınız.*

Bütün müəllif hüquqları qorunur. Kitabın daxilində eks olunan məlumatların yayılması, çapı, surətinin çıxarılması və ya digər bir şəkildə istifadə olunması yalnız müəllifdən razılıq alındıqdan sonra mümkündür.

Məlumat qeyd olunan məqamlar nəzərə alınmadan istfadə edilərsə, müvafiq qanunvericilik üzrə tədbirlər tətbiq olunacaq.

ISBN: 978-9952-8280-7-8

*"ATL Group" şirkətlər qrupunun rəhbəri Rəşad Mirzəyevə kitabıın nəşr edilməsində  
göstərdiyi dəstəyə görə təşəkkür edirəm.*

*Qafqaz Universiteti CEDAWI-nin rəhbəri Abzətdin Adamova və sözü gedən  
universitetlə əlaqələrin yaradılmasında və kitabıın ərsəyə gəlməsində  
böyük əziyyəti olan Samirə Qafarovaya təşəkkürümü bildirirəm.*

*Valideynlərimə, həyat yoldaşımı və sevimli qızıma bütün bu müddət ərzində  
mənə səbrlə dözdüklərinə, onlara vaxt ayıra bilməməyimə rəğmən daim yanımda olub,  
mənəvi dəstək olduqlarına görə dərin təşəkkürümü bildirirəm.*



# Ön söz

Səmimi olaraq, heç bir zaman kitab yazmaq haqqında düşünməmişdim və təsəvvür edə bilməzdəm ki, nə zamansa kitab yazacam. Tərcümə və qeydlərimin ilkin olaraq hazırlanma səbəbi peşəkar bilgilərimin dərinləşdirilməsi, işimin keyfiyyətinin artırılması idi. İnforsasiya texnologiyaları sahəsində bir müddət işlədikdən sonra öncə icra etmiş olduğum tapşırığı yenidən yerinə yetirmək göstərişi aldım. Bu işi öncə icra etdiyimə görə bilirdim ki, tam işlək bir nəticəni rahat əldə edə biləcəm. Lakin tələb edilən program təminatının ətraflı nəzəriyyəsi və qurulma sintaksisini tam xatırlamırdım. Program təminatını yenidən qurmaq üçün biliklərimi təkrarən yeniləməyə məcbur idim. Həmin biliklərin bərpası üçün ilk dəfə bu sahə üzrə bilgilərin toplanmasına sərf etdiyim qədər vaxt ayırmalı oldum. İşimi bitirdikdən sonra qəti qərara aldım ki, heç olmazsa, işləyən program təminatları üçün özümdə bir sintaksis "konspekti" saxlayım. Qeydlərim gələcəkdə bu cür hallarda vaxtımı qənaət etməyə imkan verəcəkdi. Yalnız bir neçə müddətdən sonra nəticəyə gəldim ki, hətta sintaksis konspekti real iş müddətində çıxan problemləri aradan qaldırmaq üçün anında köməyimə çatdırı.

Növbəti qərarım, hər dəfə hansısa bir tapşırığın reallığa çevrilməsində gördüyü işi tamamilə ardıcıl şəkildə qeydə almaq oldu. Artıq müəyyən vaxtdan sonra hansısa bir problem və ya sual ilə bağlı sorğu aldıqda, onun qurulmasını və nəzəriyyəsini internetdən yox, dəqiq olaraq tətbiq etdiyim tərcümələrimdən oxuyurdum. Anladım ki, bu, istənilən işin icra müddətini reallıqda çox azaldır, eyni zamanda bilgilərimi dərinləşdirməyə məcbur edir. Hər dəfə yazdıqlarımı oxuduğda çatışmamazlıqları təyin edir, yenilikləri əlavə edirdim və beləliklə, növbəti dəfə üçün sənəd tam yenilənmiş olurdu.

Hər bir tapşırığın reallığa çevrilməsində internetdə günlərlə axtardığım cavabı öz iş qeydlərimin içərisindən bir neçə dəqiqədə əldə edə bilirəm. Bu andan sonra, hətta oxuduğum və sinaqdan keçirdiyim hər bir kitabın nəticələrini anladığım şəkildə yazmağa başladım.

Ümid edirəm ki, bilik və iş təcrübəmdən əldə etdiyim bilgiləri kitab şəklində bölüşməklə həmkarlarına yardım etmiş oluram.

**Camal Şahverdiyev**

# Kitabdan istifadə qaydaları

Aşağıdakı açıqlamalar kitabı mütaliəsində oxucuya yardımçı olacaq:

Əsas başlıq

- **Bold və böyük hərfələr**

Əsas başlıq 1-ci dərəcəli alt başlıq

- **Arxa fon qara, şrift ağ**

Əsas başlıq 2-ci dərəcəli alt başlıq

- Altдан xətt

Əmrlər bold qeyd olunub. Əgər hansıa faylin içərisində olan sintaksisdən danışılırsa, öncədən faylin adı və tərkibinə əlavə ediləcək sətirlər bildirilir.

Qeydlər qırmızı çərçivəyə alınmışdır.

# - Isətinlən UNIX/Linux əməliyyat sistemində faylların içinde şərh üçün istifadə edilir.  
Simvoldan sonrakı sözlər oxunmur.

/\* **şərh** \*/ - DNS BİND-da və PHP programlaşdırma dilində yazılmış kodlarda göstərilən simvolların daxilində olan istenilən yazı şərhdir.

// - DNS BİND-da və PHP programlaşdırma dilində yazılmış kodlarda göstərilən simvollardan sonra olan ixtiyari yazı şərhdir.

; - DNS BIND-da sətirin sonu deməkdir.

Qeyd: "Yenidənyüklənmə" ilk dəfə olaraq, informasiya texnologiyaları sahəsinə aid "**reboot**" termininin Azərbaycan dilində qarşılığı kimi istifadə olunmuşdur.

Oxucu tərəfindən kitabı başa düşülməsi üçün tələb edilən baza biliklər:

1. TCP/IP-nin əsasları
2. Subnet mask və Gateway
3. TCP və UDP işləmə prinsipləri
4. IP ünvanlarının aralıqları (daxili, dünya, localhost, multicast və APIPA IP aralıqları)
5. ARP və MAC ünvan haqqında məlumat
6. Server, HDD, RAM və CPU
7. File system və RAID
8. Daemon və protocollar
9. Portlar və onların istifadə aralıqları
10. NAT, SSH, DHCP (relay), FTP, Firewall, NTP, VPN, DNS, Samba, Apache haqda ümumi məlumat
11. HTTP və Web server
12. Verilənlər bazası
13. CDP və VLAN
14. Netflow ümumi məlumat
15. Dynamic Routing

Bu məlumatlar aşağıdakı linklərdən əldə edilə bilər:

<https://ru.wikipedia.org/wiki/TCP/IP>

<http://en.wikipedia.org/wiki/Subnetwork>

[http://en.wikipedia.org/wiki/Gateway\\_\(telecommunications\)](http://en.wikipedia.org/wiki/Gateway_(telecommunications))

[http://en.wikipedia.org/wiki/Default\\_gateway](http://en.wikipedia.org/wiki/Default_gateway)

[https://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol)

[https://en.wikipedia.org/wiki/User\\_Datagram\\_Protocol](https://en.wikipedia.org/wiki/User_Datagram_Protocol)

[http://en.wikipedia.org/wiki/Private\\_network](http://en.wikipedia.org/wiki/Private_network)

[http://en.wikipedia.org/wiki/Reserved\\_IP\\_addresses](http://en.wikipedia.org/wiki/Reserved_IP_addresses)

[https://en.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](https://en.wikipedia.org/wiki/Address_Resolution_Protocol)

[https://en.wikipedia.org/wiki/MAC\\_address](https://en.wikipedia.org/wiki/MAC_address)

[https://en.wikipedia.org/wiki/Server\\_\(computing\)](https://en.wikipedia.org/wiki/Server_(computing))

[https://en.wikipedia.org/wiki/Server\\_\(computing\)](https://en.wikipedia.org/wiki/Server_(computing)2)

[http://en.wikipedia.org/wiki/Hard\\_disk\\_drive](http://en.wikipedia.org/wiki/Hard_disk_drive)

[http://en.wikipedia.org/wiki/Central\\_processing\\_unit](http://en.wikipedia.org/wiki/Central_processing_unit)

[http://en.wikipedia.org/wiki/Random-access\\_memory](http://en.wikipedia.org/wiki/Random-access_memory)

[http://en.wikipedia.org/wiki/File\\_system](http://en.wikipedia.org/wiki/File_system)

[http://en.wikipedia.org/wiki/Daemon\\_\(computing\)](http://en.wikipedia.org/wiki/Daemon_(computing))

[http://en.wikipedia.org/wiki/Communications\\_protocol](http://en.wikipedia.org/wiki/Communications_protocol)

[https://en.wikipedia.org/wiki/Port\\_\(computer\\_networking\)](https://en.wikipedia.org/wiki/Port_(computer_networking))

[https://en.wikipedia.org/wiki/Network\\_address\\_translation](https://en.wikipedia.org/wiki/Network_address_translation)

[https://en.wikipedia.org/wiki/Secure\\_Shell](https://en.wikipedia.org/wiki/Secure_Shell)

[https://en.wikipedia.org/wiki/Dynamic\\_Host\\_Configuration\\_Protocol](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)

[https://ru.wikipedia.org/wiki/DHCP\\_relay](https://ru.wikipedia.org/wiki/DHCP_relay)

[https://en.wikipedia.org/wiki/File\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol)

[https://en.wikipedia.org/wiki/Firewall\\_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing))

[https://en.wikipedia.org/wiki/Network\\_Time\\_Protocol](https://en.wikipedia.org/wiki/Network_Time_Protocol)

[https://en.wikipedia.org/wiki/Virtual\\_private\\_network](https://en.wikipedia.org/wiki/Virtual_private_network)

<https://ru.wikipedia.org/wiki/DNS>

[https://en.wikipedia.org/wiki/Samba\\_\(software\)](https://en.wikipedia.org/wiki/Samba_(software))

[https://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server)

[https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

[https://en.wikipedia.org/wiki/Web\\_server](https://en.wikipedia.org/wiki/Web_server)

<http://en.wikipedia.org/wiki/Database>

[https://en.wikipedia.org/wiki/Cisco\\_Discovery\\_Protocol](https://en.wikipedia.org/wiki/Cisco_Discovery_Protocol)

<https://ru.wikipedia.org/wiki/VLAN>

<https://en.wikipedia.org/wiki/NetFlow>

<http://www.comptechdoc.org/independent/networking/guide/netdynamicroute.html>

İstifadə edilən terminlərin açıqlanmasını aşağıdakı səhifədən oxuya bilərsiniz:

<http://www-03.ibm.com/ibm/history/documents/pdf/glossary.pdf>

# **1** İlkin nəzəriyyələr və əməliyyat sisteminin fərqli üsullarla yüklənməsi

- 12** Başlanğıc nəzəriyyələr
- 14** Sistemin fərqli nəzəriyyələr və uzaqdan SSH vasitəsilə yüklənməsi
- 37** Boot menyunun açıqlanması və inisializasiya strukturu
- 45** Fayl sistem strukturu
- 48** İstifadə ediləcək başlanğıc əmrlər

# **2** Symlink, SETUID, SETGID, fayl flaqları, resurslar və proseslər, rezerv nüsxələr, SUDO

- 56** Fayl tipləri, yetkilər, symlinklər
- 60** SETUID, SETGID, Sticky Bit
- 64** FreeBSD fayl flaqları, simvolik linklər, yeni diskin və USB Flash-in əməliyyat sisteminə əlavə edilməsi, UNIX Tape Drive
- 70** Sistem resursları və proseslər, faylların aktivliyinin təyin edilməsi
- 78** İşlək processlərin yoxlanılması və idarəedilməsi
- 85** Rezerv nüsxələr və bərpa edilməsi, istifadəçilərin SUDO ilə məhdudlaşdırılması

# **3** Vacib əmrlər, Swap idarə edilməsi, portlar, paketlər, arxivlər, sətir əməliyyatları

- 90** Sistemdə həmişə istifadə ediləcək vacib əmrlərin detallı açıqlanması
- 99** İşlək sistemdə Swap həcminin artırılması
- 102** Portlar, paketlər və onların idarə edilməsi
- 113** Fayl/qovluqların arxivlənməsi, sıxılması və grep sətir preprocessoru

## **4** SHELLin işləmə prinsipi, terminalda qısa keçidlər, CRON, istifadəçi üzərində əməliyyatlar, Vİ redaktoru, sistem RAID-ləri, sərt disklerin şifrələnməsi

- 122** SHELL, onun işləmə prinsipi, terminal qısa keçidləri, CRON
- 136** İstifadəçilərin yaradılması, silinməsi və deaktiv edilməsi
- 139** Vİ tekst redaktoru və vim
- 143** Fayl sistemlə praktik işlər
- 150** Disklerin bölünməsi və sistem RAID-ləri
- 152** Sistem RAID-ləri
- 160** Sərt disklerimizin şifrələnməsi

## **5** Qrafik interfeysin qurulması, kitabxanaların idarə edilməsi, Kernel kompilyasiyası, sistem control, INETD, DevFS, DevD, sistemin yenilənməsi

- 166** FreeBSD X11 quraşdırılması, paylaşılmış kitabxanaların idarə edilməsi və GPART fayl sistem genişlənməsi
- 173** Kernel kompilyasiya edilməsi, SYSCtrl idarəedilməsi, INETD
- 181** CheckSUM(İnformasiya bütövlüyü), Device File System(Devfs), DEVD quraşdırılması
- 185** İstifadəçilərin sistemə əlavə edilməsinin avtomatlaşdırılması
- 191** FreeBSD əməliyyat sisteminin, mənbə kodlarının və portların fərqli üsullarla yenilənməsi

## **6** Şəbəkənin quraşdırılması, WireLess quraşdırılması, WEBMIN, ARP, FTP, DHCP, şəbəkə alətləri

- 200** Şəbəkə kartının quraşdırılması, şəbəkə alətləri və Routing (şəbəkənin yönləndirilməsi)
- 209** WireLess quraşdırılması

- 212** Əməliyyat sistemimizin WEB browser vasitəsilə idarə olunması və şəbəkənin keçirtmə qabiliyyətinin yoxlanılması
- 215** ARP, FTP local servis və DHCP serverin quraşdırılması
- 223** Şəbəkə utilitləri

## **7** Dump- Restore, OpenSSH, RCS, TempFS- MemFS, Snapshot-lar, FSCK, Quota

- 232** FreeBSD Dump(Rezerv nüsxə) və Restore(Bərpa edilmə)
- 239** OpenSSH
- 253** Revision Control System, TempFS və MemFS
- 257** Snapshots, File System Checking və Disk Quota
- 263** Syslogd, Syslog-NG və Newsyslog

## **8** Wget, Curl, LFTP, Rsync, Unison, NFS, vaxt və tarix əməliyyatları

- 272** Uzaq serverlərə data ötürülməsi və götürülməsi üçün program təminatları
- 277** Rsync, Unison və NFS istifadəsi
- 283** Vaxt, tarix və təqvim əməliyyatları
- 289** Müxtəlif məqsədlərdə istifadə edilən utilitlərin açıqlanması

## **9** PF, İPFW, IPFilter FireWall-ların açıqlanması, İPFW- PF Fail2Ban

- 308** FreeBSD Firewall
- 314** PF FIREWALL
- 320** PF ALTQ tam açıqlama
- 331** PF Bridge Firewall

- 337** IPFW Firewall
- 349** IPFW Squid Transparent
- 360** IPFW PF Fail2Ban
- 365** IPFilter firewall

## 10 FAMP, DNS BIND

- 378** FreeBSD 10.1 x64 AMP(Apache MySQL PHP)
- 394** Berkeley Internet Domain (BIND) - DNS xidmətləri

## 11 Samba, AD ilə integrasiyası, badsect-lar, CLRI, NullFS, clonehdd

- 414** FreeBSD Samba
- 428** Samba server Active Directory Authentication
- 436** badsect – badblock(korlanmış disk blokları) faylların köçürülməsi üçün program.
- 442** CLRI - clear an inode, NullFS
- 446** Sıradan çıxmış fayl sistemin geri qaytarılması, clonehdd - sert diskin hissəsinin digərinə nüsxələnməsi

## 12 SFTP server Chroot, Tomcat8, MPD5 StS VPN, MPD5 PPTP RA VPN, NTOP netflow

- 454** OpenSSH SFTP server CHroot directory
- 458** FreeBSD 10.1 x64 Tomcat8
- 463** FreeBSD MPD5 Site-to-Site VPN
- 471** FreeBSD MPD5 PPTP(Remote Access VPN)
- 478** FreeBSD NTOP(NetFlow Trafikin monitoring edilməsi)

# **13** IPSec StS VPN, FreeBSD-Cisco StS VPN, Stunnel, HAST Cluster

- 488** FreeBSD IPSec Site-to-Site VPN
- 501** FreeBSD ilə Cisco arasında IPSec vasitəsilə Site-to-Site VPN qurulması
- 512** FreeBSD Stunnel
- 520** FreeBSD HAST Cluster

# **14** CDP, .1Q, Dinamik Routinq, Bridge, LACP, LAG, CARP, DHCP Relay

- 536** UNİX CDP, FreeBSD .1Q və Dinamik Routing
- 547** Bridging(Körpülənmə)
- 554** Link Aggregation və Failover
- 561** Common Address Redundancy Protocol(CARP)

# **15** FreeBSD inzibatçı üçün vacib olanlar

- 570** İstifadəçilər və qrupların idarə edilməsi.
  - 585** Yetki hüquqlarının idarə edilməsi
  - 595** Access Control List
  - 603** Spesifik flaglar
- 
- 607** İndekslər
  - 621** İstifadə olunmuş ədəbiyyat siyahısı

# BÖLÜM 1

## İllkin nəzəriyyələr və əməliyyat sisteminin fərqli üsullarla yüklənməsi

- / Başlanğıc nəzəriyyələr
- / Sistemin fərqli versiyalarla və uzaqdan SSH vasitəsilə yüklənməsi
- / Boot menyunun açıqlanması və inisializasiya strukturu
- / Fayl sistem strukturu
- / İstifadə ediləcək başlanğıc əmrlər

Başlığımızda FreeBSD əməliyyat sisteminin fərqli versiyalarla və fərqli üsullarla yüklənməsini öyrənəcəyik. Yüklənmənin hansı ardıcılılıqda getdiyini, yükləyici vasitəsi ilə işlərin aparılması və GRUB yükləyicisinin qurulması göstərilir. İstinadlar və fayl sistemində vacib qovluqların açıqlanması izah olunur. Başlanğıc əmrlər açıqlanır.

# Başlangıç nəzəriyyələr

Başlangıç olaraq FreeBSD haqqında nəzəri məlumat vermək istərdim. FreeBSD 4.4 BSD UNİX nəslindən gəlir və hal-hazırkı buraxılışları 8.4, 9.3, 10.1-dir. Tam açıq kodlu UNİX sistemidir.

Ümumilikdə 24000-dən çox Port(Package)-i var, <https://www.freebsd.org/ports/> linkindən ətraflı məlumat əldə edə bilərsiniz.

**Port** - Portlarda tam açıq kodlar olur ki, onları da muxtəlif opsiyalarla kompliyasiya etmək imkanı var. Unutmayın ki, asılılıq və böyüklük hacminə görə portların kompliyasiyası çox uzun zaman ala bilər. Müsbət xüsusiyyəti inzibatçını inkişaf etdirməsidir.

**Qeyd:** Portları FreeBSD paket idarəedicisindən tam idarə etmək mümkündür - `install/ remove/upgrade/manage`

**Packages** – Mənbə kodlarının kompliyasiya olunmuş formasıdır. Sadəcə sistemə yüklənilir. Portlardan fərqli olaraq, çox qısa zamanda yüklenmək imkanına malikdir. Mənfi xüsusiyyəti inzibatçını bilik tənbəlləşməsinə gətirib çıxarmasıdır.

**Qeyd:** Paketləri portlar kimi, FreeBSD package managerlə tam idarə etmək mümkündür .

**Qeyd:** Proqramları idarə etmək üçün 2 üsul var: Port (Source code) və Paketlər (Binar fayllar).

Aşağıda qeyd olunan avadanlıqların dəstəklənməsi mümkündür.

- a. i386(32-bit)
- b. ia64(64-bit)
- c. amd64(x86\_64) - bu, istifadə olunur AMD/Intel 64-bit (XEON/Core 2 processorlarda və s.)
- d. Alpha
- e. Sparc64
- f. ARM - daxilidir.
- g. PowerPC

3 tip buraxılışı mövcuddur: Release (çox az səhvi olan), **Beta** və **SNAPSHOT** (çox səhvi olan).

Çox arxitekturalı ISO nüsxələri, Linux binar fayllarının yerinə yetirilməsi imkanı var(Yüklənən özək modulundan edilir - KLM). FreeBSD kerneli Linux kerneli kimi modulludur. Mətn bazalı yüklənməni dəstəkləyir. **Sysinstall** köhnə FreeBSD versiyaları və **Bsdconfig** yeni FreeBSD versiyalarında yüklənmədən sonra idarəetmə işlərində istifadə edilir.

FreeBSD-ni HTTP, **FTP** (daha çox istifadə olunur) və **Bit Torrent** vasitəsilə rəsmi saytından <https://www.freebsd.org/where.html> yükleyə bilərsiniz.

Yükləməni daxili media (CD, DVD, USB daşıyıcıdan) və şəbəkədən (FTP, NFS) etmək mümkündür.

**Qeyd:** FreeBSD yüklənmə müddəti **ALT-F1|F2** keçidləriniz Console-lar arası keçidi dəstəkləyir

Qrafik GUI idarəetmə üçün GNOME və KDE-ni və disk şifrələnməsini dəstəkləyir (**dm-crypt**). Boot yükleyicisi GRUB2-dir.

Yükləmək üçün sistem tələbləri aşağıdakılardır:

- a. 486(min) - 1 GHz P4 processor və ya daha yaxşısı
- b. 24MB (min) - 256 MB RAM və ya daha çox
- c. 1 GB və ya daha çox HDD.

Nəzəri açıqlamaların sonunda bildirmək istərdim ki, ümumiyyətlə, şəbəkədə istifadə ediləcək program təminatları üçün FreeBSD ən uğurlusudur. Oracle və Java ilə bağlı programların işləmə mühiti daha çox Linuxlarda effektiv olur. Ancaq apache-tomcat server-də FreeBSD üzərində çox yaxşı və dayanıqlı işləyir.

# Sistemin fərqli versiyalarla və uzaqdan SSH vasitəsilə yüklənməsi

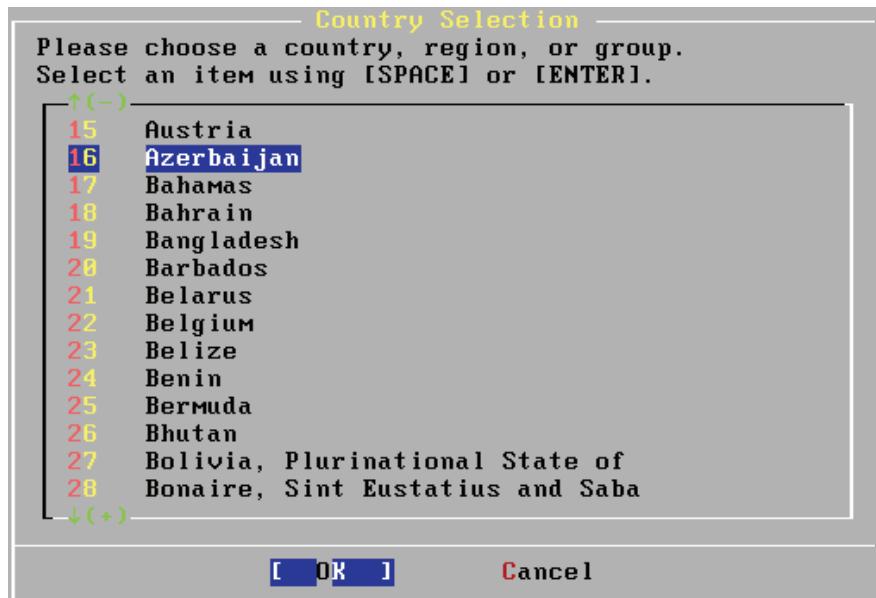
Nəzərimizdə tutaq ki, hər bir əməliyyat sistemi 20 GB-lıq həcmdə olan disklərə yüklənilib.

Öncə 8.4 versiyası ilə başlayaq:

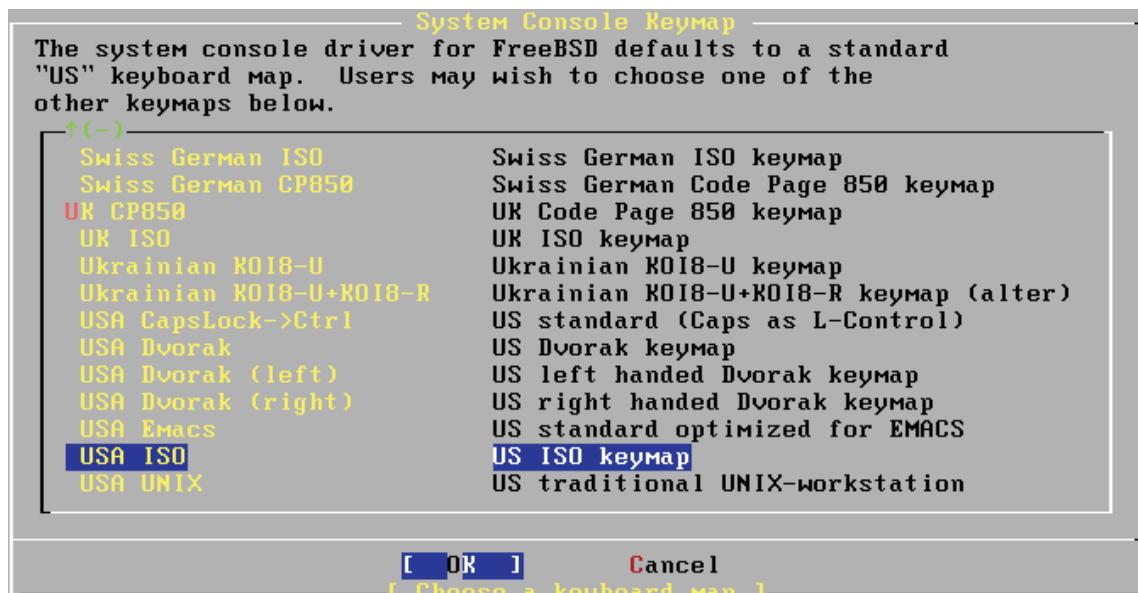
1. Aşağıda gördünüz şəkil ISO nüsxədən VmWare maşınınında ilk yüklenmə səhifəsidir (Heç nəyə toxunmuruq və sistem yüklenməsinə davam edirik. Görünən menyu barəsində bir azdan ətraflı açıqlama verəcəyik):



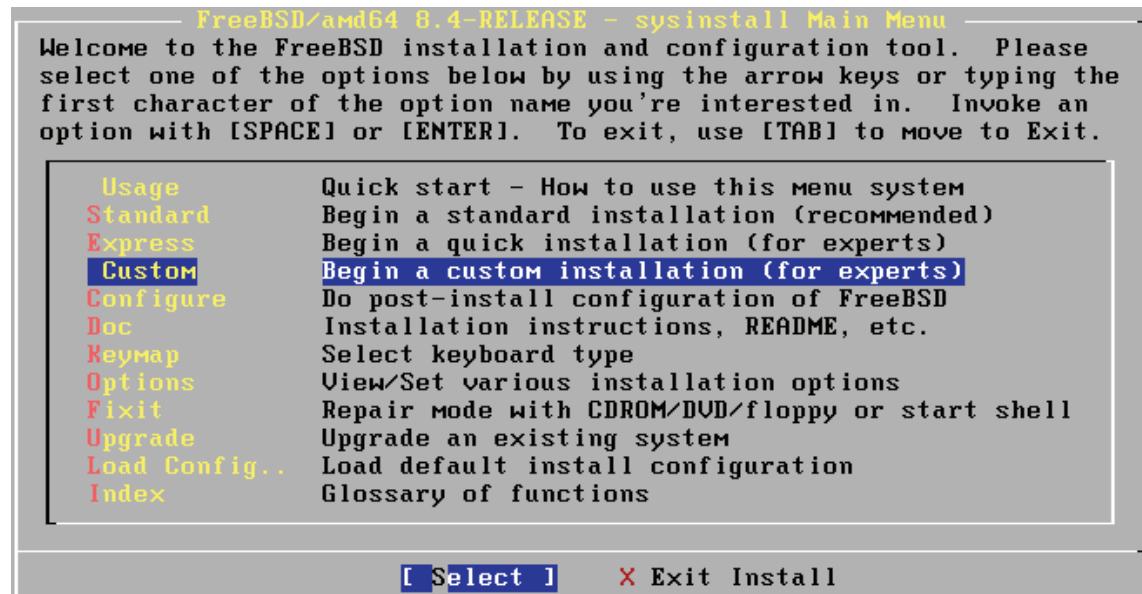
2. Sonra yerleşdiyimiz ərazini, yeni Azərbaycanı seçib **OK** düymesini sıxırıq:



3. Klaviatura tipini təyin edirik (Susmaya görə oları seçirik, USA ISO):

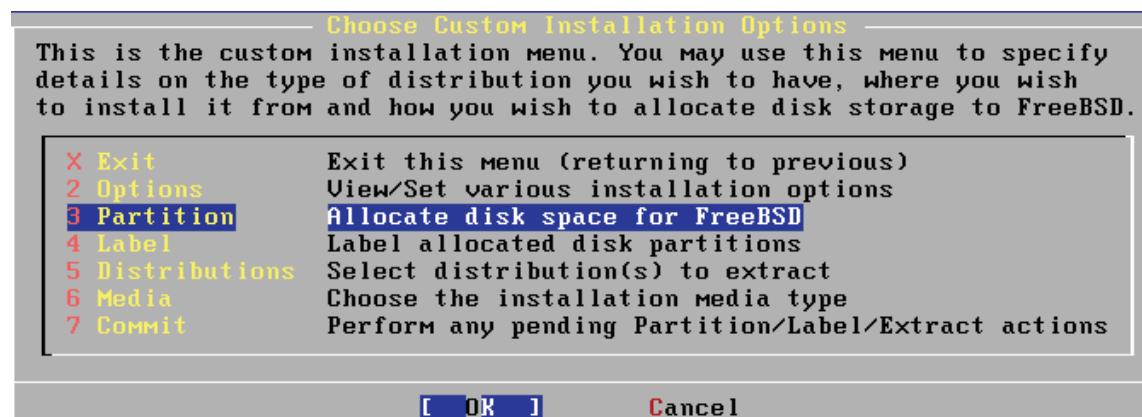


4. Şəkildə göründüyü kimi, köməkçi hissədə menyunun idarə edilməsi məqsədilə məlumat verilir. Biz bu menyudan gələcəkdə çox istifadə edəcəyik, ancaq hələ yükləməmizi davam edək. **Custom** seçirik və **OK** düyməsini sıxırıq:



5. İşimizə açıqlamalar verərək davam edirik.

**Options** – Bu bölümde yüklenəcək FreeBSD versiyası FreeBSD repositoriyalarına qoşulduğda anonim ftp istifadəçi adı, default metn redaktoru tipi və bəzi dəyişən tipləri yüklenməyə başlamazdan önce təyin edə bilərsiniz.



**Partition** bölümünü seçirik ve **OK** düymesini sıxırıq. Açılan şəkil bize fiziki disk haqqında məlumatlar verir. **da0** adlı fiziki disk və həcmının 20GB olması göstərilir. Göründüyü kimi, C simvolunu sıxmaqla diskimizi yaratmış oluruq.

```
Disk name: da0 FDISK Partition Editor
DISK Geometry: 2610 cyls/255 heads/63 sectors = 41929650 sectors (20473MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype   Flags
  0    41943040  41943039      -       12    unused      0

The following commands are supported (in upper or lower case):
A = Use Entire Disk  G = set Drive Geometry  C = Create Slice
D = Delete Slice     Z = Toggle Size Units  S = Set Bootable  : = Expert M.
T = Change Type      U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.
```

C simvolunu sıxdıqdan sonra isə aşağıdakı şəkil ekranda görünür. Ayıracaq slice-ımız üçün həcmi megabayt, gigabayt və ya blocklarla təyin edə bilərik. Ancaq biz bloklarda göstərilən tam həcmi seçib **OK** düymesini sıxırıq.

```
Disk name: da0 FDISK Partition Editor
DISK Geometry: 2610 cyls/255 heads/63 sectors = 41929650 sectors (20473MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype   Flags
  0    41943040  41943039      -       12    unused      0

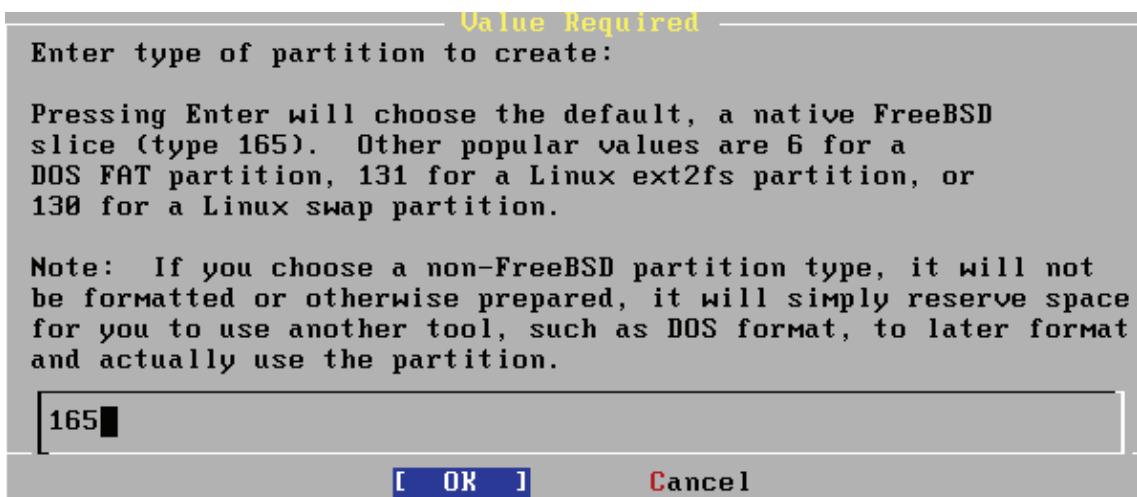
Value Required
Please specify the size for new FreeBSD slice in blocks
or append a trailing 'M' for Megabytes (e.g. 20M).
[ 41943040 ]

The follow [ OK ] [ Cancel ]

A = Use Entire Disk  G = set Drive Geometry  C = Create Slice
D = Delete Slice     Z = Toggle Size Units  S = Set Bootable  : = Expert M.
T = Change Type      U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.
```

Ayırdığımız disk hissəsinin format tipini təyin edirik. Yəni **UFS2(165)** seçib **OK** düyməsini sıxırıq.



Nəticəmiz aşağıdakı kimi alınacaq (**Esc** düyməsini sıxırıq):

The screenshot shows the final output of the fdisk command. It displays the disk name as da0 and the disk geometry as 2610 cylinders/255 heads/63 sectors = 41929650 sectors (20473MB). The table lists three partitions:

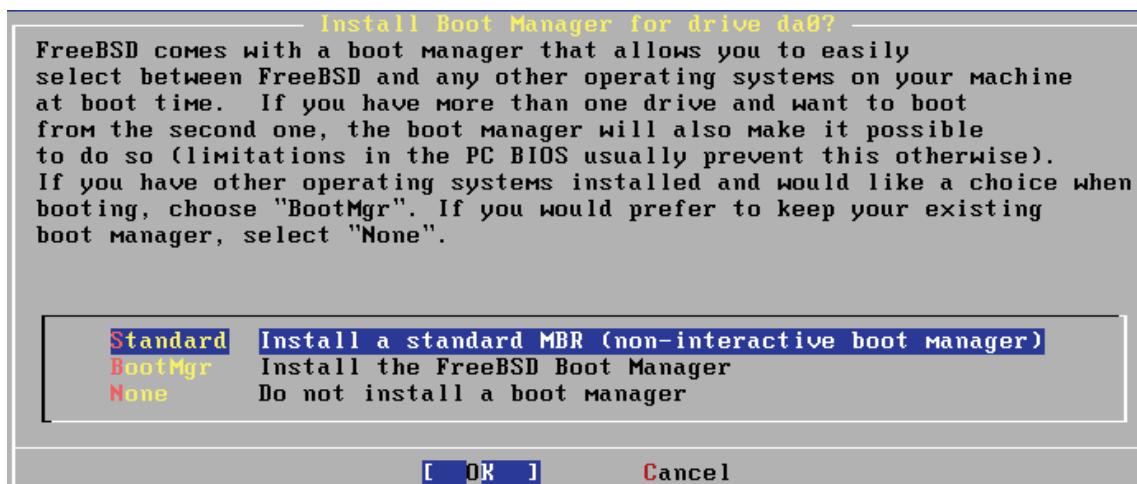
Offset	Size(ST)	End	Name	PType	Desc	Subtype	Flags
0	63	62	-	12	unused	0	
63	41929587	41929649	da0s1	8	freebsd	165	
<b>41929650</b>	<b>13390</b>	<b>41943039</b>	-	<b>12</b>	<b>unused</b>	<b>0</b>	

The following commands are supported (in upper or lower case):

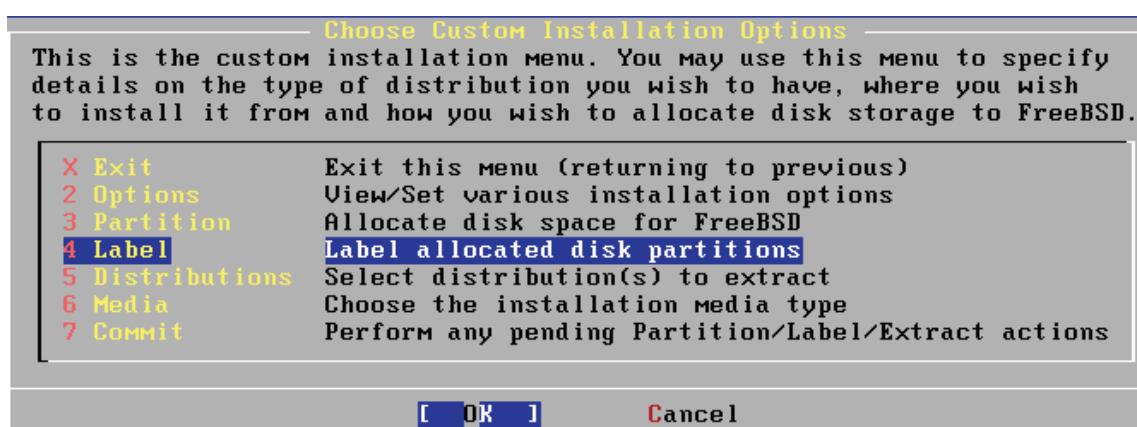
A = Use Entire Disk    G = set Drive Geometry    C = Create Slice  
D = Delete Slice    Z = Toggle Size Units    S = Set Bootable    I = Expert M.  
T = Change Type    U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.

**ESC** ilə çıkış etdikdən sonra Boot idarəedicisini seçirik. Yəni MBR (Master Boot Record). **Standart** seçirik və **OK** düyməsini sıxırıq:



Ardınca isə 5-ci mövqedə olan şəkile qaydır və **Label** seçib **OK** düyməsini sıxırıq:



Yeni şəkildə aq rəngli xəttlə qeyd etdiyim kimi, **da0** diskimizin ilk slice-na tam **20GB**-lik həcmi və **s1** slice adının verilməsi haqqında məlumatı görürük. Öncədən deyim ki, növbəti addımı qabaqcıl inzibatçı kimi, ya da başlayan şəxs kimi görə bilərsiniz. Hər halda qismən irəliləmiş inzibatçı kimi, gediləcək addımları nəzərinizə çatdırıram. C simvolunu sıxbi diskimizi irəliləmiş inzibatçı kimi, özümüzə uyğun olan hissələrə bölək. Bu halda biz fayl sistem tipini(**UFS2** və ya **swap**) təyin etməliyik.

```

FreeBSD Disklabel Editor

Disk: da0      Partition name: da0s1      Free: 41929587 blocks (20473MB)
Part      Mount          Size Newfs   Part      Mount          Size Newfs
----      ----          -----      ---      ----          -----

```

The following commands are valid here (upper or lower case):  
C = Create D = Delete M = Mount pt.  
N = Newfs Opts Q = Finish S = Toggle SoftUpdates Z = Custom Newfs  
T = Toggle Newfs U = Undo A = Auto Defaults R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.

C düymesini sıxıqdan sonra isə, yaradacağımız **partition** həcmini **bloklarla** (yəni rəqəmlərlə), **G** (giabayıt), **M** (Meqabayıt) və **C** (silindirlərlə) müəyyən edə bilərik. Tam həcmi seçirik və toxunmadan OK düymesini sıxırıq. Misal üçün, ardıcıl şəkildə fayl sistem üçün **10GB** seçirik. Unutmayın ki, ardıcıl gedən şəkillər sadəcə imkan haqqında məlumatdır. Bizim halda **A** düymesini sıxıb avtomatik davam edəcəyik.

```

FreeBSD Disklabel Editor

Disk: da0      Partition name: da0s1      Free: 41929587 blocks (20473MB)
Part      Mount          Size Newfs   Part      Mount          Size Newfs
----      ----          -----      ---      ----          -----

```

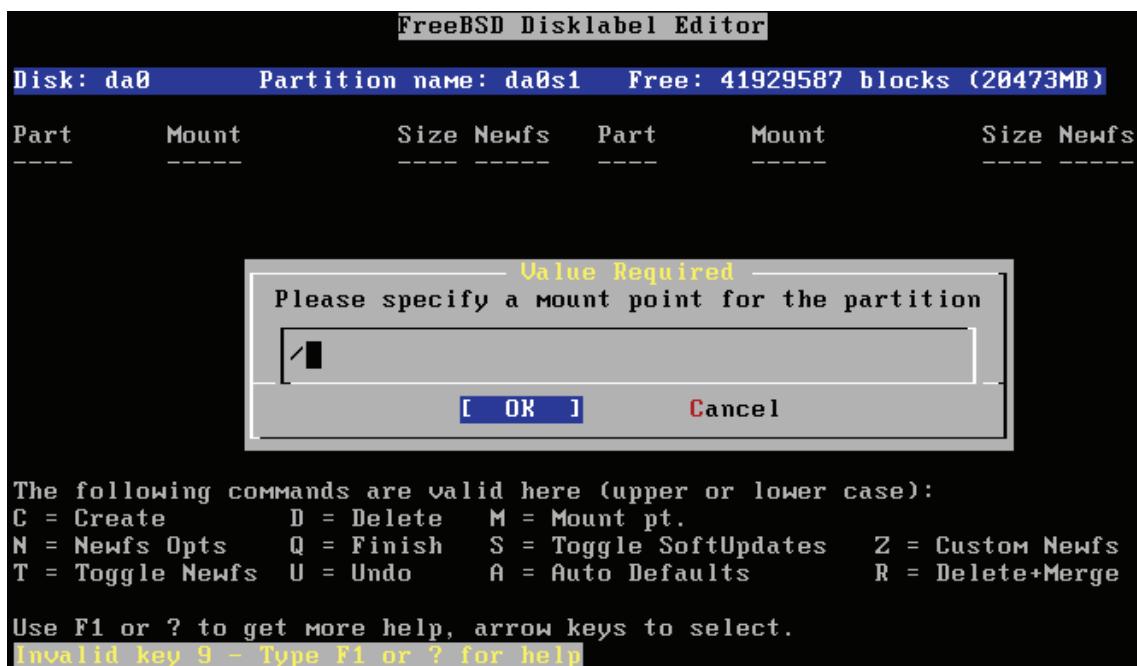
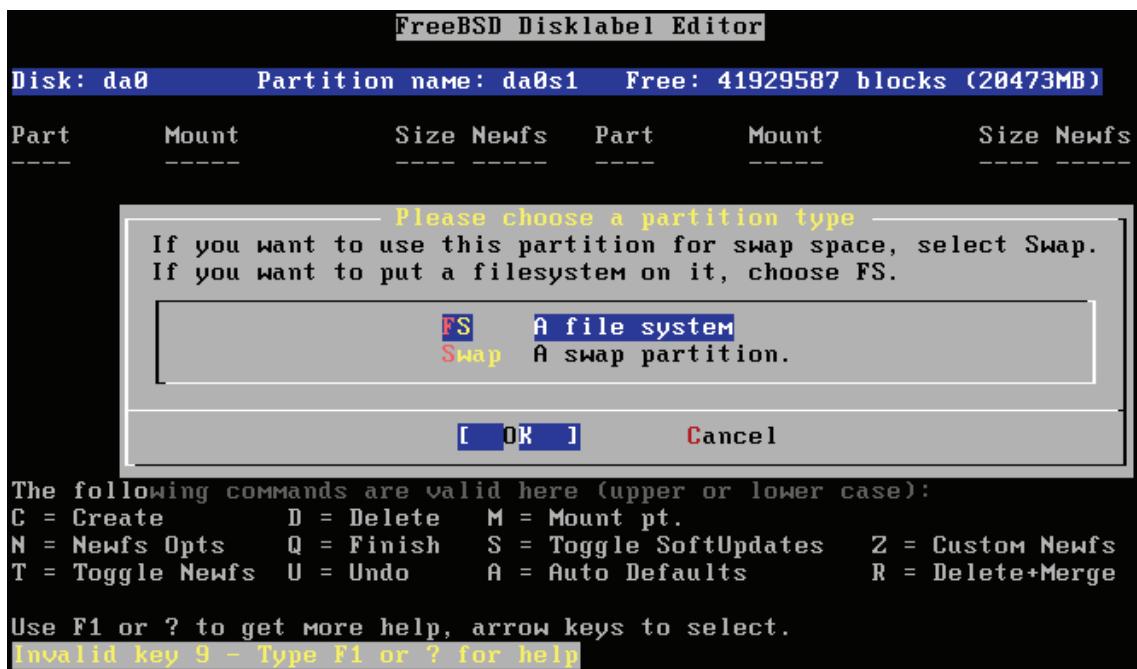
**Value Required**  
Please specify the partition size in blocks or append a trailing G for gigabytes, M for megabytes, or C for cylinders.  
41929587 blocks (20473MB) are free.

**10GB**

[ OK ]      [ Cancel ]

The following commands are valid here (upper or lower case):  
C = Create D = Delete M = Mount pt.  
N = Newfs Opts Q = Finish S = Toggle SoftUpdates Z = Custom Newfs  
T = Toggle Newfs U = Undo A = Auto Defaults R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.  
**Invalid key 9 - Type F1 or ? for help**



```

FreeBSD Disklabel Editor

Disk: da0      Partition name: da0s1      Free: 20958067 blocks (10233MB)

Part      Mount      Size Newfs      Part      Mount      Size Newfs
----      ----      -----      ----      ----      -----      -----
da0s1a    /          10240MB UFS2      Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete      M = Mount pt.
N = Newfs Opts  Q = Finish     S = Toggle SoftUpdates  Z = Custom Newfs
T = Toggle Newfs  U = Undo      A = Auto Defaults      R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.

```

A simvolunu sıxırıq və aşağıdakı nəticəni əldə edirik. Gördüyüümüz şəkildə disk bölgümüz **UFS2** fayl sistemi ilə fərqli **mount** ediləcək ünvanlara və **swap** hissəsinə avtomatik ayılmışdır.

```

FreeBSD Disklabel Editor

Disk: da0      Partition name: da0s1      Free: 0 blocks (0MB)

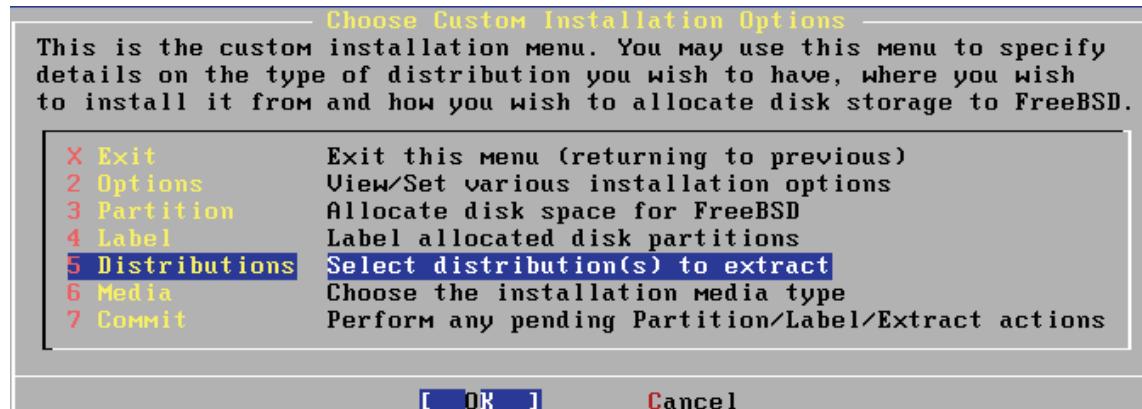
Part      Mount      Size Newfs      Part      Mount      Size Newfs
----      ----      -----      ----      ----      -----      -----
da0s1a    /          1024MB UFS2      Y
da0s1b    swap       4055MB SWAP
da0s1d    /var        6123MB UFS2+S  Y
da0s1e    /tmp        1024MB UFS2+S  Y
da0s1f    /usr        8246MB UFS2+S  Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete      M = Mount pt.
N = Newfs Opts  Q = Finish     S = Toggle SoftUpdates  Z = Custom Newfs
T = Toggle Newfs  U = Undo      A = Auto Defaults      R = Delete+Merge

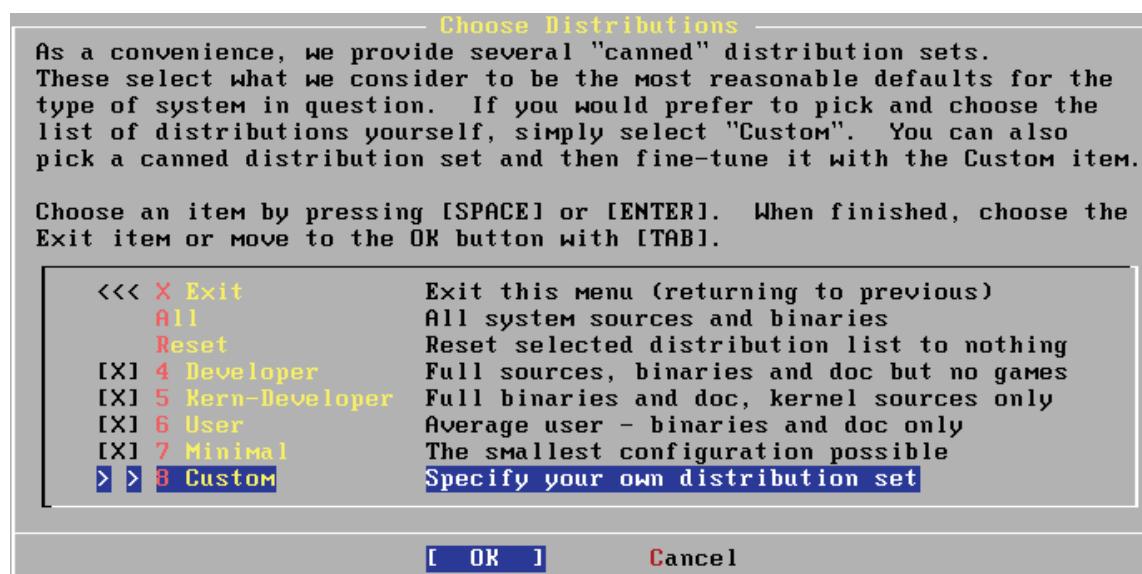
Use F1 or ? to get more help, arrow keys to select.

```

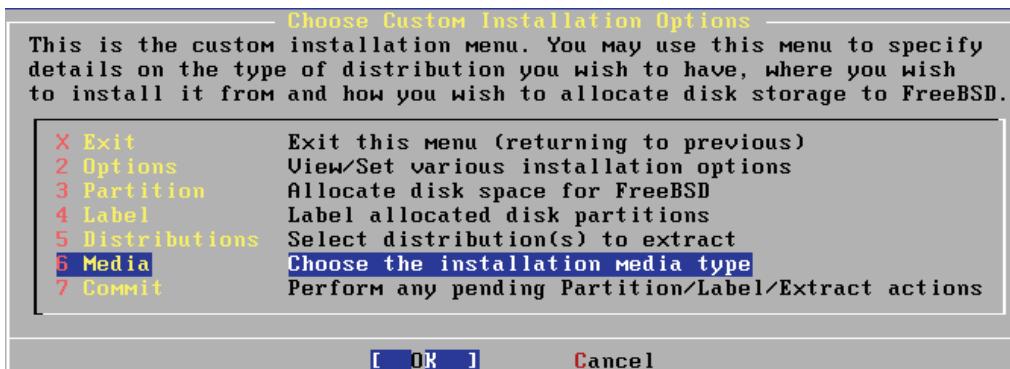
ESC düyməsini sıxırıq və 5-ci mövqedə olan şəkilə qayıdırıq. **Distributions** seçirik və **OK** düyməsini sıxırıq. Burada bizə lazım olan yüklemək istədiyimiz paketləri, mənbə kodları, binar faylları və sənədləri seçirik.



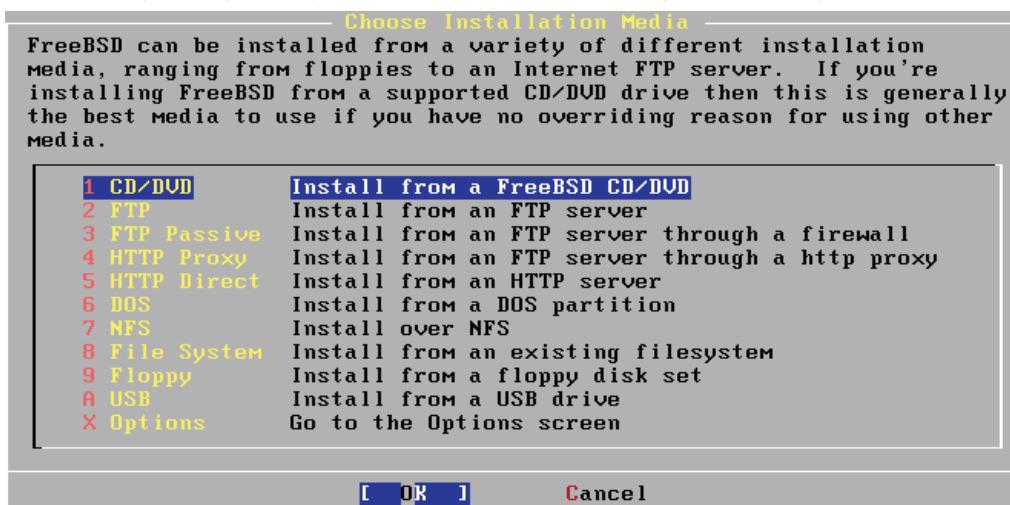
**Developer**, **Kern-Developer**, **User** seçimin hər biri üçün dil olaraq ancaq EN(İngilis) dili seçirik. **Custom** bölümündə isə **games**-dən başqa hər şeyi seçirik. Neticə aşağıdakı kimi olacaq.



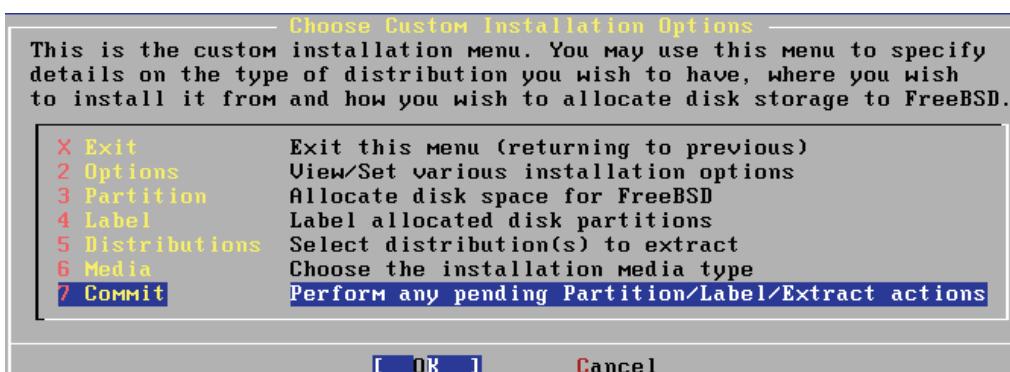
Sonra **OK** düyməsini sıxırıq və 5-ci mövqedə olan menyuya qayıdırıb media bölümünüə daxil olub **OK** düyməsini sıxırıq.



Diskdən yüklediyimizə görə, CD/DVD seçirik və OK düyməsini sıxırıq.



Nəticədə, yenə 5-ci menyuya qayıdırıq və Commit seçib OK düyməsini sıxırıq.



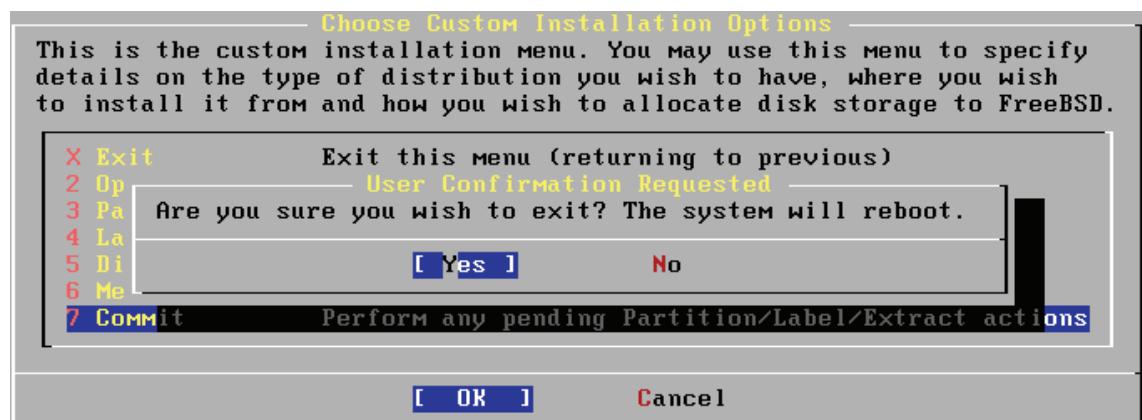
Bu pəncərə vasitəsi ilə sizə bildirilir ki, əgər diskinizdə lazımi məlumatlar varsa, onlar silinəcək. Xahiş olunur, rezerv nüsxə götürüb sonra davam edəsiniz. **Yes** düyməsini sıxırıq və davam edirik. Bundan sonra yüklənmə başlayır.



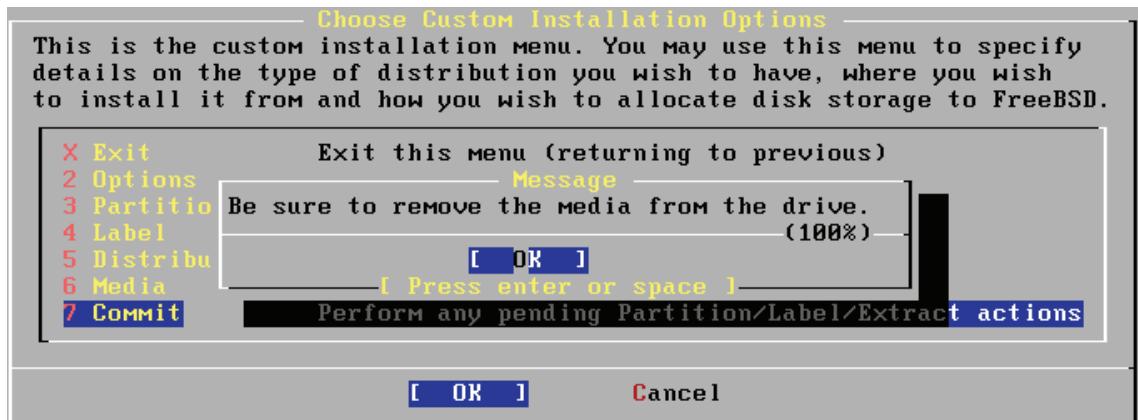
Yüklənmədən sonra əsas menyuya qayıtmaq istəyib-istəmədiyimizi bizdən soruşur və **No** deyirik.



Növbəti suallarda **Cancel** və **Exit Install** deyə qərar veririk. Soruşular ki, çıxışınıza eminsinizmi? Bu halda sistem yenidən yüklənmə edəcək. Və biz **Yes** deyirik.



Diskin çıkışından emin olmamızı bildirir və **OK** sıxırıq davam edirik.



Nəticədə **root** istifadəçi ilə yeni yüklenmiş sistemə daxil ola bilərsiniz. Sistem yeni yükləndiyinə görə root istifadəçinin şifrəsi hələ olmayıcaq və siz özünüz kitabımızın ardıcılığında şifrəni təyin edəcəksiniz.

İndi isə **FreeBSD9.3** və **FreeBSD10.1**-in yüklenmə qaydasını açıqlayacaq:

**FreeBSD8.4**-də olduğu kimi, ilk görünüş demək olar ki, eynidir. **ENTER** sıxırıq və davam edirik.



Çıxan menyudan **install** seçirik:

```
Welcome
Welcome to FreeBSD! Would you
like to begin an installation
or use the live CD?

<Install> < Shell > <Live CD>
```

Susmaya görə olan klaviatura tipi qalır və **Select** sıxıb davam edirik(Yəni ingilis dili):

```
Keymap Selection
The system console driver for FreeBSD defaults to standard "US"
keyboard map. Other keymaps can be chosen below.

>>> Continue with default keymap
--> Test default keymap
( ) Armenian phonetic layout
( ) Belarusian Codepage 1131
( ) Belarusian Codepage 1251
( ) Belarusian ISO-8859-5
( ) Belgian ISO-8859-1
( ) Belgian ISO-8859-1 (accent keys)
( ) Brazilian 275 Codepage 850
( ) Brazilian 275 ISO-8859-1
( ) Brazilian 275 ISO-8859-1 (accent keys)
( ) Bulgarian BDS
+--v(+)

[Select] [Cancel]
[Press arrows, TAB or ENTER]
```

Server üçün **hostname** daxil edib, **OK** düyməsini sıxaraq davam edirik.

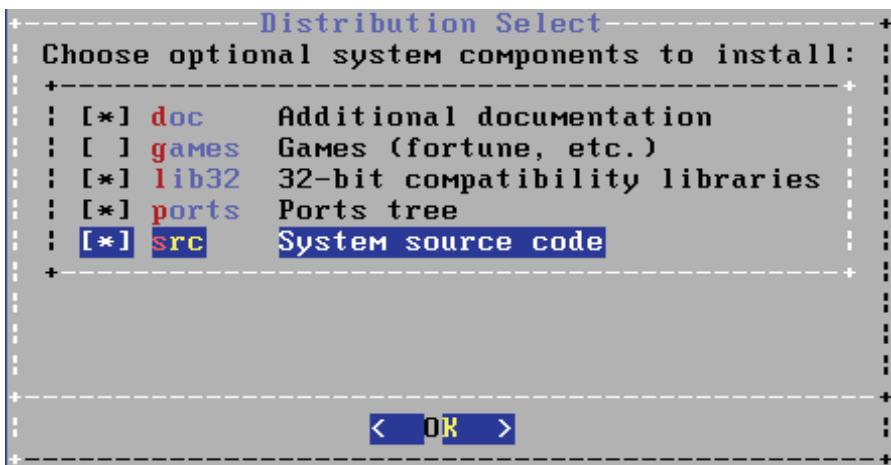
```
Set Hostname
Please choose a hostname for this machine.

If you are running on a managed network, please
ask your network administrator for an appropriate
name.

!freebsd9.3

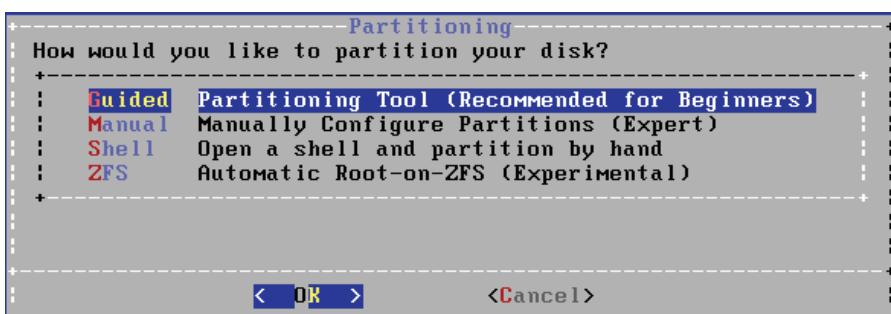
< OK >
```

Göründüyü kimi, `games`-den başka her şeyi seçirik ve **OK** düymesini sıxırıq.

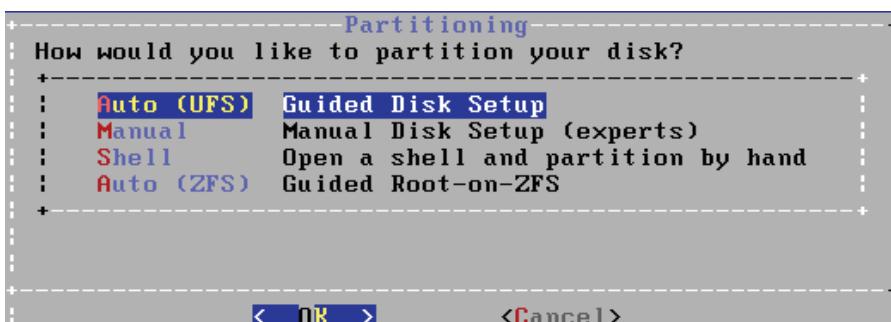


Diskbölgüsünü yeni başladığımız için, **Guided** seçirik ve **OK** düymesini sıxırıq. **FreeBSD 9.3** ve **10.1** yüklenmesində yalnız bu hissədə fərq var və aşağıdakı şəkillərdə həmin fərq göstərilir.

### FreeBSD 9.3



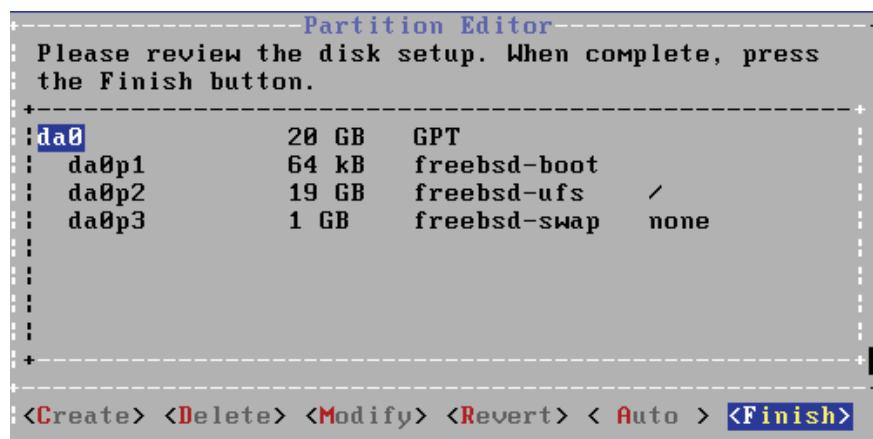
### FreeBSD 10.1



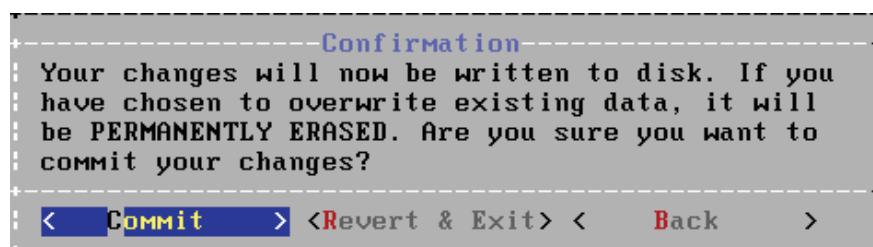
Sonra bizdən sorușulacaq ki, siz **da0** diskini bütövlükde FreeBSD üçün istifadə etmək isteyirsiniz, yoxsa eyni diskə başqa əməliyyat sistemini də yükləyəcəksiniz? Xəbərdarlıq edir ki, bütün diskni seçsəniz, hər şeyi siləcək.



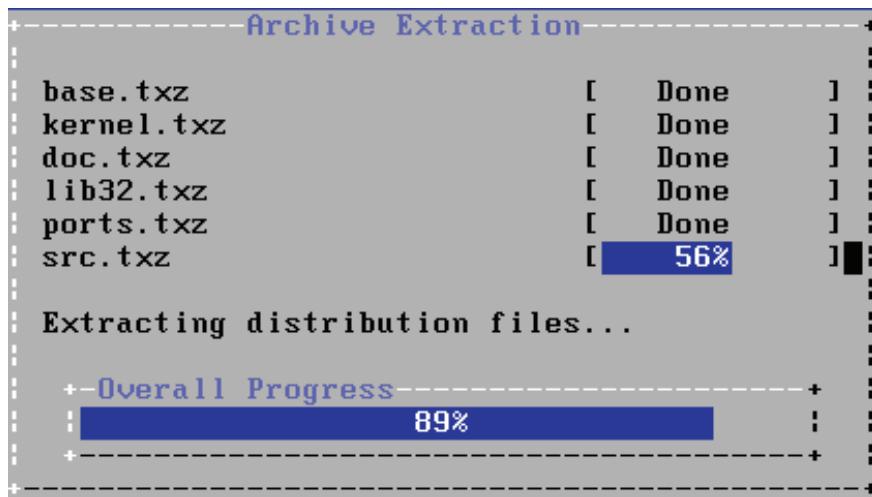
Növbəti şəkil çap ediləcək və **Finish** seçirik.



Sonra təsdiq edib davam edirik. **Commit** düyməsini sıxırıq.



Yüklənmə aşağıdakı kimi gedir.



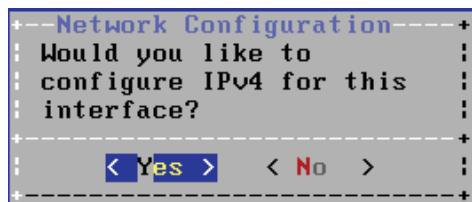
Yüklənmə müddətində **root** istifadəçisi üçün şifrə təyin edirik (eyni şifrəni iki dəfə təkrar daxil edirik):

```
FreeBSD Installer
=====
Please select a password for the system management account (root):
Changing local password for root
New Password:
Retype New Password:
```

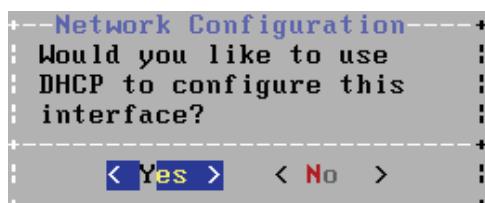
Qurulacaq şəbəkə kartı seçilir və **OK** düyməsini sıxırıq (Bizim halda **em0**):

```
Network Configuration
Please select a network interface to configure:
+-----+
|   em0    Intel(R) PRO/1000 Legacy Network Connection 1.0.6 |
|   plip0   PLIP network interface                            |
+-----+
|          < OK >          <Cancel>
```

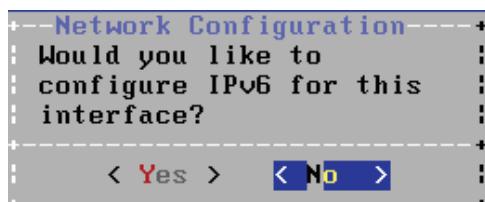
Bu kart-da **IPv4** quracağımız soruşular və **Yes** seçirik.



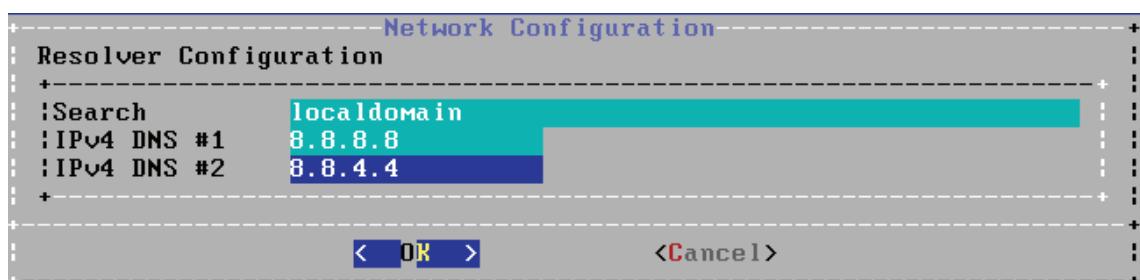
IP-nin **DHCP** ilə alacağımı təyin edirəm, çünkü mənim vəziyyətimdə **VMWare NAT** idi. **Yes** deyirik.



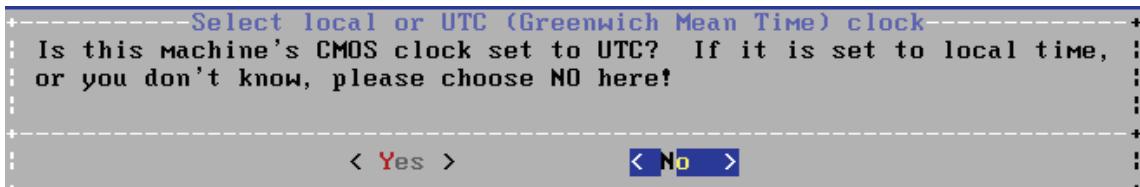
Növbəti menyuda **IPv6**-ya **No** deyirik.



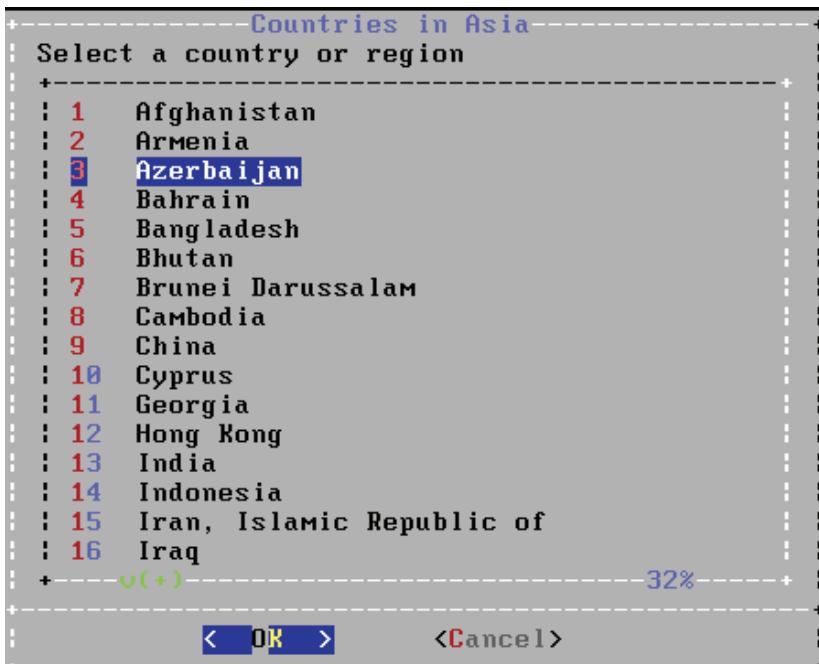
DNS istifadəçi quraşdırımızı aşağıdakı kimi edib **OK** düyməsini sıxırıq.



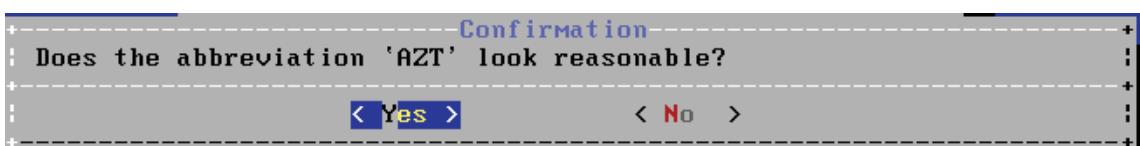
Serverimizin yerləşdiyi vaxt aralığını təyin etmək üçün **No** sıxırıq.



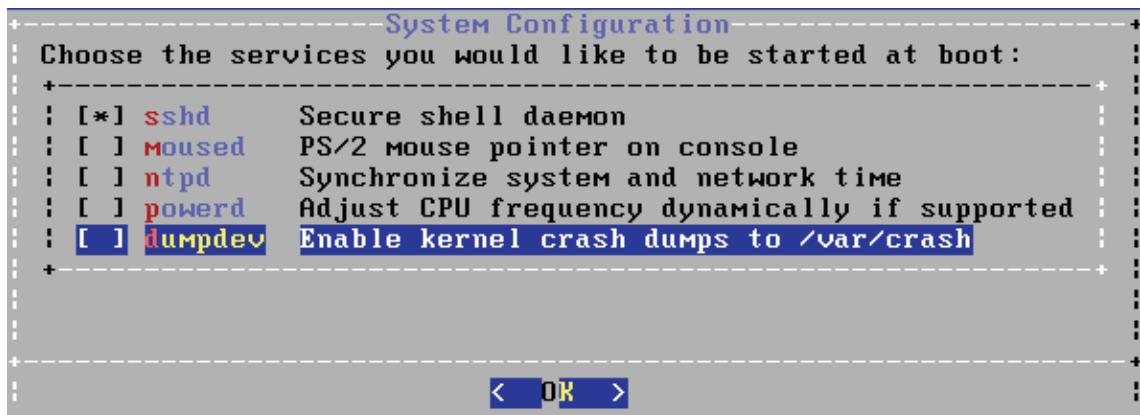
Asia altında Azərbaycanı seçib **Ok** düyməsini sıxırıq.



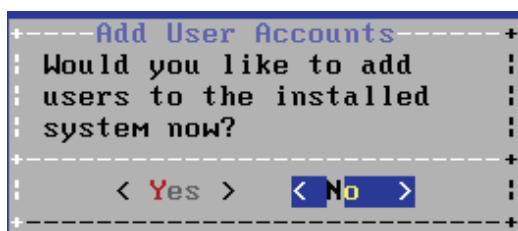
AZT-in bize uyğun olması soruşturulacaq və **Yes** düyməsini sıxırıq.



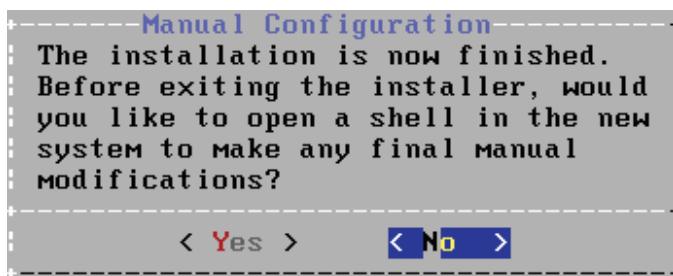
Sistemin susmaya görə olan quraşdırılmalarından **sshd** seçirik və **OK** düyməsini sıxırıq.



Sistemə yeni istifadəçinin əlavə edilməsi soruşular və hal-hazırda bu, bizi lazımlığı üçün **No** deyirik. Yüklənmədən çıxmak üçün **Exit** seçirik və **Enter** düyməsini sıxırıq.



Artıq yüklenməmiz bitmişdir və yenidən dəyişiklik üçün **SHELL** mühitinə qayıtmaq istəmədiyimizə görə **No** sıxaraq bildiririk.



Sonda sistemə yenidənyüklənmə əmri veririk.

```
+-- Complete --+
| Installation of FreeBSD
| complete! Would you like
| to reboot into the
| installed system now?
+-----+
| <Reboot> <Live CD> |
```

Nəticədə, sistem açılır və yüklənmə prosesində **root** istifadəçi üçün yaratdığımız şifrəni sistemə **root**-la giriş etdikdə daxil edirik.

### FreeBSD 10.1 SSH vasitəsi ilə yüklemək qaydası

Məqsədimiz PUBLIC IP üzərindən SSH ilə FreeBSD-nin yüklənməsidir. Yəni deyək ki, Gəncə şəhərində bir ədəd FreeBSD server işə salınmalıdır. Ancaq təhlükəsizlik üçün bunu SSH üzərindən şifrələnmiş yolla etmək lazımdır. Bunun üçün Gəncə tərəfdə **FreeBSD10.1 x64** ISO nüsxəsi arıq mövcud olmalıdır. Bu nüsxə serverlə, ya DVDROM-la, ya da yüklənilə biləcək USB FLASH ilə əlaqələndirilməlidir. Aşağıdakı ardıcılıqla başlanğıc üçün Gəncə şəhəri tərəfdə ediləcək ilkin addımlar şəkillərlə açıqlanır.

Yüklənmənin ilk səhifəsində **Live CD** seçib, **ENTER**-i sıxırıq.

```
+-- Welcome --+
| Welcome to FreeBSD! Would you
| like to begin an installation
| or use the live CD?
+-----+
| <Install> < Shell > <Live CD> |
```

Şifrəsiz **root** istifadəçi adını daxil edib, **ENTER**-i sıxırıq.

```
Updating motd: /etc/motd is not writable, update failed.
Mounting late file systems..
Configuring syscons: blanktime.
Starting cron.
Starting background file system checks in 60 seconds.

Sun Mar  8 10:14:31 UTC 2015

FreeBSD/amd64 (Amnesiac) (ttyv0)

login: root
```

**SSH** daemon-u işə salmazdan önce biz yazılı bilən müvəqqəti **/etc** qovluğu yaratmalıyıq. Bu qovluğu **unionfs**-lə mount edirik ki, yazılı bilməyən diskdə yazma hüququmuz olsun.

```
root@:~ # mkdir /tmp/etc  
root@:~ # mount_unionfs /tmp/etc/ /etc/
```

Artıq SSH-i quraşdırırıq.

**/etc/ssh/sshd\_config** faylinin içinde aşağıda göstərildiyi kimi, **PermitRootLogin** sətrini taparaq **Yes** edib, faylı yadda saxlayıb çıxırıq.

```
# Authentication:  
  
#LoginGraceTime 2m  
PermitRootLogin yes  
Port 22
```

SSH daemon-u bir dəfə üçün işə salırıq.

```
/etc/rc.d/sshd onestart
```

**root** istifadəçimizə şifrə təyin edirik ki, uzaqdan daxil olduqda istifadə etsinlər.

```
root@:~ # passwd root  
Changing local password for root  
New Password:  
Retype New Password:
```

### Şəbəkənin quraşdırılması

Siz şəbəkəni DHCP və ya Statik IP ünvanı ilə quraşdırıa bilərsiniz.

Təsəvvür edək ki, bizim dünyada tanınan IP ünvanımız 192.168.121.131-dir. Ancaq qismən real mühitə oxşadılması üçün statik IP ünvanı təyin edək. Həmçinin trafikimizin susmaya görə hansı ötürücü üzərindən keçməsini də müəyyənləşdiririk.

```
ifconfig em0 inet 192.168.121.131 netmask 255.255.255.0 up  
route add default 192.168.121.2
```

Hər halda avtomatik etmək istəsəniz, aşağıdakı əmr kifayət edər.

```
dhclient em0
```

Artıq istənilən SSH client programı vasitəsilə SSH serverimizə qoşulub yükləmə işini görə bilərik. Misal üçün, **putty**.



A screenshot of a PuTTY terminal window titled "192.168.121.131 - PuTTY". The window shows a black terminal screen with white text. It starts with "login as: root", followed by "Using keyboard-interactive authentication.", and then "Password for root@:". A green cursor bar is visible at the bottom of the terminal window.

```
login as: root
Using keyboard-interactive authentication.
Password for root@:
```

Girişi uğurla etdikdən sonra **bsdinstall** əmri kifayət edir ki, yükləməyə başlayasınız.

```
root@:~ # bsdinstall
```

Növbəti gələn bütün ardıcılıq adı yüklənmə prosedurudur.

# Boot menyunun açıqlanması və inizializasiya strukturu

Əməliyyat sistemi ilk yükləndikdə şəkildə göründüyü kimi, Boot menyu ekrana çıxır. Menyuda sıralanan hər bir rəqəmin altında hansı işlərin görüldüyünü ardıcılıqla sadalayaq.



1. **Boot FreeBSD** (Susmaya görə sistem yüklenir)
2. **Boot FreeBSD with ACPI disabled**(Advanced Configuration and Power Interface). Adından da göründüyü kimi, bu mühitdə inzibatçının görəcəyi iş azdır.
3. **Boot FreeBSD in Safe Mode** (Sistemi təhlükəsiz mühitdə işə salır, hansı ki,

DMA sonluqlu olur "Direct Memory Access".) IDE və CDROM alətləri üçün cache-ə yazma funksiyası de-aktiv olur. EISA slotlarının yoxlanması sənəlülə olur. "Extended Industry Standard Architecture" IBM məhsuludur.

4. **Boot FreeBSD in single user mode** (Adı istifadəçi rejimi, çox lazımlıdır)
5. **Boot FreeBSD with verbose logging** (Sistem yüksəldikdə bütün fiziki alətlər və ünvanları çap edilir.)
6. **Escape to loader prompt** (Boot yükleyici rejiminə keçid üçün istifadə edilir və seçdikdən sonra "OK" başlıqlı console açılır.)
7. **Reboot** (Adından da göründüyü kimi, yenidən yüklenmədir.)

Öncə daha geniş olan 6-cı menyunu tam açıqlayaq. Yəni siz sistem yüklenməsi zamanı menyuda 6 düyməsini sıxsanız, aşağıdakı şəkli görmüş olacaqsınız:

```
Type '?' for a list of commands, 'help' for more detailed help.  
OK _
```

<b>?</b>	- Bütün menyunu çap edir.
<b>boot</b>	- Sistemi yükleyir. (-s opsiyası ilə single user rejimdə yükleyir.)
<b>show</b>	- Boot-da olan bütün dəyişənləri çap edir.
<b>set</b>	- Boot-da olan dəyişənləri buradan istifadə eləmək olar. Məs: <b>set acpi_load=NO</b>
<b>unset</b>	- Parametr kimi ötürülen dəyişən, ümumiyyətlə, silinir. Məs: <b>unset acpi_load</b> , sonra " <b>show</b> " ilə baxın.
<b>lsdev</b>	- Boot alətlərini çap edir.
<b>lsmod</b>	- Yüklenən modullarını çap edir.
<b>ls</b>	- Kökü siyahiləyir. Yəni "/"
<b>load</b>	- Seçdiyim kernel və ya modulu yükleyə bilərsiniz.
<b>smap</b>	- BIOS-da alətlərin üzərinə gedən xəritələnməni çap edir.
<b>more, less</b>	- Seçdiyim faylı səhifələyərək çap edir.

Məs: "**more /etc/default/rc.conf**". 'b' - bir səhifə geri, '**spacebar**' və ya 'f' - bir səhifə irəli.

### x86 tipli sistemlərdə BOOT/INIT ardıcılılığı

**BIOS** -> 512 baytlıq "Master Boot Record (MBR)" -> **boot0** -> **boot1** -> **boot2** -> **loader** -> **INIT** (System initialization)

## 1. BIOS

2. Yüklenen diskdə **MBR** (512 baytlıq limit var), ünvanı - '**/boot/boot0**'
3. **Boot1** - yüklenir '**/boot/boot1**' faylından və bslabel-i istifadə edir ki, **boot2**-ni çağırınsın(daxilində yalnız "**boot2**"-ni yüklemək üçün kodlar var) - '**/boot/boot2**' yükleyir - FreeBSD (active) Slicelar-ın yerləşdiyi yeri təyin edir.
4. **Boot2** - yüklenir '**/boot/boot2**' faylından - yükleyir **UFS[1 və ya 2]** - **FS** (File system) '/-in yüklenməsinə dəstək verir və '/boot/loader' -i çağırır.
5. **Loader** - yüklenir '**/boot/loader**' faylından - kernel yüklenir, tərcüməçi işini görür, başlanğıc mühiti formalaşdırır (**INIT**).
6. **INIT** - '**/sbin/init**' - Bütün Unix və Unix tipli sistemlərdə ilk prosesdir. 1 Process ID ilə (PID=1).

a. **Qeyd:** Init strukturu boot prosesinin son mərhələsidir. Init strukturunu siz boot rejimindən çağırı bilərsiniz. **BSD**, **INIT** göstərilən yüklənmə səviyyələrini dəstəkləyir:

- a1. **0** - **shutdown** (halt – sistemi söndürür).
- a2. **1** - Single user mode (Susmaya görə olan rejimə qayıdış üçün '**exit**' və ya "**Ctrl+D**" əmrini daxil edin). Bu halda '**init**', **/etc/rc startup** scriptini sistemi tez işə salmaq üçün yerinə yetirir və disklerin yoxlanışını etmir.
- a3. **6** - Sistemə **reboot** (Yeni yenidənyüklənmə).
- a4. **c** - Ardıcıl giriş etmək istəyənləri bloklayır,tutur. (**Bug** var, sonuncu 3 buraxılışın heç birində işləmir).
- a5. **q** - Yeni informasiya üçün '**/etc/ttys**' faylini təkrar yoxlayır.

**Qeyd:** Əgər siz server otağının 24 saat keşiyini çəkən gözətçilərə inanmırınzsa, o halda '**/etc/ttys**' faylında **console tty**-na aid olan 'secure'-i dəyişib aşağıdakı kimi '**insecure**' edin. Bu, onun üçündür ki, əgər kimse gizli şəkildə server otağına daxil olub, serveri **reboot** edib **root** istifadəçi şifrəsini sindirmaq istəsə, ondan istənilən halda **root** şifré tələb ediləcək.

Single user rejimi ilə daxil olan şəxslərin **root** istifadəçi şifrəsini dəyişmək istəyinin qarşısını aşağıdakı addımlarla alırıq.

**/etc/ttys** terminal quraşdırımları yerləşən fayla daxil oluruq və aşağıdakı sətri uyğun şəkildə edib, yadda saxlayaraq fayldan çıxırıq:

**console none**

**unknown off insecure**

b. Sonra INIT çağrıır: '/etc/default/rc.conf' && '/etc/rc.conf' - Resource Configuration (rc-dən), 'rc' proses

b1. '/etc/rc.conf'-da sistem açılında debug etmek üçün: **rc\_debug="YES"** sətrini '/etc/rc.conf'-a əlavə edirik.

c. '/etc/ttys' - daxilində '**getty**' (terminal alətlərinin siyahısı)-lər saxlayır, hansı ki, istifadəçi: '**login**' olduqda onu autentifikasiya edir və əgər uğurludursa, ona **\$SHELL** verir. Misal üçün, ilk virtual aləti açıqlayayaq.

**cat /etc/ttys | grep ttv0** - Əmr ilk terminal sətrini çap edir.

**ttv0 "/usr/libexec/getty Pc" cons25 on secure**

**ttv0**

- Console aləti deməkdir, sistemimizdə olan ilk virtual terminaldır.

**"/usr/libexec/getty Pc"**

- Bu sütun isə programdır, hansı ki, login müraciətin cavabına terminal yaradır. FreeBSD bunun üçün "getty" istifadə edir.

**cons25**

- Bu isə terminal tipidir. FreeBSD 80-simvolluq terminal tipindən 25-simvolu istifadə edir (vt100 terminalın digər tipidir). Uzaq istifadəçi, server programlarında isə onların özlərinin dəstəklədiyi Pseudo-Terminal tipləri yaradılır.

**on**

- Bu, o deməkdir ki, terminala qoşulmaq icazəlidir, yoxsa yox. (on - aktiv, off - passiv)

**secure**

- Bu, o deməkdir ki, console-da bu ttv0 seansi yalnız "**root**"-a malikdir.

**Qeyd:** Əgər /etc/ttys faylında dəyişiklik etsək, o ya "**init q**" ilə həmin an işləyəcək, ya da sistemə **reboot** etmək lazım olacaq.

**ttv0 "/usr/libexec/getty std.9600" dialup off secure**

- Serial console-u işletmək üçün "off"-u "on" edirik. (COM1-COM4 arası qoşulma mümkün olur).

- d. '/etc/rc.d' - Şəbəkə və service scriptlərinin yerləşdiyi ünvandır.
- d1. '/etc/rc.d'-də yerləşən böyük hərfli skriptlər prioritet baxımından kiçik hərfli lərdən üstündür (Lazımı dəstəyi vermek üçün).
- e. '/usr/local/etc/rc.d' - System Startup və Shutdown bu ünvandakı skriptləri yoxlayır.
- e1. Susmaya görə olan ünvanı dəyişmək istəsək, "/etc/rc.conf"-a əlavə edirik -> **local\_startup="/path/to/startup\_dir"**

<b>kenv</b>	- kernel-də dəyişiklik etmək üçün istifadə olunur.
<b>kenv   grep kernelname</b>	- kernel yüklediyi fayl ünvanını çap edir.

Əgər kerneldə boot dəyişiklik etmək istəsək, sistemə **reboot** edib, 6-nı seçirik "**Enter to loader prompt**"

Susmaya görə olan boot quraşdırma faylı '**/boot/defaults/loader.conf**'-dur və siz özünüzə lazım olan boot imkanlarının sistem yüklenməsində işə düşməsini istəsəniz, '**/boot/defaults/loader.conf**' faylından lazım olanları seçib **/boot/loader.conf** faylına əlavə etməlisiniz.

- 1.'**autoboot\_delay="10"**' - Sistem boot olanda 10 saniyə istifadə olunan vaxt.
- 2.'**boot\_verbose="YES"**' - Sistem açılanda əlavə bütün alətlərin işə salınması console-a çap olunur.
- 3.'**loader\_logo="beastie"**' - Boot-da çıxan şəkli rəngli formada çap edir.
- 4.'**beastie\_disable="YES"**' - Sistem yüklenmə zamanı boot menyusunu silir.

**Qeyd:** Sistem yüklenəndə müəyyən bir screen-in ekrana çıxarılmasını istəsək, aşağıdakı şəkildə edirik.

Misal üçün, **/boot/loader.conf** yüklenmə faylına aşağıdakı sətirləri əlavə etsək, sistem yüklenməsini qrafik şəkildə görə bilərik.

**Qeyd:** Ancaq .bmp genişlənməli şəkillər dəstəklənir.

<b>vesa_load="YES"</b>	- Şəkil genişlənməsi <b>1024x768</b> dəstəklənir.
<b>splash_bmp_load="YES"</b>	- Ekran Screen-i kimi <b> bmp</b> yüklenilməsini aktiv edirik.
<b>bitmap_load="YES"</b>	- BitMap şəkil yüklenməsi modulunu aktiv edirik.
<b>bitmap_name="/boot/splash.bmp"</b>	- Seçdiyimiz şəkin ünvanı, " <b>/boot</b> " qovluğununda yerləşməsi önemlidir.

## Boot nüsxəsinin yoxlanışı və dəyişdirilməsi

**boot0cfg -v da0**

- **da0** sərt diskimizin **boot0** yükleyicisini oxuyuruq.  
Qeyd: Bu, **Standart Boot Manager** yox,  
**FreeBSD Boot Manager** üçün keçərlidir.

#	flag	start chs	type	end chs	offset	size
1	0x80	0: 1: 1	0xa5	521:254:63	63	8385867

```
version=2.0  drive=0x80  mask=0xf  ticks=182  bell=# (0x23)
options=packet,update,nosetdrv
volume serial ID a8a8-a8a8
default_selection=F1 (Slice 1)
```

**1** - 1-ci slice-da yalnız boot var.

**0x80** - 1-ci diskdə yalnız boot var.

**start chs** - Slice start olunub, cylinder 0-da, head 0-da və sector 0-da.

**type** - Slice-in tipi '0xa5'-dir.

**end chs** - Slice bitib cylinder 521-də, head 254-də və sector 63-də.

**drive** - Versiyadan sonra göstərir ki, BIOS deyəcək ki, boot-u birinci hard diskdən "0x80" götür.

**mask** - 4 disk '0xf' slice-i aktivdir.

**ticks** - Və boot timeout-u 182 tick-dir, yəni 10 saniyəyə yaxın.

**bell** - Tickers-ə, yəni saniyələrə siqnal ötürmek üçün istifadə olunur.

**options** - "**packet**" diski **I/O**-üstündür, "**update**" master boot recoreda boot manager tərəfindən yenilənməyə icazə verir, "**nosetdrv**" və diskə məcbur etmir ki, işlək disk üzərinə '**-d**' opsiyasını yollasın.

**Qeyd:** Boot record üzərində dəyişiklik edən zaman sistem, ümumiyyətlə, açılmağa bilər, problemin həlli yolu FreeBSD-nin öz diskidir, diskdən boot recordu yeniləmək olar.

**boot0cfg -B da0**

- '**/boot/boot0**'-ı yükleyin, slice table-da dəyişiklik yoxdur.

**boot0cfg -b /root/boot0 da0**

- '**/root/boot0**'-da seçdiyimiz ünvana boot manager faylini yazın və da0 diskinə qəbul etdirin.

```
boot0cfg -t 364 da0           - 'da0' diskində 't' timeoutu 364 ticks və ya 20 saniyə etdik.
boot0cfg -v -s 2 da0          - 'da0' diskində 2-ci '-s' slice-i susmaya görə olan slice təyin edin.
boot0cfg -v -d 0x81 da0        - 'da0' diskini üçün boot loaderi ikinci '0x81' sərt diskdə axtarın.
fdisk da9                     - Bütün disk slice-lar haqqında informasiya verəndə görəcəyik ki, boot-da neçə əməliyyat sistemi var.
```

### FreeBSD GRUB boot loader

FreeBSD əməliyyat sisteminə susmaya görə olmayan GRUB boot yükleyicisini də yükleyib quraşdırı bilərik. İndi isə bunu həyata keçirək. Ancaq unutmayın ki, portların yüklenməsi qaydalarını siz 3-cü başlıqda daha ətraflı görəcəksiniz.

```
cd /usr/ports/sysutils/grub      - GRUB boot loaderi yükleyirik. (GRUB yalnız i386 tipli OS-da işləyir.)
cp -Rf /usr/local/share/grub/i386-freebsd/ /boot/grub - Lazımi faylları grub qovluğununa nüsxələyirik.
cd /boot/grub                   - GRUB ünvanına daxil oluruq.
ee menu.lst                    - Menu faylı yaradırıq, hansı ki, nəyin və necə yüklenəcəyini başa salır. Faylin içində FreeBSD və Debian əməliyyat sisteminin boot açılmasını əlavə edirik.
default=0                      - Susmaya görə boot olan 0-ci əməliyyat sistemi FreeBSD olacaq.
timeout 15                     - GRUB susmaya görə ilk əməliyyat sisteminin yüklenməsi 15 saniyədən sonra başlayır.
title FreeBSD                 - Başlıq FreeBSD
rootnoverify (hd0,0)          - İlk hard diskin 1-ci hissəsində yerləşir.
makeactive
chainloader +1

title Debian GNU/Linux
root (hd0,1)
chainloader +1
```

**Qeyd:** Bundan sonra biz sərt diskin boot-nu silib yazmaliyiq. "**sysctl kern.geom.debugflags=16**" bu əmr boot recordu yenidən yazmağımıza şərait yaradır.

**Qeyd:** İşimizi bitirdikdən sonra boot recordu yenidən yazmaliyiq. "**grub-install**" əmri ilə. Unutmayaq ki, səhv olsa, sistem heç qalxmayacaq.

**grub-install /dev/ad0**

- **ad0** diskinin boot recordunu təzələyirik.

**Qeyd:** 'chainloader' istifadə etmək əvəzinə, növbəti yükləmə mərhələsi üçün daha təyinatlı ola bilərik. FreeBSD əməliyyat sistemləri üçün '/boot/loader'-i asılı olmadan da çağırıa bilərik. Linux üçün isə kernelin ünvanını da göstərə bilərik.

```
title FreeBSD
root (hd0,0,a)                                     - FreeBSD '/boot/loader'-i 1-ci HDD-nin slice
                                                       0-dan və partition a-dan götürəcək.
kernel /boot/loader

title Debian GNU/Linux, kernel 2.6.18-4-686
root (hd0,1)
kernel /boot/vmlinuz-2.6.18-4-686 root=/dev/hda2 ro
                                                       - Linux isə boot-u
                                                       birbaşa seçilmiş kerneldən götürəcək, vmlinuz-u
initrd /boot/initrd.img-2.6.18-4-686             - Initial RAM disk.
```

**Qeyd:** Biz GRUB-a sistem yüklenəndə Screen şəkildə mənimşədə bilərik. 16-rəngli **640x480**-in '**xpm**' formatlı şəkil yaradaq. Və gz ilə sıxaq. Yaratıldıqdan sonra da **/boot/grub** qovluğununa nüsxələyirik. Məs: '**/boot/grub/splash.xpm.gz**' sonra aşağıdakı sətri '**menu.lst**' faylinə əlavə edirik.

**splashimage=(hd0,0,a)/boot/grub/splash.xpm.gz**

```
reboot          - Sistemi yenidən yükleyirik.
halt -n         - Söndürəndə SERT diskleri sync etməyin.
halt -p         - Sistemi power off edin. Yəni söndürün.
shutdown -r 12:14 - Sistemi saat 12:14-də reboot edin.
shutdown +10 'Bye!'   - 10 dəqiqədən sonra sistemi dayandırın və userlərə 'bye' mesajı yollayın.
Ctrl+alt+del    - reboot edin.
```

# Fayl sistem strukturu

Başlanğıcda müəyyən nəzəri məlumatları çatdırmaq istərdim, cünki bu məlumatlar ən önemli nöqtələrdir. Fayl sistemdə genişlənmə anlayışı yoxdur və bütün fayllar adı fayl olaraq da görünür. Sadəcə hər bir faylı özümüz uyğun genişlənmə ilə təyin edirik ki, gələcəkdə lazım olduqda, faylin tipini xüsusi əmr istifadə etmədən təyin edə bilək. Fayl sistemdə hər bir qovluqda “.” və “..” qovluqları mövcuddur. Bir nöqtə yerləşdiyim ünvanı, iki nöqtə isə, ondan bir ünvan önə keçidi təyin edir. Beləliklə, fayl sistemdə qovluqlar arası kecid etdikdə həmişə bu nöqtələrə müraciət olunur.

## Hard link(Sərt istinad) və Soft link(Yumşaq istinad)

Soft linklər daha çox windows maşınlarda olan **shortcut** (kəsə və ya asan yol)-a oxşayır. Yönəldirməni əsas faylin adı ilə edir. Hard linklər isə təyinatlı inode-larla edir.

**Qeyd:** Soft linklər müxtəlif tipli fayl sistemlər arasında kecid edə bilir.

**Qeyd:** Hard linklər fayl sistemlər arasında kecid edə bilmir(Hətta UFS-in özündə belə).

**Qeyd:** Inode-lar öz identifikasiyasını fayl sistemdə olan **chunks/blocks**-da göstərilən təyinatdan götürür. Əməliyyat sistemində '**ls -i**' əmri istifadə edərək inode-ları görə bilərsiniz.

Beləliklə, gündəmdə hər kəsi maraqlandıran sual Hard link ilə Soft linkin fərqləri nədən ibarətdir? Bu fərqlər aşağıdakılardır:

1. Hard link-lərin yönəldirilməsi inode-lar, soft linklərin yönəldirilməsi isə adlardır.
2. Hard linklər fayl sistemlər arasında kecid edə bilmir, soft linklər edir. Hər bir fayl sistem inode rəqəm düzülüşlü siyahı təşkil edir, hansı ki, üst-üstə düşür. Yəni iki ayrı-ayrı fayl sistem strukturu ilə yaradılmış sərt disk arasında Hard linki ona görə edə bilməzsiniz ki, inode fayl sistemlər arasında kecid etmək istədikdə, özünü üst-üstə düşən eyni rəqəmlərə görə təyin edə bilməyəcək.

Ümumiyyətlə, fayl sistemdə bütün qovluq strukturunu tam araşdırmaq istəsəniz, aşağıdakı əmrənən istifadə edə bilərsiniz.

## **man 7 hier**

Həmin qovluqlardan bir neçəsinin qismən açıqlamalarına baxaq.

- / - Fayl sistemin özək qovluğudur.
- /bin** - Single-user və multi-user mühitləri üçün fundamental istifadəçi proqramları.
- /boot** - Əməliyyat sistemi qalxdıqda istifadə edilən quraşdırma faylları və proqramlar.
- /cdrom** - Sısmaya görə CDROM üçün təyin edilmiş mount nöqtəsi.
- /compat** - Linux uyğunluğu olan proqram təminatlarının yüklənməsi üçün qovluq.
- /dev** - Əməliyyat sistemində olan bütün fiziki alətlər bu ünvanda yerləşir.
- /dev/ttys** - '/etc/ttys' faylında olan terminal portları(virtual).
- /dev/console** - Sistemin console aləti (Serverə uzaqdan yox, fiziki daxil olduqda istifadə edilən virtual alətdir).
- /dist** - Mount nöqtəsi **sysinstall** tərəfindən istifadə edilir
- /etc** - Sistem quraşdırma faylları və skriptləri.
- /etc/hosts** - Bu fayl DNS clientlərə müraciət edilməzdən önce tərkibində olan IP ünvanlarını tərkibində olan adlara çevirir.
- /etc/localtime** - Bu fayl sistemin "tzdata" əmri ilə yaradılan timedatani saxlayır.  
(Əslində "/usr/share/zoneinfo" ünvanından nüsxələnir.)
- /etc/mtree** - Bu qovluq sistemin sısmaya görə istifadə etdiyi qovluq ağac quruluşudur.
- /etc/pccard\_ether** - Bu script çıxarılabilən şəbəkə kartlarını stop, start etmək üçün istifadə edilir.
- /etc/portsnap.conf** - Portsnap əmri işə düşəndə quraşdırma fayldan oxuyur.
- /etc/resolv.conf** - Bu fayl **sysinstall** və ya **bsdconfig** vasitəsilə DNS clientləri yazış yadda saxladığınız anda yaranır. Fayl DNS adlarının IP ünvanlarına çevrilmesi üçün DNS serverlərin IP ünvanlarını özündə saxlayır. Əməliyyat sistemi DNS adın IP ünvana çevrilmesi üçün önce **/etc/hosts** faylına müraciət edir və cavabı tapmadığı halda **/etc/resolv.conf** DNS quraşdırma faylında yazılmış DNS serverlərinə müraciət edir.

<b>/etc/rpc</b>	- Uzaq kompyuterdə əmrləri yerinə yetirmək üçün istifadə olunur. (Faylda bu xidmətlər və onların port nömrələridir).
<b>/etc/security/</b>	- Qovluqda audit security utilitin quraşdırma faylları var.
<b>/etc/snmpd.config</b>	- İlkin SNMP işlənməsi.
<b>/etc/termcap</b>	- Terminalın susmaya görə olan quraşdırma faylı. (Dəstəklənən terminal quruluşları)
<b>/etc/ttys</b>	- Terminal inisializasiyası üçün informasiya faylı.
<b>/etc/rc</b>	- Sistem qalxdıqda işə düşəcək scripti-dir, hansı ki, sistem tərəfindən ona 'autoboot' parametri ötürünləndə sistemi sürətlə işə salır.
 <b>/lib</b>	 - Kritik sistem kitabxanaları (/bin v. /sbin-də olan binar fayllar üçün tələb edilir).
<b>/libexec</b>	- Kritik sistem utilitləri(/bin v. /sbin-də olan binar fayllar üçün tələb edilir).
<b>/media</b>	- CD-lər, USB-lər və Floppy disklerin mount edilməsi üçün mount ediləcək alt qovluqları özündə təşkil edir.
<b>/mnt</b>	- Müvəqqəti mount edilməsi üçün ünvan.
<b>/proc</b>	- Proseslər üçün fayl sistem <b>procfs(5)</b>
<b>/rescue</b>	- Fövqaladə bərpa üçün statik link edilmiş programlar <b>rescue(8)</b> .
<b>/root</b>	- Root istifadəçisinin ev qovluğu
<b>/sbin</b>	- Single-user və multi-user mühitləri üçün fundamental sistem utilitləri və inzibatçı programları.
<b>/tmp</b>	- Sistemin yenidənyüklənməsinədək qalan müvəqqəti fayllar
<b>/usr</b>	- İstifadəçi utilitləri və programlarının əksəriyyəti burada olur.
<b>/var</b>	- Çoxməqsədli jurnal, müvəqqəti, yeridəyişdirilən və növbə faylları burada olur.
<b>/var/run/utx.active</b>	- Sistemdə olan hal-hazırkı istifadəçilər bura yazılır. (Kodlaşdırılmış formada)
<b>/var/log/utx.log</b>	- Bütün sistemə daxil olma və çıkışlar bura yazılır.

# İstifadə ediləcək başlanğıc əmrlər

Oxucu kitabı ardıcıl oxuduğunda hər bir yeni gördüyü əmrin haqqında detallı məlumat almaq üçün artıq sistemdə olan effektiv utilitlərdən istifadə edə bilər. Bunun üçün başlanğıc və gələcəkdə lazım olacaq ən önəmlı əmrlər haqqında öncədən açıqlamalar verək ki, kitabın oxunması asanlaşsun.

Sistem yüklənən kimi, əgər minimal deyilsə, özündə bütün əmrlərin manualını da yükleyir. Bu əmrlərin manuallarının səliqəli axtarış metodikası mövcuddur, hansı ki, öyrəndikdən sonra hansısa əmrin axtarışı bizim üçün çox asanlaşacaq.

**apropos crontab**

- 'crontab' əmri üçün sistemdə olan bütün 'man' səhifələrin siyahısını çap edəcək.

**whatis cat**

- 'man cat' əmrindən çıxan nəticədən "NAME" bölümünün ilk sətrini çap edir.

Manuallar sistemdə tip fərqlərinə görə 9 hissəyə bölünür və onlar aşağıda açıqlanır.

- 1 - Əsas istifadəçi əmrləri man 1-də olur.
- 2 - Sistem çağrıqlarının manları man 2-də olur.
- 3 - Proqramlaşdırma quruluşu kitabxana funksiyaları man 3-də olur.
- 4 - Alətlər man 4-də olur. (Məs: **man acd**, **man ata**)
- 5 - Quraşdırma faylları və fayl formatları man 5-də olur.
- 6 - Oyunların manualları man 6-da olur.
- 7 - Müxtəlif tipli manlar man 7-də olur.
- 8 - Administrativ əmrlər və daemonlar haqda məlumatlar man 8-də olur.
- 9 - Kernel programçıları üçün manlar man 9-da olur. Məs: **man accept\_filter**

<b>man</b>	- Manual deməkdir. Başqa sözlə desək, əmrlərin açıqlanması.
<b>man -a crontab</b>	- 'crontab' əmrinə aid olan bütün manları çap edəcək.
<b>man 5 crontab</b>	- 5 nömrəli manualların içindən 'crontab' manualını çap edəcək.
<b>man mount -P more</b>	- 'man' önce 'mount' əmri üçün, ardınca da 'more' əmri üçün manualı çap edəcək.
<b>man -f mount</b>	- 'man' burada 'mount' əmrinin bütün manual vərəqlərində olan "NAME" bölümünün ilk sətrini çap edir.
<b>man -k crontab</b>	- 'man', 'crontab'-a aid olan manualları bir sətirdə çap edir.
<b>man hier   col -b &gt; hier.txt</b>	- hier əmrinin səhifəsini filtr edib, plaintext formatında hier.txt faylinə yazır.
<b>col</b>	- Girişdə daxil olan sətirləri filtr edir.
<b>man 7 tuning</b>	- Sistemi tuning etmək üçün müxtəlif imkanları göstəririk. (Ancaq onları tam dərindən araşdırmadan tətbiq etməyin).

**Qeyd:** Bəzi hallar olur ki, developerlər öz manuallarını 'info' səhifələrində yerləşdirirlər. Orada da axtarmaq mümkündür.

<b>info ls</b>	- 'ls' əmri haqqında informasiyanı çap edəcək.
<b>?</b>	- 'info' səhifəsində istifadə edilə biləcək əmrləri çap edir.
<b>L</b>	- Öncə açdığım info səhifəsini çap edin.
<b>n, p, u</b>	- next, previous, up
<b>TAB</b>	- Növbəti manual seçilib ENTER sıxılır.
<b>Q</b>	- info səhifəsindən çıxış.

<b>dmesg</b>	- Sistemə ən son qoşulmuş fiziki aletləri çap edir.
<b>mkdir -p Atesh/Ramiq/Asif/Vugar</b>	- İç-içə rekursiv qovluq yaradılır.
<b>rm -R *</b>	- Diqqət! Yerləşdiyiniz qovluqda hər şeyi siləcək.
<b>du -sh *</b>	- Yerləşdiyiniz ünvanda hər bir qovluğun və ya faylin həcmini göstərəcək.
<b>passwd</b>	- Heç bir arqumentsiz işə salındıqda, <b>root</b> istifadəçisinin şifrəsini dəyişir.
<b>passwd elcin</b>	- elcin adlı istifadəçinin şifrəsini dəyişir.

```
cut -f 2 -d ":" /etc/passwd
```

"**etc/passwd**" faylından "-f" (**field**) parametri ilə 2-ci sütunu, "-d" (**delimiter**) parametri ilə iki nöqtə ilə ayrılan sütunu çap edəcək.

```
locate rc.conf
```

- Fayl və ya qovluq fərqi olmadan sistemdə '**rc.conf**' adı ilə axtarış edəcək.

**Qeyd:** Sistem yükləndikdə susmaya görə "**locate.database**" olmayacaq. Bunun üçün "**/usr/libexec/locate.updatedb**" əmrini yığmaq yetər. Susmaya görə locate, registr(hərfin böyük və ya kiçik yazılmamasına)-a görə dəyişə bilər.  
Ancaq nəzərə alın ki, bunu etmək çox təhlükəlidir, çünki biz öz sistemimizdə olan fayl və qovluqlar haqqında olan bütün məlumatları həmin fayla yazırıq. Bu iş HACKER üçün gözəl bir məlumat olacaq. ☺

```
locate -S
```

- "**locate.database**" haqqında məlumat çap edir.

```
locate -i bash
```

- **register** fərqi olmadan "**bash**" adlı başlıqla axtarış edin.

```
locate -l 5 kernel
```

- **kernel** adı ilə tapılan sətirlərdən yalnız 5-ni çap edin.

```
type ifconfig
```

- '**ifconfig**'-in sistemdə olan **PATH** (tam ünvanını)-ni çap edəcək (yalnız 9 və yuxarı versiyalar).

```
whereis mount
```

- '**mount**' əmri üçün **binar**, **man** və **sourcecode** fayllarını görə bilərik. (Qeyd: portların axtarışında belə bu əmri istifadə etmək olar.)

```
which ls
```

- '**ls**' əmrinin sistemdə olan bütün **PATH** və ya aliasların içində axtaracaq.

```
find /usr/ports/ | grep apache22
```

- '**/usr/ports**' ünvanında axtarış edin və nəticələrin içində '**apache22**'-ni axtarın.

**Qeyd:** Əgər "**find**" əmri adı istifadəçi adından işə salınsa, o, bir çox cavablarda "**Permission Denied**" alacaq.

```
find / -name wlan_wep.ko | grep -v "Permission denied"
```

- "**Permission denied**" sözünü çıxmaq şərtilə "**wlan\_wep.ko**" faylı üçün axtarış edin.

```
find / -name wlan_wep.ko 2> /dev/null
```

- "**Permisson Denied**" çıxan sətirləri boşluğa yollayacaq. Eynilə **wlan\_wep.ko**" faylı üçün axtarış edir.

```
find /usr/bin/ -amin -2
      - "/usr/bin" ünvanından "-amin"(a minute)-lə təyin edir ki, son 2 dəqiqə ərzində istifadə olunan əmr hansıdır.

find /home/cavid/ -atime +60
      - "/home/cavid" ünvanının son 60 gün ərzində toxunulmayan fayl və qovluqları hansılardır.

find /etc -type d -print 2>/dev/null
      - "/etc" qovluğunda tipi "-d" qovluq olanların hamisini çap edin, səhvləri boşluğa yollayın.  
(Yalnız BASH SHELL-də işləyir.)

find /sbin/ -perm 555
      - "/sbin" qovluğunda "-perm" hüququ 555 olan bütün faylları çap edin.

find /var -user cavid -exec ls -l {} \;
      - "/var" qovluğunda "-user"-lə cavid istifadəçi adında olan bütün faylları tapın və onları "ls -l" əmrinə ötürün ki, siyahilayın.

find /var -user salman | xargs ls -l
      - "/var" qovluğunda "-user"-lə salman istifadəçi adında olan bütün faylları tapın və onları xargs-la siyahilayın.

find / ! -group wheel -type f 2>/dev/null | xargs ls -l
      - "/" kök qovluqda olan, "wheel" qrupu adında olanlardan başqa, bütün faylları tapın, çıxan nəticədə səhvlər varsa, onları boşluğa yollayın və nəticəni ekrana çap edin.

find /sbin/ -type f ! -perm o+x | xargs ls -l
      - "/sbin" qovluğunda "-type"-i "f" fayl olan bütün faylları çap edin, "others"-də "exec" hüququ olan faylları çıxməq şərtilə.

find / -xdev -size +10M | xargs ls -lS
      - "/" kök slice-da həcmi 10Mb olan faylları kiçikdən tutmuş böyüyə çap edin. Burada "-xdev" axtarışı kök slice-dan hər yerə təyin edir.

find /etc -type f -exec md5 {} \; 2>/dev/null > /tmp/md5.list
      - "/etc" qovluğun altında olan və tipi "f" olan bütün faylları md5-lə yoxlayın, səhvləri boşluğa yollayın, nəticəni isə "/tmp/md5.list" faylinə yazın.

find /usr/ports -type d | grep quake | less
      - Portların içində 'quake' adı ilə başlayan bütün qovluqları çap edin.

find / -xdev -printf '%h\n' | sort | uni -c | sort -k 1 -n
      - Kiçikdən böyüyədək hər bir qovluğun nə qədər inode tutduğunu çap edir (Linux-da da işləyir).
```

<b>find .   xargs rm</b>	- <b>rm -rf</b> * arqumentinə ötürürlən faylların adı uzun olduqda, o, silə bilmir, bu zaman bizim köməyimizə məhz bu əmr gəlir.
<b>diff testfile testfile2</b>	- İki fayl arasında olan fərqləri çap edir.
<b>head -n 3 /var/log/messages</b>	- Göstərilən faylin ilk üç sətrini çap edir.
<b>tail -n 3 /var/log/messages</b>	- Göstərilən faylin son 3 sətrini çap edir.
<b>mkdir dfg{1,2,3}</b>	- Bir dəfəlik çoxlu qovluq yaradırıq.
<b>cp -R /usr/home/cavid .</b>	- Göstərilən qovluqda olan bütün məlumatları mövcud yerləşdiyiniz ünvana nüsxələyin. Nöqtə işarəsi olduğunuz ünvanın özü deməkdir.
<b>cat /etc/passwd   grep elcin &gt;&gt; asd</b>	<ul style="list-style-type: none"> <li>- <b>passwd</b> faylinin içindən "elcin" sözü olan sətri seçib "<b>asd</b>" faylinin ən sonuna yazacaq.</li> </ul>
<b>cat /etc/passwd   grep elcin &gt; asd</b>	<ul style="list-style-type: none"> <li>- Hər şeyi silib yalnız elcin olan sətri yazacaq.</li> </ul>

**Qeyd:** Əməliyyat sisteminin üzərində istənilən işləyən əmr və ya programın standart girişi, çıxışı və səhvleri var. Bunlar üzərində olan limitlər kernel tərəfindən idarə edilir. Aşağıda onları sadalayıraq.

**STDIN** (Məlumatın daxil olması) - '<' - susmaya görə giriş faylı adlanır. - '**/dev/fd/0**'

**STDOUT** (Məlumatın çıxışı) - '>' || '>>' - **Default = Screen** - '**/dev/fd/1**'

**STDERR** (Standart səhvler) - '**/dev/fd/2**'

**Qeyd:** '.' - hal-hazırkı qovluq.

'..' - bir qovluq geriyə.

<b>jobs</b>	- Arxa fonda işləyən programları göstərir.
<b>jobs -l</b>	- Arxa fonda işləyən prosesin ID-si və işlədən programın adı çap olunur.
<b>fg %1</b>	- foreground-da olan işi arxa fondan önə çıxarırr.
<b>Ctrl+Z</b>	- Suspend rejim
<b>bg %1</b>	- Arxa fonda olan işi çağırıb işə salır.
<b>top</b>	- Sistem prosesləri, sistem DDR-1 və sistem servislərinin proseslərini primitiv şəkildə göstərir.
<b>ldconfig -r</b>	- Sistemdə istifadə olunan bütün kitabxanaları çap edir.
<b>ldd /bin/sh</b>	- Əsas shell-in istifadə elədiyi kitabxanaları çap edir.
<b>ps</b>	- İşləyən prosesləri göstərir.

<b>systat</b>	- Sistem prosessorunun vəziyyətini göstərir.
<b>stat</b>	- Lazimi meta-verilənləri çap edir.
<b>ps -ax   grep ftp</b>	- Sadəcə ftp prosesini görmək üçün.
<b>kill -9 757</b>	- 757-ci id-li prosesi söndürmək üçün (-9 force)
<b>killall ftpd</b>	- ftpd adlı bütün prosesləri öldürür.
<b>"&amp;&amp;"</b>	- Əmr ardıcılılığı üçün istifadə olunur, əgər birinci əmr düzgün yerinə yetirilibsə, ikinci də düzgün yerinə yetiriləcək.
<b>"  "</b>	- Əmr ardıcılığı üçün istifadə olunur, əgər 1-ci işləsə, 2-ci işləməyəcək. Əgər 1-ci işləməsə, 2-ci işləyəcək.
<b>uname -a</b>	- Sistemin versiyasını göstərir.
<b>uptime</b>	- Serverin dayanmadan istifadə müddətini göstərir.
<b>ps awx   grep -v grep   grep v0</b>	- Sistemə ilk daxil olan istifadəçinin <b>tty</b> seansının prosesini axtarıb çap edir.
<b>who</b>	- Sistemdə olan istifadəçiləri göstərir.
<b>whoami</b>	- Hal-hazırda sistemdə olan istifadəçiləri göstərir.
<b>tty</b>	- Hal-hazırkı <b>tty</b> -i çap edir.
<b>w</b>	- Hal-hazırkı <b>tty</b> -in istifadə etdiyi əmrləri çap edir.
<b>ls -ltr</b>	- Son yenilənmə cədvəli şəklində sort edir.
<b>users</b>	- Sistemdə olan istifadəçiləri yalnız istifadəçi adı ilə göstərir.
<b>date</b>	- Sistem tarixini çap edir.
<b>time</b>	- Sistem tarixini vaxt möhürü formasında çap edir (Programlaşdırılmışda istifadə olunur.)
<b>cat /etc/passwd   wc -l</b>	- (söz sayı) <b>passwd</b> faylinin içinde neçə sətr olduğunu göstərir.
<b>cat /etc/passwd   nl   more</b>	- (Rəqəmli sətir) faylin içinde hər sətrin əvvəline rəqəm yazır.
<b>nl</b>	- Sətirləri sayırlar.
<b>sort</b>	- İstənilən ünvan və ya faylı sort edir.
<b>sort /etc/passwd -t":" -k3 -n</b>	- "/etc/passwd" faylında UID-ə görə kiçikdən böyüye doğru sort edək.

"-t" ayırıcı ':' olan,  
"-k3" axtarış 3-cü sütuna görə aparılsın,  
"-n" sorting rəqəmlər arasında gedir, hərflər yox.

**strings /bin/ls | grep -i libc** - "strings" əmri "/bin/ls" əmrini açıb aydın şəkildə oxuyur, "grep" isə '-i' opsiyası ilə yazıya hissəyyatlılığı söndürür, "libc"-ni axtarış edir.

**mv** - Faylin adını və ya ünvanını dəyişmək üçün istifadə olunur.

**touch** - Fayl yaratmaq üçün istifadə olunur.

# BÖLÜM 2

## Symlink, SETUID, SETGID, fayl flaqları, resurslar və proseslər, rezerv nüsxələr, SUDO

- / Fayl tipləri, yetkilər, symlinklər
- / SETUID, SETGID, Sticky Bit
- / FreeBSD fayl flaqları, simvolik linklər, yeni diskin və USB Flash-in əməliyyat sisteminə əlavə edilməsi, UNIX Tape Drive.
- / Sistem resursları və proseslər, faylların aktivliyinin təyin edilməsi
- / İslək proseslərin yoxlanılması və idarə edilməsi
- / Rezerv nüsxələr və onların bərpa edilməsi, istifadəçilərin SUDO ilə məhdudlaşdırılması

Başlığımızda fayl sistemimizdə olan fayl və qovluqlar üzərində olan yetkilərin idarə edilməsi haqqında danışılır. Sistemə yeni disk əlavə edib istifadə etmək qaydası, proseslərlə işləmək üsulları, rezerv nüsxələr və onların bərpa edilməsi, istifadəçilərə fərqli ardıcılıqla sistem əmrlərindən yararlanmaq üçün yetkilərin təyin edilməsi açıqlanır.

# Fayl tipləri, yetkilər, symlinklər və SetUID/SetGID/StickyBIT

Bu başlıqda fayl tipləri, onların fərqləndirilməsi qaydaları, fayllar/qovluqlar üzərində olan yetkilərin idarə edilməsi, fayllar üzərində olan üzvlüyün idarə edilməsi, symlinklərin təcrübədə istifadə edilməsi və spesifik bitlərin təyinatları açıqlanacaq.

Fayl tiplərini, yazılıma ardıcılığını açıqlayacaq:

1. ardıcılıq 10-bit təşkil edir:
  - a. Fayl tipi – 1 bit istifadə edir
  - b. İstifadəçi yetkiləri – 3 bit
  - c. Qrup yetkiləri – 3 bit
  - d. Other/Everyone(Digərlər və ya hər kəs) permissions – 3 bit

*Misal:*

```
-rw-r--r-- 1 cavid cavid 113788806 2014-09-23 16:48 test.txt
```

**bit 0** = fayl tipi sıfırdır(Yeni sıralamada olan ilk bit):

- a. Fayl - (-)

- b. Qovluq - (d)
- c. Link - (l) > symlink deməkdir
- d. Character device - (c) - /dev > hansısa fiziki qoşulmuş qurğu
- e. Block device - (b) - /dev > block tipli qurğu

**bit 1,2,3** = istifadəçi yetkiləri

- bit 1 = read (r) oxumaq və ya oxumamaq yetkisi(-)
- bit 2 = write (w) yazmaq və ya yazmamaq yetkisi (-)
- bit 3 = execute (x) yerinə yetirilən olsun və ya olmasın (-)

**bit 4,5,6** = qrupların yetkiləri

- bit 1 = read (r) oxumaq və ya oxumamaq yetkisi(-)
- bit 2 = write (w) yazmaq və ya yazmamaq yetkisi (-)
- bit 3 = execute (x) yerinə yetirilən olsun və ya olmasın (-)

**bit 7,8,9** = hər kəsin yetkiləri

- bit 1 = read (r) oxumaq və ya oxumamaq yetkisi(-)
- bit 2 = write (w) yazmaq və ya yazmamaq yetkisi (-)
- bit 3 = execute (x) yerinə yetirilən olsun və ya olmasın (-)

r = 4

w = 2

x = 1

Tam yetki rəqəmi = 7-dir.

**umask** əmrinin nəticəsi obyekt yetkisinin tam yetkidən çıxılmasından alınır.

**Default umask = 0022**

0777 - 0022 = 755 – Bu yetki adətən qovluqlara verilir.

**Qeyd:** Yerinə yetirmə yetkisi istifadəçilərin qovluqları aça bilmələri üçün təyin edilir. Misal üçün:

- a. (rwx) yaradan üçün
- b. (r-x) qrup üçün
- c. (r-x) hər kəs üçün

**Qeyd:** Fayllar susmaya görə **0644** yetkisinə sahib olurlar:

- a. (**rw**) yaradan tərəfindən
- b. (**r**) qrup tərəfindən
- c. (**r**) hər kəs tərəfindən

**chmod** - sistem obyektlərinin yetkisinin idarə etməsi üçün utilitidir:

**Qeyd:** Dəyişiklik üçün ya faylı yaradan istifadəçi olmalıdır, ya da SuperUser ('**root**')

Misal: Fayl üzərində bir neçə test aparaq. Məqsədimiz **Cavid** adlı istifadəçinin test.txt faylinə olan yetkisini tam almaqdır.

**chmod 000 test.txt**

**Qeyd:** '**root**' - istifadəçisi SuperUser (Sistem İnzibatçısı istifadəçi adı) istənilən ünvana tam yetkilidir.

'cavid' adlı istifadəçinin '**test.txt**' faylinə yalnız oxumaq və yazmaq yetkisi tələb edilsə, aşağıdakı əmrden istifadə etməliyik.

**chmod 600 test.txt**

Növbəti əmrlə isə '**test.txt**' faylinə bütün istifadəçi və bütün qruplar üçün maksimal yetki təyin etmiş oluruz.

**chmod 777 test.txt**

Həmçinin yetkiləri '**chmod**' əmri ilə hərflərlə də təyin etmək mümkündür. Ardıcıl olaraq bir neçə misalımızda nümayiş etdirək.

**chmod o+x test.txt**

- Hər kəs üçün executable (yerinə yetirilmə) yetkisi təyin edirik ('**o**' -> **everyone** deməkdir).

**chmod u+x test.txt**

- Faylin yaradıcısı üçün executable yetkisi təyin edirik ('**u**' istifadəçi deməkdir).

**chmod gou+x test.txt**

- '**test.txt**' faylinin istifadəçisi, qrupu və hər kəs üçün "executable" yetkisi təyin edirik.

**chmod gou-x test.txt**

- '**test.txt**' faylinin istifadəçisi, qrupu və hər kəs üçün "executable" yetkisini silirik.

**chmod a+xrw test.txt**

- '**test.txt**' faylinə hər kəs üçün tam yetki veririk. "-a" all deməkdir.

**chmod a-xrw test.txt**

- '**test.txt**' faylından yetkini hər kəs üçün tam silirik.

Fayl sistem obyektlərində olan üzvlüyü '**chown**' əmri ilə dəyişmək olur. Bu əmrlə həm qrup, həm də istifadəçiye aid olan üzvlüyü idarə etmək mümkündür. Ancaq təkcə qrupların üzvlüyünü idarə etmək istəsəniz, bunun üçün '**chgrp**' əmri mövcuddur. Aşağıdakı misallarımızda biz bu əmrlərin hər birini praktikada nümayiş edəcəyik:

<b>chown root test.txt</b>	- 'test.txt' faylinin istifadəçi sahibini 'root' istifadəçisi edirik.
<b>chown -R cavid:cavid temp/</b>	- Bu əmrlə biz <b>rekursiv</b> olaraq temp qovluğunda olan bütün qovluqlar/fayllar üçün yetkini " <b>cavid</b> " adlı istifadəçi və " <b>cavid</b> " adlı qrupa təyin edirik.
<b>mkdir -m 700 /tmp/new2</b>	- Əmr avtomatik olaraq '/tmp/new2/' qovluğununu yaratdıqda həmin qovluğa " <b>chmod 700</b> " yetkisini təyin edir.
<b>chgrp wheel file</b>	- File faylini <b>wheel</b> qrupunun üzvü edirik.
<b>chgrp -R wheel folder</b>	- Folder adlı qovluq və onun daxilində olan istənilən fayllar və qovluqlar rekursiv olaraq, <b>wheel</b> qrupunun üzvü olacaq.

# **SETUID, SETGID, Sticky Bit**

SETUID, SETGID - Əməliyyat sistemində qadağan olmuş əmrləri adı istifadəçi hiss etmədən "**root**" istifadəçi adından işə sala bilir. Məs: "**passwd**" əmri istifadəçi tərəfindən işləyərkən o, 'root' istifadəçi adının **UID**-ni istifadə edəcək.

Buna **EUID** və **EGID** deyilir. 'E' simvolu "**Effective**" deməkdir.

**UID** - prosesi işə salan istifadəçinin identifikasiatorudur.

**EUID** – istifadəçinin realda işləyən proses ID-sidir.

Məs: "**passwd**" əmri adı istifadəçi tərəfindən işə salınanda ancaq öz UID-i altında işləyir, şifrə yeridiləndə bazaya **EUID** adından ötürülür. Bu, o deməkdir ki, şifrə bazaya "**root**" user ID tərəfindən yeridilir. Yəni hiss etmədən müvəqqəti "**root**" yetkisi əldə edir.

Faylda **SUID** identifikasiatoru "4" rəqəmi olur, Group identifikasiatoru isə "2" rəqəmi olur.

Eyni məntiq həm də **SETGID**-ə aiddir, fərqli ondan ibarətdir ki, **SETUID**-də yalnız istifadəçiyə, **SETGID**-də isə qrupa təyin etmək olur.

Test üçün iki console açaq. Birində "**user**" adlı adı istifadəçi ilə, digərində isə "**root**" adlı istifadəçi ilə giriş edək.

1. "user" adlı istifadəçinin console-una "passwd" əmrini daxil edib, "root" istifadəçisinin console-na keçid alaq.

2. "root" adlı istifadəçinin console-una, "ps -aux | grep passwd" əmrini daxil etsək, görəcəyik ki, "passwd" əmri "root" istifadəçi adından işə salınıb.

**Qeyd:** Unutmayın ki, istifadəçiye hər bir halda **SETUID** və ya **SETGID** təyin olunsa belə, o, digər istifadəçinin şifrəsini dəyişə bilməz. Bu sistemin dəyişməz siyasetidir.

**Qeyd:** Bir fayla həm **SETGID**, həm də **SETUID** verilməsi təhlükəlidir, mütləq **Sticky bit** istifadə edin.

**Qeyd:** Əgər /etc/fstab faylında hansısa fayl sistem slice-i üçün "**nosuid**" yetkisi təyin olunubsa, **SETUID** və **SETGID** işləməyəcək. Ancaq onun üstündən keçmək asandır, yəni əslində elə də mənası yoxdur.

**chmod 4777 file** - İstifadəçi **SETUID**-i "s" görünəcək, file adlı fayla **SETUID**-i təyin etdik.  
**chmod 2777 file** - Qrup **SETGID**-i "s" görünəcək, file adlı fayla **SETGID**-i təyin etdik.

**SETUID** və **SETGID** təyin etdiyimiz faylin görüntüsü aşağıdakı şəkildə olacaq:

-rwsrwsrwx 1 root wheel 0 Mar 9 11:14 file\*

**Qeyd:** Hal-hazırda bütün UNIX və Linux sistemlərində susmaya görə **SETUID** və **SETGID** imkanlarının skriptlərlə işləməsinin qarşısı alınmışdır. Həmin skripti istifadə etmək üçün gərək C-də Compile edilmiş programdan çağırısanız. Aşağıda bunun üçün bir misal çəkə bilərik.

/root qovluğunun altında /root/script.sh adlı skript yaradıraq və içində aşağıdakı sətirləri əlavə edirik.

**#!/bin/sh**

```
# Deyirik ki, '/var/log/maillog'-u təmizləsin.  
cat '/dev/null' > /var/log/maillog
```

**chown root:wheel /root/script.sh**

- Faylin üzvlüyünü **root** istifadəçisinə və **wheel** qrupuna mənimşədirik.

**chmod 4755 /root/script.sh**

- Həmin fayla istifadəçi üçün **SETUID** yetkisini veririk. Eyni ilə bu istifadəçi üçün **read**, **write**, **exec** yetkisi var. Ancaq qruplar və digərləri üçün yalnız **read**, **exec** yetkisi var.

Bu, əslində o deməkdir ki, skripti hər kəs işə sala bilər. Amma işləməyəcək. Yoxlamaq üçün adı istifadəçi adından '**/root**' qovluğuna daxil olub script.sh faylini işə salmağa çalışın.

```
> ./script.sh  
cannot create /var/log/maillog: Permission denied
```

- Cavab aydınlaşdır. Bunu açmaq üçün gərək biz C-də bir program yaradıb ona əmr verək ki, bu skripti yerinə yetirsin.

Aşağıdakı ardıcılıqla bunu edə bilərik. **runscript.c** adlı skript faylı yaradıb içini göstərilən C kodları əlavə edirik və sonra həmin faylı kompilyasiya edirik. Kompilyasiya etdikdən sonra isə çıxışda bir program emələ geləcək, hansı ki, o program işə salındıqda '**/root/script.sh**' skriptinə **SETUID** sıfır (0) təyin edərək işə salır.

```
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int main()  
{  
    setuid( 0 );  
    system( "/root/script.sh" );  
  
    return 0;  
}
```

```
gcc runscript.c -o runscript
```

- C kodlu yaratdığımız faylı kompilyasiya edib '**runscript**' adlı program yaradırıq.

```
chown root:wheel runscript
```

- Sonra da '**runscript**' programını '**root**' istifadəçisinin və '**wheel**' qrupunun üzvü edirik.

```
chmod 4755 runscript
```

- '**runscript**' programına **SETUID** təyin edirik və '**root**' özü programı **read**, **write**, **exec** edə bilər, qruplar və digərləri isə ancaq **read**, **exec** edə bilər.

Test üçün adı istifadəçi adından sistemə daxil olun və C-də olan **runschript** programını işə salın.

```
> cd /root  
> ./runschript
```

- Nəticə uğurlu olacaq, yəni işləyəcək. Sonra isə '**/var/log/maillog**' faylini yoxlayın (fayl təmiz olmalıdır, yəni boş).

Biz yuxarıda sistem tərəfindən qadağan olunmuş '**/var/log/maillog**' faylini adı istifadəçi adından təmizləyə bildik. Sözsüz ki, '**root**', yəni SuperUser-in icazə verməsi ilə mümkün oldu.

## Sticky bit

İmkanlar:

Bütün istifadəçilərin bir-birlərinin fayllarını silmədən eyni qovluğu istifadə etmək imkanı '**chmod**' - əmrini istifadə edərkən **4** ardıcıl rəqəm istifadə olunur (4-aparıcı səkkiz mənalı rəqəm). OS və Filesystem təminat verir ki, yalnız faylı yaradan istifadəçi və '**root**' adlı istifadəçi faylı silmə bacarığına malik olsun.

İşimiz:

**/kursfolder** qovluğuna sticky və **SETGID** bit yetkisi verək.  
**chmod 3777 /kursfolder**

Qeyd: **/tmp** bir çox unix-lərdə artıq **sticky** kimi flaglanmış olur.

```
drwxrwxrwt    7 root  wheel      512 Jan 29 13:12 tmp
```

**chmod 1777 file** - sticky "t" simvolu ilə görünəcək  
t - sticky bit  
s - SETUID, SETGID

# FreeBSD fayl flagları, simvolik linklər, yeni diskin və USB Flash-in əməliyyat sisteminiə əlavə edilməsi

## Fayl flagları

FreeBSD əməliyyat sistemində fayllara müxtəlif flaglar təyin etmək olar.

Məs: Sərt şəkildə sistemə deyirik ki, faylı silmək olmaz.

**chflags sunlink test**

- "test" faylına silmək olmaz flagı təyin edirik.  
Hətta **root** belə sile bilməyəcək.

**chflags nosunlink test**

- "test" faylından silmək olmaz flagını silirik.

**chflags uunlink /home/cavid/file**

- Root istifadəçisi adından '**cavid**' adlı istifadəçinin yaratdığı '**file**' adlı fayla (**User Undelete**) flag təyin edirik. Yeni **cavid** bu faylin yaradıcısı olduğu halda da sile bilməyəcək.

**chflags nodump file**

- "**file**" adlı fayla nodump flagı qoyuruq ki,

**dump** edəndə onu götürməsin.

**chflags schg /boot/kernel/kernel**

- Kernel file-na flag təyin edirik ki, kerneli heç kəs

tərəfindən dəyişmək mümkün olmasın. Hətta

**root** belə dəyişə bilməsin.

```
chflags -R schg /bin           - "/bin" qovluğuna dəyişməz flag təyin edirik  

                                (dəyişiklik mümkün deyil).  

chflags noschg /boot/kernel/kernel - Təyin etdiyimiz flagı geri götürürük.  

ls -lo test                  - Bu əmr sayəsində faylda flag olduğunu  

                                görmək mümkündür.
```

## Simvolik linklər

Simvolik linklərin nəzəri hissəsinə 1-ci başlığımızda açıqlamışdıq. Bu başlıqdə isə sadəcə praktik hissələri göstərəcəyik. Aşağıda ardıcıl olaraq, birinci soft(yumşaq) link və sonra hard(sərt) link misallarını sadalayıraq.

<b>ln -s /home/user/linkfile</b>	- Yerleşdiyimiz qovluğun içində <b>/home/user</b> qovluğunda olan <b>linkfile</b> faylini eyni adla soft link edirik.
<b>ln -s /home/user/linkfile /root/newlink</b>	- Eyni ilə öncəki sətirdə göstərildiyi kimi, <b>/home/user/linkfile</b> faylini <b>/root/newlink</b> ünvanına soft link edirik.
<b>ln file file-hardlink</b>	- "file" faylinə " <b>file-hardlink</b> " adlı sərt link təyin edirik. Ancaq unutmayın ki, hard link müxtəlif fayl sistemlər arasında işləmir.

## Test şərti operatoru

Fayl sistemdə faylların tipini təyin etmək üçün əvəzolunmaz alətdir.

<b>-b file</b>	- Yoxlayın, bu, block tipli alətdirmi?
<b>-c file</b>	- Yoxlayın, simvol tipli alətdirmi?
<b>-d file</b>	- Yoxlayın, qovluqdurmۇ?
<b>-e file</b>	- Yoxlayın, bu fayl varmı?
<b>-f file</b>	- Yoxlayın, fayl varmı? Məs: Qovluq deyil, socket deyil, pipe deyil, link deyil, alət deyil.
<b>-g file</b>	- Yoxlayın, faylin GID-i verilən giddirmi?
<b>-h file</b>	- Yoxlayın, fayl simvolik linkdirmi? "-L"-lə eynidirmi?
<b>-k file</b>	- Yoxlayın, faylda sticky bit varmı?
<b>-L file</b>	- Yoxlayın, fayl simvolik linkdirmi? "-h"-lə eynidirmi?
<b>-s file</b>	- Yoxlayın, fayl mövcuddurmu və həcmi 0 baytdan böyükdürmü?

<b>-s file</b>	- Yoxlayın, fayl mövcuddurmu və socketdirmi?
<b>-t file</b>	- Yoxlayın, deskriptor terminala qoşulubmu?
<b>-u file</b>	- Yoxlayın, faylin ID-si, User ID ilə eynidirmi?
<b>-w file</b>	- Yoxlayın, fayla yazma yetkisi var, yoxsa yoxdur?
<b>-x file</b>	- Yoxlayın, fayl yerinə yetiriləndir, yoxsa deyil?
<b>-z string</b>	- Yoxlayın, göndərilən sətrin həcmi 0 baytdan böyükdürmü?
<b>expr1 -a expr2</b>	- Yoxlayın, hər iki söz doğrudurmۇ?
<b>expr1 -o expr2</b>	- Yoxlayın, hər iki sözdən heç olmasa biri doğrudurmۇ?
<b>file1 -nt file2</b>	- Yoxlayın, file1 file2-dən yenidirmi?
<b>file1 -ot file2</b>	- Yoxlayın, file1 file2-dən köhnədirmi?
<b>file1 -ef file2</b>	- Yoxlayın, bu iki fayl arasında link varmı?
<b>var1 = var2</b>	- Yoxlayın, 1-ci dəyişən ikinciye bərabərdirmi?
<b>var1 -eq var2</b>	- Yoxlayın, 1-ci dəyişən ikinciye bərabərdirmi?
<b>var1 -ge var2</b>	- Yoxlayın, 1-ci dəyişən 2-cidən böyük və ya ona bərabərdirmi?
<b>var1 -gt var2</b>	- Yoxlayın, 1-ci dəyişən 2-cidən böyükdürmü?
<b>var1 -le var2</b>	- Yoxlayın, 1-ci dəyişən 2-cidən kiçik və ya ona bərabərdirmi?
<b>var1 -lt var2</b>	- Yoxlayın, 1-ci dəyişən 2-cidən kiçikdirmi?
<b>var1 != var2</b>	- Yoxlayın, 1-ci dəyişən 2-ciye bərabər deyil ki?
<b>var1 -ne var</b>	- Yoxlayın, 1-ci dəyişən 2-ciye bərabər deyil ki?

## Sistemimizə yeni disk əlavə edək!

Diski serverimizə əlavə etdikdən sonra dmesg əmri ilə onun adına baxırıq. Ən son sətirdə və server console-da yeni disk haqqında məlumat ekrana çap ediləcək. Əmrin nəticəsi aşağıdakı şəkildəki kimi olacaq:

```
dal at mpt0 bus 0 scbus0 target 1 lun 0
dal: <VMware, VMware Virtual S 1.0> Fixed Direct Access SCSI-2 device
dal: 320.000MB/s transfers (160.000MHz, offset 127, 16bit)
dal: Command Queueing enabled
dal: 20480MB (41943040 512 byte sectors: 255H 63S/T 2610C)
root@:~ #
```

**newfs /dev/dal**

- **dmesg** əmrindən aldığımız nəticədə gördükümüz disk /**dev/dal**-i **UFS2** fayl sistemine format edirik.

**mkdir /disk**

- Bize lazım olan ünvanda disk yaradırıq.

**mount /dev/dal /disk**

- Və diskimizi sistemə mount edirik.

Diqqət! Əgər biz artıq fayl sistemi mövcud olan diskni sistemə mount etmək istəyiriksə, onda biz onun fayl sistem tipini müəyyən edib, sonra sistemə mount etməliyik.

Məsələn: External sərt disk sistemimizə mount edək. Eyni ilə dmesg əmri ilə adına baxırıq.

**Qeyd:** VmWare-də USB və ya External USB Storage istifadə edəndə diqqətli olun, çünki VmWare həmin driveri Windows-dan UNIX-ə ötürəndə düzgün emulyasiya etməyə bilər. 'dmesg' əmrinin çıxışında siz 'ugen' və 'umass' sətirlərindən başqa event görməsəniz, demək ki, VmWare-də problem var. Əgər onlardan sonrakı sətirlərdə 'da' adında yeni disk görsəniz, demək, hər şey qaydasındadır.

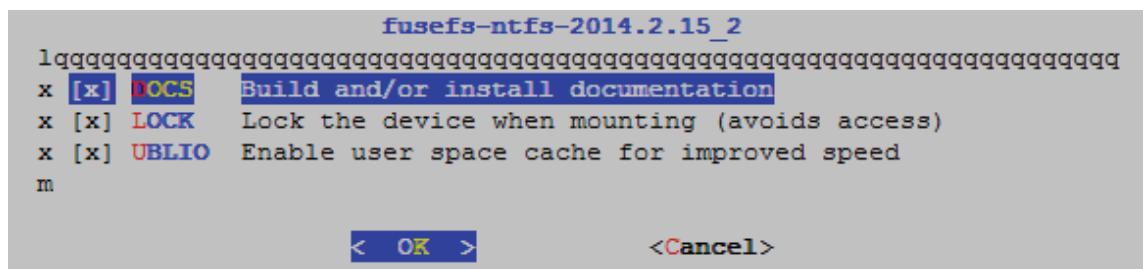
**dmesg** - da1 cavabı alırıq və 1-ci slice kimi mount edirik. Sistem disk ilk disk olduğuna görə da0 olur və ikinci disk avtomatik olaraq da1 adını alır.

```
mount_ntfs /dev/dals1 /mnt/
```

**Qeyd:** Bu halda **mount** edilən NTFS disk yalnız **READ only** (ancaq oxumaq) olacaq. Diskə yazma yetkisini təyin etmək üçün başqa metodla mount etmək lazımdır. Aşağıdakı qaydada bu göstərilir. Nəzərə alın ki, portlardan yüklemək üçün hələ sizin biliyiniz kifayət etmir. Yalnız üçüncü başlıqda portlar haqqında oxuduqdan sonra aşağıdakı əmrlər sizə aydın olacaq.

```
cd /usr/ports/sysutils/fusefs-ntfs
make config
```

- NTFS xüsusi driver(Port ünvanına daxil oluruq).  
- Bütün modulları seçirik.



```
make install clean
```

- Yükləyirik

**/etc/rc.conf** startup faylidir, hansı ki, sistemə aid olan **daemon**-lar və sonradan yüklənmiş bütün daemonların yenidənyüklənmədə avtomatik işə düşməsini idarə edir. Bunun üçün də "**/etc/rc.conf**" faylinin içində aşağıdakı sətirləri əlavə edirik ki, daemon yenidənyüklənmədə avtomatik işə düşsün.

```
fusefs_enable="YES"
```

**/usr/local/etc/rc.d/fusefs start** - CLI-dan əmri daxil edərək daemonu işə salırıq.

Modulun sistem yüklenməsi vaxtı aktiv olmasını istəsək, **/boot/loader.conf** faylinin içində aşağıdakı sətri əlavə edib, yadda saxlayaraq çıxırıq.

**fuse\_load="YES"**

**kldload /usr/local/modules/fuse.ko** - Sistemə reboot əmri vermədən CLI-dan bu əmr ilə modulu çağırıa bilərik.

**mkdir /disk**

**chmod -R 777 /disk**

- Mount etmək üçün qovluq yaradırıq.  
- Diskə tam yetki veririk (Ancaq verməsəniz də olar).

**ntfs-3g /dev/dals1 /disk**

- **ntfs-3g** (3-cü nəsil generasiya edilən Read/Write driver) **/dev/dals1** diskini **/disk** qovluğuna mount edirik.

**umount /disk**

- İşimizi bitirdikdən sonra umount edib diskini ayırırıq.

Həmçinin FAT32 fayl sistemdə olan USB Flash-i mount etmək üçün dmesg əmri ilə adına baxırıq.

**dmesg**

- Deyək ki, ilk disk olduğuna görə yenə də adı **dal** alırıq və içində fayl sistem olduğuna görə ilk slice-i mount edirik.

**mount -t msdosfs /dev/dals1 /mnt**

**df -h**

- Artıq USB Flash-a disk kimi baxa bilərik.

- Əmri daxil edib dəqiqləşdiririk ki, diskimiz sistemdədir.

**Qeyd:** Əgər mount etdiyimiz ünvanın sistemin yenidənyüklənməsindən sonra işləməsini istəyiriksə, mütləq həmin diskə **/etc/fstab** (fayl sistemlərin sistem yenidənyüklənməsində avtomatik mount olunması üçün fayl) faylinin sonuna uyğun sintaksislə əlavə etmək lazımdır.

Artıq biz diskimizi istifadə edə bilərik.

**file \***

- USB mount etdiyimiz qovluqda bütün fayl və qovluqların tipini bu əmrlə çap edirik.

## **Unix Tape Drive**

Susmaya görə Unix tape diskləri aşağıda göstərilən ünvanlardan oxuyur:

**/dev/sa0**  
**/dev/nsa0**

Hər bir halda tape alətinin statusunu öyrənmək istəsək, onu qlobal dəyişənə mənimsdədirik.

**setenv TAPE /dev/nsa0**

**mt status**  
**tar -t**

- Tape alətinin statusunu çap edir.
- Tape-in içinde olan arxivə siyahılayır.

# Sistem resursları və proseslər

Bu başlıqda sistemin fiziki resurslarının idarə edilməsi, onların asan təyin edilməsi üçün spesifik əmrlərdən istifadə və sistemdə işləyən proseslərin detallı açıqlanmasını müzakirə edəcəyik.

**pciconf -l -v**

- Sistemdə qoşulmuş bütün fiziki alətləri çap edir.
- "-l" detallı olmadan fiziki alətləri çap edir.
- "-v" hər fiziki alətin detallı olaraq bütün informasiyasını çap edir.

Məsələn: "-l" olduqda nəticə aşağıdakı kimi olacaq:

```
hostb0@pci0:0:0:0: class=0x060000 card=0x330b103c chip=0x34068086
rev=0x13 hdr=0x00
pcib1@pci0:0:1:0: class=0x060400 card=0x330b103c chip=0x34088086
rev=0x13 hdr=0x01
pcib2@pci0:0:2:0: class=0x060400 card=0x330b103c chip=0x34098086
rev=0x13 hdr=0x01
```

"-b" olduqda nəticə aşağıdakı kimi olacaq:

```
bar [10] = type Memory, range 32, base 0xda060000, size 131072, enabled
```

"-c" olduqda nəticə aşağıdakı kimi olacaq:

```
cap 01[dc] = powerspec 2 supports D0 D3 current D0
```

Əgər "-b" opsiyası təyin edildisə, onda "**pciconf**" bütün qeydiyyat indekslərini çap edəcək, hansı ki, hər bir alət üçün mənimsədilmiş resursdur.

BAR(Base Address)-dan sonra kvadrat mötərizelərdəki rəqəm BAR-ın 16-liqda yaddaşda olan yeridir.

BAR-ın tipləri "**Memory**", "**Prefetchable Memory**", ya da "**I/O Port**" ola bilər.

Range isə BAR-ın maximum decode ünvanıdır. "**base**" və "**size**" BAR adres pəncərəsinin əvvəlini və sonunu göstərir. Sonuncu flag isə BAR-ın işlədiyini və ya söndürüldüyünü göstərir.

Hal-hazırda yuxarıda 3 sətir görünür, hansı ki, üç fiziki alət deməkdir. İlk sütün alətin adını, modulun rəqəmini və selektorunu çap edir.

**Qeyd:** Əgər PCI alət üçün kernel quraşdırılmayıbsa, onda alətin adı "**none**" olacaq. Quraşdırılmamış modulun ilk rəqəmi sıfırdan başlayır və davam edir. Selektorun quruluşu isə digər bütün alətlərdə də istifadə oluna bilir.

İkinci sütün isə kodun klasıdır, bayt klası ilə birlikdə **subclass** və interfeys baytı ilə iki 16-liq rəqəm kimi çap olunur.

Üçüncü sütün "**subvendorid**" qeydiyyat informasiyasını çap edir, hansı ki, PCI standartının 2.1-ci versiyasından götürülür. İlk yarısında kartın öz ID-si, ikinci yarısında isə istehsalçının ID-si olur.

**Qeyd:** Köhnə kartlar üçün bu sıfır olacaq.

Dördüncü sütunda isə mikrosxemin ID-si olur, hansı ki, mikrosxemi identifikasiya edir, hansında ki, kartın özü qurulub. Bu da eynilə mikrosxemin ID-sini və istehsalçının ID-sini təşkil edir.

Beşinci sütun isə CHIP-in versiyasını çap edir.

Altıncı sütun başlığın tipini açıqlayır.

Hal-hazırda əksər alətlərin başlıq tipi 0-la başlayır, PCI və PCI bridge üçün 1, PCI və CardBus alətləri üçün 2.

**Qeyd:** Əgər ilk dəyəri olan qeydiyyat biti başlıq kimi PCI alətinə sıfır təyin olunubsa, bu, çox funksiyalı asılı olmayan və bir mikrosxem üzərində yığılmış alətdir.

```
dmesg | grep CPU:  
grep -i cpu /var/run/dmesg.boot
```

```
dmesg | grep memory
```

```
grep -i mem /var/run/dmesg.boot
```

- Sistem CPU-su haqqında məlumat almaq olur.
- Prosessorumuz haqqında tam informasiya çap edir.
- Sistemin real fiziki yaddaşı haqqında məlumatı megabaytlarla çap edir.
- RAM haqqında tam informasiya verir.

**Qeyd:** Əgər fiziki RAM haqqında detallı informasiya almaq istəsək, "free" PERL-də yazılmış paketlə baxa bilərik.

Məhz Linux-un "free" adlı programının nəticəsinə çox oxşayır. Ancaq PERL-də yazılmış skriptdir.

**Qeyd:** Unutmayın ki, skript PERL skriptidir və sistemdə perl-in mövcud olmasını tələb edir. Bu, o deməkdir ki, sizin sistemdə artıq PERL yüklənmiş olmalıdır.

Bizə lazım olan free paketini internetdən endiririk .

```
fetch http://www.cyberciti.biz/files/scripts/freebsd-memory.pl.txt  
mv freebsd-memory.pl.txt /usr/local/bin/free
```

```
chmod +x /usr/local/bin/free  
free | grep mem_total
```

- Adını dəyişib "free" edirik və sistem PATH-i olan "/usr/local/bin"-ə yerləşdiririk.
- Yerinə yetirilən edirik ki, əmr kimi işləsin.
- Bizə lazım olan tam fiziki RAM-ı görə bilərik.

## "ps" açıqlanma

```
ps -ajx
```

- "-a" terminal üçün istifadə olunan proseslərdən başqa prosesi çap etməyin.
- "-j" sütun düzülüşünü növbəti ardıcılıqla edin.

```
user  pid  ppid  pgid  sid  jobc  state tt      time  command
```

- "x" əvvəlki opsiyalar da daxil olmaqla terminal idarə etməsi olmayan bütün prosesləri də daxil edin.

USER	PID	PPID	PGID	SID	JOBC	STAT	TT	TIME	COMMAND
root	0	0	0	0	0	DLs	??	0:01.52	[kernel]

<b>USER</b>	- Process hansı istifadəçi adından işə salınıb?
<b>PID</b>	- Process IDentifier number
<b>PPID</b>	- Parent Process ID number, normal halda ata prosesdən nəsilliklə götürülür. Signal Processing və Job Control üçün istifadə olunur.
<b>PGID</b>	- Process Group Number ID
<b>SID</b>	- Session Identifier, PGID-ləri qruplaşdırmaq üçün istifadə olunur, adı halda single user, ya da daemon tərəfindən işə salınır. Process bir SID-dən digərinə "migration" keçid edə bilməz.
<b>JOBC</b>	- Job Control Count təyin edir ki, bu proses backgrounda işləyən Jobs-dur, ya yox.
<b>STAT</b>	- Symbolic Process State, Processlərin hal-hazırkı vəziyyətdə nə iş gördüyünü simvollarla çap edir. <b>D</b> - Prosesin diskdə gözləməsini təyin edir (kəsintisiz qısa müddət üçün). <b>I</b> - Passiv olan prosesi göstərir (hansı ki, 20 saniyədən artıq yatmış prosesləri çap edir) <b>L</b> - Prosesə nişan (flag) təyin edir, hansı ki, bağlanmanı gözləyir. <b>R</b> - Yerinə yetirilən prosesi çap edir. <b>S</b> - 20 saniyədən az olaraq, yatmış prosesləri göstərir. <b>T</b> - Dayanmış prosesi nişanlayır. <b>W</b> - Aktiv olmayan kəsintili axını təyin edir. <b>Z</b> - Ölmüş (dead) prosesi təyin edir. "zombie" Zombie - Bu, o prosesdir ki, ata prosesi var, amma prosesi gözləmədən çıxmışdır. Buna <defunct> da deyilir. Bu proses çıxmağa çalışanda blocklanır və "<exiting>" vəziyyətlə qeydə alınır.
<b>TT</b>	- Control terminal name (iki hərfli açıqlaması olur) "v" və ya "p" ilə başlayan terminal rəqəmlərini bildirir.
<b>??</b>	- Terminaldan işə salınmayan prosesleri çap edir.
<b>1</b>	- Bu, o deməkdir ki, proses /dev/ttyp1 console tərəfindən işə salınıb.
<b>TIME</b>	- Prosesin nə qədər vaxt həm istifadəçi, həm də kernel tərəfindən istifadə olunduğunu göstərir.

<b>COMMAND</b>	- İşə salınan əmrləri və arqumentləri bildirir.
<b>ps -U cavid</b>	- Əmr 'cavid' adlı istifadəçisinin istifadə etdiyi prosesləri çap edir.
<b>ps -U cavid -u</b>	- Əmr 'cavid' adlı istifadəçisinin hal-hazırkı proseslərini CPU və Memory ilə çap edir.
<b>ps -vU cavid</b>	- 'cavid' adlı istifadəçisinin işlək prosesləri, "re" və "sl" ilə. - 're' nüvənin iqamətgah vaxtı, saniyələrlə 127=sonsuzluq saniyə təyin olunub. - 'sl' yatdığı müddət saniyələrlə 127=sonsuzluq təyin olunub.
<b>ps -x1</b>	- Uzun müddət işləyən prosesləri çap edir.
<b>ps auwwx</b>	- Burada 'u' istifadəçi, 2 ədəd 'w' opsiyası o deməkdir ki, ps lazımlı olduqda çox sütun istifadə edəcək və bu da bizim səhifəyə belə yerləşməyə bilər.
<b>ps -L</b>	- Bütün mümkün ola bilən sütunları çap edəcək. Sonra onlardan birini seçərək hər sütuna ayrıraqda baxa bilərik. Məs: " <b>ps -xo %CPU</b> " təkcə CPU sütununu çap edəcək. " <b>ps o %mem</b> " təkcə MEM sütununu çap edəcək. " <b>ps -o blocked,state,user,usrpri,vsiz,vsz,wchan</b> " göstərilən düzülüşdə prosesləri çap edəcək. " <b>ps -xo ppid,user,%mem,tsiz,vsz,comm</b> " ppid-i istifadəçi prosesləri, memory-ni və s. çap edəcək.
<b>ps -aux -r   less</b>	- '-r' proses istifadəsinə görə CPU-nun istifadəsi.
<b>ps -m   less</b>	- "-m" çıxışın sort işini Memory üçün edin.
<b>ps -t ttyp0</b>	- Yalnız 'ttyp0'-in '-t' terminal proseslərini çap edəcək.
<b>ps -p 1129 -o pid,ppid,time,args</b>	- '-p' proses ID 1129 olan prosesin pid-ni, parent pid-ni time-ni və arqumentlərini çap edin.
<b>ps -U cavid,faxri -o pid,ruser,tty,stat,args</b>	- <b>cavid</b> və <b>faxri</b> istifadəçilərinə aid olan bütün prosesləri göstərilən sütun ardıcılığında edin.

PS sütunlarının açıqlanması aşağıdakı cədvəldə əks olunub:

<b>Opsiyası</b>	<b>Sütun başlığı</b>	<b>Açıqlanması</b>
%cpu	%CPU	CPU-nun istifadəsi faizlərlə
%mem	%MEM	Memory-nin istifadəsi faizlərlə
acflag	ACFLG	Accounting flag
args	COMMAND	Əmrlər və arqumentlər
comm	COMMAND	Ancaq əmr
command	COMMAND	Əmrlər və arqumentlər (args-la eynidir)
cpu	CPU	CPU istifadəsinin qısa termini (planlaşdırma üçün)
etime	ELAPSED	Yerinə yetirilmənin keçmiş vaxtı
flags	F	Process flags, 16-liqda (f-lə eynidir)
inblk	INBLK	Ümumi blockları oxuyur (inblock-la eynidir)
jid	JID	Jail ID
jobc	JOBC	Job control sayı
ktrace	KTRACE	Flag-ların trasirovkası
label	LABEL	MAC (Mandatory Access Control)-un təyin olmuş nöqtəsi
lim	LIM	Memory istifadəsinin limiti
lockname	LOCK	Bağlanma, hal-hazırda bağlanan (simvol adına görə)
logname	LOGIN	Login adı, kim ki, sessiyani işə salıb
lstart	STARTED	İşə salınan vaxt
majflt	MAJFLT	Fault olunan page-lərin sayı
minflt	MINFLT	Düzəldilmiş page-lərin ümumi sayı
msgrcv	MSGRCV	Qayıdan ümumi mesajlar (pipes/sockets oxunulması)
msgsnd	MSG SND	Ümumi göndərilmiş mesajlar (pipes/sockets yazılıması)
mwchan	MWCHAN	Kanalı gözləyin, ya da blocklayın, hal-hazırda blockludur
nice	NI	Prioritet dəyişəni ilə eynidir
nivcsw	NIVCSW	Kontekstin tam məqsədi olmadan keçidləri
nsigs	NSIGS	Ümumi götürülmüş siqnallar (nsignals-la eynidir)
nswap	NSWAP	Swap-in ümumi girişi və çıxışı

<b>nvcs w</b>	<b>NVCSW</b>	Kontekstin istəyi olan tam keçidləri
<b>nwchan</b>	<b>NWCHAN</b>	Kanalı gözləyin (ünvan kimi)
<b>oub lk</b>	<b>OUBLK</b>	Ümumi yazılmış blocklar (oublock-la eynidir)
<b>paddr</b>	<b>PADDR</b>	Swap ünvani
<b>pagein</b>	<b>PAGEIN</b>	Page giriş üçün (majflt-la eynidir)
<b>pgid</b>	<b>PGID</b>	Qrupun Process nömrəsi
<b>pid</b>	<b>PID</b>	Process ID
<b>ppid</b>	<b>PPID</b>	Valideyn process ID
<b>pri</b>	<b>PRI</b>	Prioritetin planlaşdırılması
<b>re</b>	<b>RE</b>	Nüvənin saniyələrlə olan iqamətgah vaxtı (127 sonsuzluq deməkdir)
<b>rgid</b>	<b>RGID</b>	Realda olan qrup ID
<b>rgroup</b>	<b>RGROUP</b>	Qrup adı (real "group ID"-lə əlaqəlidir)
<b>rss</b>	<b>RSS</b>	Resident yığılma həcmi
<b>rtprio</b>	<b>RTPRIO</b>	Real- vaxtda olan priority (101 deməkdir ki, bu proses real-time deyil)
<b>ruid</b>	<b>RUID</b>	Real user ID
<b>ruser</b>	<b>RUSER</b>	User name ("real user ID"-lə əlaqəlidir)
<b>sid</b>	<b>SID</b>	Session ID
<b>sig</b>	<b>PENDING</b>	Gözləmə siqnalları (pending 'gözləmə' ilə eynidir)
<b>sigcatch</b>	<b>CAUGHT</b>	Tutulma siqnalları (tutulma ilə eynidir 'caught')
<b>sigignore</b>	<b>IGNORED</b>	Məhəl qoyulmayan siqnallar (ignored-lə eynidir)
<b>sigmask</b>	<b>BLOCKED</b>	Blocklanmış siqnallar (blocked-lə eynidir)
<b>sl</b>	<b>SL</b>	Yatma vaxtı saniyələrlə (127 sonsuz rəqəm deməkdir)
<b>start</b>	<b>STARTED</b>	İşə düşmə vaxtı
<b>state</b>	<b>STAT</b>	Symbolic prosesin statusu
<b>svgid</b>	<b>SVGID</b>	Yerinə yetirilən "setgid" dən yadda saxlanılmış gid.
<b>svuid</b>	<b>SVUID</b>	Yerinə yetirilən "setuid" dən yadda saxlanılmış UID.
<b>tdev</b>	<b>TDEV</b>	Terminalı idarə edən alətin adı.
<b>time</b>	<b>TIME</b>	Yığılmış CPU vaxtı, user + system (cpitime-la eynidir)
<b>tsid</b>	<b>TSID</b>	Control terminal session ID

<b>tsiz</b>	<b>TSIZ</b>	Mətn həcmi Kbyte-larla
<b>tt</b>	<b>TT</b>	Control terminal name, iki hərflə açıqlama
<b>tty</b>	<b>TTY</b>	Control terminalin tam adı
<b>uprocp</b>	<b>UPROCP</b>	Prosesi göstərən
<b>ucomm</b>	<b>UCOMM</b>	accounting istifadə olunan ad
<b>uid</b>	<b>UID</b>	Effective user ID
<b>upr</b>	<b>UPR</b>	Sistem çağırışı ilə gələn prioritətin planlaşdırılması (usrpri-lə eynidir)
<b>user</b>	<b>USER</b>	İstifadəçi adı (user ID-dən)
<b>vsz</b>	<b>VSZ</b>	Virtual həcm Kbyte-larla (vsize-la eynidir)
<b>wchan</b>	<b>WCHAN</b>	Symbolic ad kimi gözləmə kanalı
<b>xstat</b>	<b>XSTAT</b>	Exit, ya da stop status (dayandırılmış və ya ancaq zombie proseslər)

#### fstat – aktiv faylların təyin edilməsi

Əməliyyat sistemimizdə işləyən hər hansıa istifadəçinin mövcud vaxt üçün nə qədər fayldan istifadə etdiyinin təyin olunması sualı bizə hər an verilə bilər. Ya da başqa bir tələb forması uyğun olaraq işlək proses üçün ola bilər. Misal üçün, 2376 ID ilə açılmış bütün faylların siyahısını çap etmək.

- fstat -u root** - "root" istifadəçi tərəfindən açılmış faylları çap edir.
- fstat -p 1885** - "1885"-ci Process ID-dən açılmış bütün fayllarını çap edir.
- fstat -m -p 1885** - Həmçinin yaddaşda xəritələnmiş faylları da çap edəcək. Adı halda olmur.

# İşlək proseslərin yoxlanılması və idarə edilməsi

Bu başlığımızda proseslərin sistemin öz daxili imkanları və kənar paketlərin vasitəsilə statusların əldə edilməsi və onların idarə edilməsi haqqında danışılır.

## ProcFS

Bütün prosesləri çıkışda fayllarda görmək üçün biz '**procs**' file sistemini mount etməliyik. Bunun üçün **/etc/fstab** faylinə aşağıdakı sətri əlavə edirik:

<b>proc</b>	<b>/proc</b>	<b>procfs</b>	<b>rw</b>	<b>0</b>	<b>0</b>
-------------	--------------	---------------	-----------	----------	----------

**mount -a** - Proc file sistemini işə salırıq.

Hər bir prosesin altında olan struktur aşağıdakı ardıcılıqla olur:

- **status** (read-only): Proses statusunu qaytarır
- **mem** (read/write): Proseslərin virtual memory nüsxəsi
- **file** (depends): İslək proseslərə simvolik link
- **regs** (read/write): Proses qeydiyyatları
- **ctl** (write-only): Prosesə siqnal ötürmək üçün istifadə edilir, ya da debug etmək üçün onun attach/deattach edilməsi
- **cmdline** (read-only): İslək proseslər üçün komanda sətri arqumentləri

- <b>rlimits</b> (read-only):	İşlek proses üçün hal-hazırkı proses limiti
- <b>map</b> (read-only):	İşlek prosesler üçün yaddaş xəritələnməsi
- <b>etype</b> (read-only):	Yerinə yetirilmə tipi(eg. FreeBSD ELF32)
- <b>fregs</b> (read/write):	Üzən nöqtə qeydləri

### ProcStat

Proses haqqında ətraflı informasiya çap edir.

<b>procstat -a</b>	- Bütün proseslərin statusunu çap edir.
<b>procstat -kk 1122</b>	- Prosesin kernel-də olan stack-ni çap edir, CPU-da işləyən stack-ləri və diskdən gələn stack-ləri çıxməq şərti ilə. Əgər flag təkrarlanırsa, funksiyaların adları da çap olunur.
<b>procstat -t 10</b>	- '-t' 10 ID-li proses haqqında olan ( <b>thread</b> ) informasiya axınıni çap edir.
<b>procstat -v 10</b>	- '-t' 10 ID-li proses altında olan virtual yaddaş xəritələnməsini çap edir.
<b>procstat -s 10</b>	- Bu proses haqqında təhlükəsiz verilənləri çap edir.
<b>procstat -b 10</b>	- '-b' 10 ID altında işləyən binar informasiyanı çap edir.
<b>procstat -c 10</b>	- '-c' 10 ID-sinin CLI arqumentini çap edir.
<b>procstat -f 10</b>	- '-f' 10 ID-sinin File Deskriptorunu çap edir.
<b>procstat -i 10</b>	- '-i' 10 ID-si altında işləyən prosesin siqnal gözləməsini və yerləşmə informasiyasını çap edir.
<b>procstat -j 10</b>	- '-j' 10 ID-si altında işləyən prosesin siqnal gözləməsini və bu proses axınının bloklanmış hissəsini çap edir.

### PStree

Proseslərin iyerarxiyasına baxmaq üçün 'pstree' adlı paketdən istifadə edə bilərik.

<b>cd /usr/ports/sysutils/pstree/</b>	- Port ünvanına daxil oluruq.
<b>make install clean</b>	- Yükləyirik.
<b>pkg install pstree</b>	- Həmçinin paketlərdən yükleyə bilərik.
<b>pstree</b>	- Nəticədə hansı prosesin ata və hansıların bala olduğunu görə bilərik.

Məsələn: HTTPD-nin altında hal-hazırda 5 ədəd bala proses var.

```
|+= 00983 root /usr/local/sbin/httpd -DNOHTTPACCEPT
| |--- 01013 www /usr/local/sbin/httpd -DNOHTTPACCEPT
```

```
| |--- 01014 www /usr/local/sbin/httpd -DNOHTTPACCEPT  
| |--- 01015 www /usr/local/sbin/httpd -DNOHTTPACCEPT  
| |--- 01016 www /usr/local/sbin/httpd -DNOHTTPACCEPT  
| \--- 01017 www /usr/local/sbin/httpd -DNOHTTPACCEPT
```

**pstree -U** - Root istifadəçisinə aid olan proses konteynerini çap etmək.

**pstree -u www** - Yalnız **www** istifadəçisinə aid olan prosesləri çap edəcək.

**pstree -s httpd** - **httpd** adlı proseslərin hamısını çap edəcək.

**pstree -h** - Köməkçini çap edir.

**pstree -p 00921** - 00921 PID-i və ona aid olan bütün prosesləri çap edəcək.

#### Proseslərin təpiləsi və idarə edilməsi

**pgrep** – Proseslərə adı ilə siqnal ötürür, ya da axtarır.

**pgrep httpd** - **httpd** adı altında işlənən bütün prosesləri yalnız PID kimi çap etmək.

**pgrep -l httpd** - **httpd** adı altında işlənən bütün proseslərin həm adını, həm də proseslərini çap edir.

**pgrep -lu cavid** - **cavid** istifadəçi adından işlənən bütün prosesləri çap etmək.

**ps -p `pgrep httpd`** - **pgrep** əmri ilə **httpd** adına axtarış et və çıxan nəticəni **ps** əmri vasitəsilə '**-p**' prosesə görə axtarış et.

**ps -fp `pgrep httpd`** - **pgrep** əmri ilə **httpd** adına axtarış et və çıxan nəticəni **ps** əmri vasitəsilə '**-p**' prosesə görə '**-f**' tam axtarış et.  
Qeyd: '**-f**' opsiyası yalnız root istifadəçi adından işlənə bilər.

**renice -5 `pgrep httpd`** - **httpd** proseslərini axtarış edib, renice əmrinə ötürürük ki, '**-5**' prioriteti təyin etsin. Prioritet 0 .. 20 və -20 aralığında verilir. Daha kiçik rəqəm daha üstün deməkdir.

## fuser

**fuser** – Bir və ya bir neçə fayl açmış bütün proseslərin İD-lərini çap edir. Sayəsində hansı prosesin hansı file və ya socket açdığını görə bilərik. Fuser prosesi tapdıqdan sonra ona signal da göndərə bilir. FreeBSD köhnə versiyalarında portlarda **/usr/ports/sysutils/fuser** ünvanında yerləşirdi. Artıq sistemin bir hissəsinə çevrilib.

**pkg\_add -r fuser** - Hər halda köhnə versiya FreeBSD-də bu əmrlə yükleyə bilərsiniz.

**fuser -cu /home** - "/**home**" file sistemində işə salınmış prosesləri çap etmək (istifadəçi adı ilə)  
'-c' əmri '/**home**' file sistemini tamlıqla götürür, yəni daxilində olan bütün faylları, hətta gizli olanları belə.  
'-u' Prosesi işə salan hər bir istifadəçi list olunacaq.

**Qeyd:** İstifadəçi ilə proses arasında olan simvollar müəyyən məna daşıyır.

'x' - file, mətn prosesindən yerinə yetirilən programdır.  
'c' - fayl üçün hal-hazırkı işlək kataloqun prosesi.  
'r' - fayl proseslərə kök qovluğundan gəlib.  
'j' - fayl proseslərdə olan root Jail-dır.  
't' – proseslərdə olan fayl kernel trace faylıdır.

'y' - proses bu faylı '**tty**'-i control eləmək üçün istifadə edir.

'm' - fayl xəritələnmişdir.  
'w' - fayl yazılımaq üçün açılmışdır.  
'a' - fayl ancaq əlavə edilmək üçün açılmışdır.  
's' - faylin shared lock-u var.  
'e' - fayl tamamilə lock olmuşdur.

**fuser -c -m /boot** - "/**boot**" qovluğundan işə salınan proseslərin **PIDs/symbols**-nu çap edir.

**fuser -u /boot** - "/**boot**" qovluğundan işə salınan proseslərin **PIDs/symbols/user**-ni çap edir. Alt qovluqları yox.

**fuser -k /tmp/my.txt** - "/**tmp/my.txt**" faylini açan bütün prosesləri '-k' kill etmək.

### Proseslərə prioritetin təyin olunması

Susmaya görə prioritet **0** olur. **Root** istifadəçisi istənilən prosesin prioritetini dəyişə bilər. FreeBSD OS-da prioritet **-20** və **19** aralığında təyin edilə bilər.

**-20** ən böyük prioritetdir, **19** ən zəif prioritetdir. **0** - sıfırıncı prioritetdir.

İstifadəçi isə yalnız böyük prioriteti olan prosesin prioritetini azalda bilər.

**nice -n 12 nroff -man a.woff | less** - Man səhifə üçün prioriteti azaldırıq.

**nice -n -10 designer** - Dizayner üçün prioriteti yüksək qoyuruq.

**renice +2 -u cavid** - "**cavid**" adlı istifadəçi tərəfindən açılan proseslərin prioritetini +2 edirik.

**renice +5 4737** - '**4737**' ID ilə işləyən prosesin prioritetini +5 edir.

**renice -3 `pgrep -u cavid spamd`** - '**pgrep**' (cavid) adlı istifadəçini '**spamd**' prosesini tapıb renice-a ötürür. O isə öz növbəsində onu artırıb "**-3**" edir.

### Proseslerin öldürülməsi və siqnalın ötürülməsi

Proseslərə ötürfmək üçün standart siqnallar.

<b><u>Signal sayı</u></b>	<b><u>Signal adı</u></b>	<b><u>Açıqlanması</u></b>
1	<b>SIGHUP</b>	Terminaldan dondurulmuş, ya da idarəedici proses olmuşdur
2	<b>SIGINT</b>	Program kəsilmişdir
3	<b>SIGQUIT</b>	Programdan çıxmış
4	<b>SIGILL</b>	Qadağan olunmuş əmr
5	<b>SIGTRAP</b>	Trasirovka kəsilmələri
6	<b>SIGABRT</b>	İşin qəza vəziyyətində olmasından kəsilməsi, " <b>abort function</b> "-dan yollanılmış əmr
7	<b>SIGEMT</b>	Göstərilən instruksiyani emulyasiya etmək.
8	<b>SIGFPE</b>	Nöqtə vergüllə istisna
9	<b>SIGKILL</b>	Öldür siqnalı
10	<b>SIGBUS</b>	Bus error
11	<b>SIGSEGV</b>	Bölüşdürülmənin pozulması

12	<b>SIGSYS</b>	Mövcud olmayan sistem çağrıları çağrılmışdır
13	<b>SIGPIPE</b>	Zədələnmiş kanal (pipe-a yazmaq və ya oxumaq üçün heç bir şey yoxdur)
14	<b>SIGALRM</b>	"alarm system call"-dan siqnal timeri
15	<b>SIGTERM</b>	İşin bitmə siqnalı "termination"
16	<b>SIGURG</b>	Atma siqnalı
17	<b>SIGSTOP</b>	Prosesi dayandır
18	<b>SIGTSTP</b>	Terminaldan stop signal tipi gəlmışdır
19	<b>SIGCONT</b>	Proses dayanırsa, hər bir halda davam etmək
20	<b>SIGCHLD</b>	Bala proses vəziyyəti dəyişdirilmişdir
21	<b>SIGTTIN</b>	Terminal background fonda olan prosesi oxumağa çalışır
22	<b>SIGTTOU</b>	Terminal background fonda olan prosesə yazmağa çalışır

- kill 28665** - '28665' ID-li prosesə **SIGTERM** siqnalı yollamaq
- kill -9 4895** - '4895' ID-li prosesə **SIGKILL** siqnalı yollamaq
- kill -SIGCONT 5254** - '5254' ID-li dayandırılmış prosesi davam etmək.
- kill -9 %3** - '%3' jobsda 3-cu ID altında olan işe öldür siqnalı yollamaq.
- killall spamd** - spamd adı ilə başlayan bütün prosesləri öldür .
- killall -SIGHUP sendmail** - sendmail adlı proses varsa, quraşdırma fayllarını yenidən oxumaq.

**Qeyd:** Hal-hazırkı seansımızdan çıxsaq, belə prosesin işləməsini istəsək, bir neçə yol var.  
Məsələn: nohup-la prosesə dayanma əmri çatılmaz siqnalı ötürürük.

**nohup myscript.sh &** - 'myscript.sh' skriptini elə işə salın ki, o '**interrupt**' siqnalını qəbul etmir.

**nohup nice -9 gcc hello.c &** - Yerinə yetirilən "gcc" kəsilməzdir və daha böyük prioritetə malikdir.

## Proseslerin işə salınması üçün planlaşdırılması

Cron-la sistemdə planlaşdırılan işlər birbaşa **CLI**-dan da yerinə yetirilə bilər. Bunun bir neçə yolu var: '**batch**' və '**at**'. Hər birinin imkanlarını açıqlayırıq.

**at now +1 minute**

- Proqramı '**now**' indi '**+1**' 1 dəqiqədən sonra işə salmaq. Bu əmrden sonra console açılır və ora lazımı əmrlərimizi daxil edirik, sonra yadda saxlayıb çıxməq üçün "**Ctrl+D**" əmrindən istifadə edirik.

**ls -R /usr/ports > /tmp/portlist.txt**

- at-in console-unda yazılıq ki, "**/usr/ports**"-da olan hər şeyi siyahılıyb, "**/tmp/portlist.txt**" faylına yazsın. Sonra "**Ctrl+D**" ilə çıxırıq. '**Ctrl+D**' prosesi növbəyə sal deməkdir.

**at teatime**

- Əmri bu gün saat 16:00-da işə sal deməkdir.

**at now +5 days**

- Əmri 5 gündən sonra işə sal deməkdir.

**at 03/25/12**

- Əmri 2015-ci ilin 03-cü ayının 25-də işə sal deməkdir.

Növbəti əmr işə əmri işə salır, hansı ki, hal-hazırkı shell-lə qoşulu deyil. "**batch**" əmri ilə. '**batch**'-la əmri o zaman işə salırıq ki, prosessor özü hazır olsun. (Yüklənmə aralığı 8%-dən az olmalıdır.)

**Qeyd:** Yəni CPU özü qərar verəcək ki, nə zaman işə salsın.

**batch**

- Əmrini işə salırıq, hansı ki, həmin anda da işləsin. Burada da 'batch' mühiti açılır və ora öz əmrlərimizi daxil edib, "**Ctrl+D**" ilə çıkış edirik.

**find /home/cavid | grep txt\$ > /tmp/mytxts**

- **.txt** fayllarını axtarıb fayla yazılıq. '**Ctrl+D**' ilə çıkış edirik.

Əmlər '**at**'-dan, ya da '**batch**'-dan daxil edildikdən sonra onların növbələşmə sıyahısını '**atq**' əmri ilə görə bilərik.

**atq**

- Əmri daxil etdikdən sonra aşağıdakı nəticə çıxır. Və biz JOB ID-yə görə növbələməni silə bilərik.

**Date**

**Owner**

**Queue**

**Job #**

**Thu Mar 26 16:43:00 AZT 2015**

**root**

**E**

**5**

**atrm 5**

- **5** Job ID altında olan job-u silirik.

# Rezerv nüsxələr və bərpa edilməsi, istifadəçilərin SUDO ilə məhdudlaşdırılması

Əməliyyat sistemin üzərində 24/7 dayanıqlı vəziyyətdə çox vacib bir program təminatı işlərsə, həmin programın və sistemin son informasiya dəyişikliyi statusu mütləq rezerv nüsxəyə alınmalıdır (Bildiyimiz **Backup**). OS üzərində elə zamanlar olur ki, müəyyən bir əmrin yalnız seçilmiş parametrlərlə işləməsi hansısa adı istifadəçi üçün verilməldir. O halda köməyimizə SuDO (**substitute Do**) çatacaq. Bu başlıqda biz sistemin rezerv nüsxəyə alınması, onun necə bərpa edilməsi və SuDO haqqında danışacaqıq.

**dd if=/dev/da0 of=/dev/da1** - Bütövlükə 'da0' diskini 'da1' diskinə nüsxələyəcək.

Öncə göstərilən əmrde '**bs**' (**block size**) təyin etmədiyimiz üçün disklər həddən artıq gec nüsxələnəcək. Çünkü susmaya görə '**bs**' **512** baytlə köçürülcək.

Diskin cache-ni artırısaq, məs: **bs=8M**, o, hər porsiyada 8 megabayt köçürəcək. Əgər "**conv=sync,noerror**" artırısaq, onda deyəcəkyik ki, sinxronizasiyanı "**bit to bit**" etsin və oxunma zamanı çıxan səhv'lərə məhəl qoymasın.

**Qeyd:** dd işləyən zaman köçürülen informasiyanın statusuna "**Ctrl+t**" əmri ilə baxmaq olar. Bu, CSH mühitində işləyir.

```
dd if=/dev/da0s1a of=/home/da0s1a.img bs=8M conv=sync,noerror
```

- Burada "da0s1a" slice-nı "/home"-da, "da0s1a.img" adlı image formatına yazırıq.

Göstərilən əmrde "da0s1a" slice-nı "/home"-da, "da0s1a.img" adlı image-ə, gzip-lə sıxıb yazırıq.

```
dd if=/dev/da0s1a bs=8M conv=sync,noerror | gzip -c > /home/da0s1a.img
```

```
dd if=/dev/acd0 of=/home/8.0-RELEASE-x64.iso bs=9M
```

- Burada cdrom-da olan FreeBSD diskinin ISO nüsxəsini home qovluğumuza çıxarıraq.

**Qeyd:** Öncə "/dev/acd0" cdrom qovluğuna mount olmalıdır.

```
mdconfig -a -t vnode -f /home/10.1-RELEASE-x64.iso -n 0
```

- Burada yaratdığımız iso faylini sistemdə "md0" adlı virtual aləti mənimsədirik.  
"-a" – mənimsədirik.  
"-t" - tipi "vnode"  
"-f" - faylı haradan götürəcək?  
"/home/8.0-RELEASE-x64.iso"  
"-n" - hansı rəqəmli "md" virtual aləti yaranacaq?

```
mount_cd9660 /dev/md0 /cdrom
```

- Burada isə "/dev/md0" adlı aləti "/cdrom" qovluğuna mənimsədirik.

**Qeyd:** Hər hal üçün cdrom-u öncə umount edirik.

### Rezerv nüsxənin bərpə edilməsi

```
dd if=/home/da0s1d.img of=/dev/dal bs=8M conv=sync,noerror
```

- "das1d.img" image-nı 'dal' diskinə bərpə edirik.

Növbəti əmr ilə sıxılmış "da0s1d.img" nüsxəsini açıb, həmin kanalla 'dd'-yə yollayıraq ki, "dal" diskinə yazsın.

```
gunzip -c /home/da0s1d.img | dd of=/dev/dal conv=sync,noerror bs=8M
```

```
dd if=/home/var.img of=/dev/da0s1d bs=8M conv=sync,noerror
- 'var.img' nüsxəsini "/dev/da0s1d" slice-na
bərpa edirik.
```

**Qeyd:** Subslice-i bərpa eləmək üçün, o, unmount olmalıdır. Yəni yenidənyüklənmə edib, adı istifadəçili rejimə keçirik və "var" slice-dan başqa bütün slice-ları mount edirik. Sonra da "dd" ilə bərpa işini görürük. Ardınca "fsck -y /dev/da0s1d" əmri ilə diskı yoxlayırıq və "var" slice-ni "/dev/da0s1d" ünvanına mount edirik .

## Dövr üçün SEQ

Sistemdə müyyəyen sınaqları aparmaq üçün və ya saygac yaratmaq ucun "jot", ya da "seq2" əmrlərindən istifadə etmək olar.

**Qeyd:** "seq2" susmaya görə sistemdə olmur. Siz onu portlardan yükləməlisiniz.

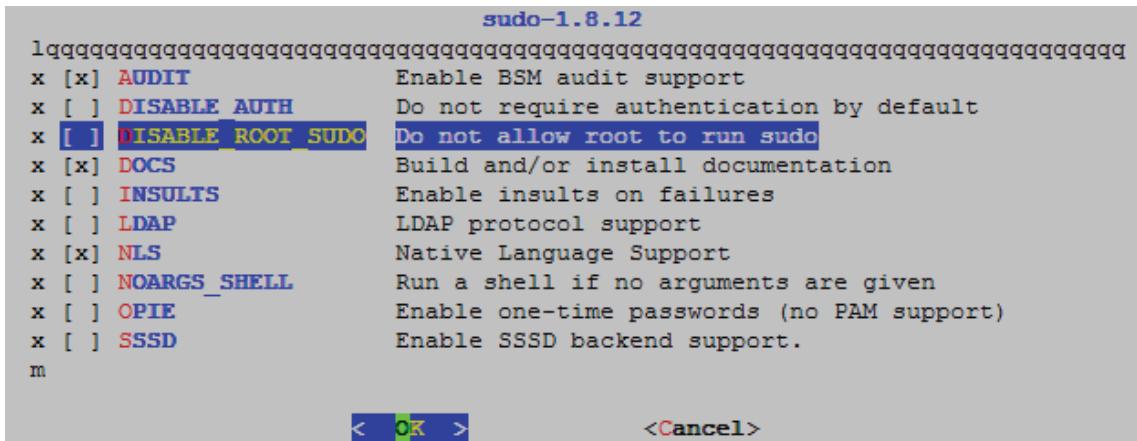
```
jot 10
cd /usr/ports/misc/seq2
make install clean
seq2 -s 1 -e 7 -n > file
- Ardıcıl olaraq 10-dək sayacaq.
- Port ünvanına daxil olub,
- yükləyək.
- "seq2" əmri deyir ki, say "s" (start) 1-dən
başlayaraq "-e" (end) 7-dək,
"-n" yeni sətrə keçidi söndür və "file" adlı fayla yaz.
```

## İstifadəçilərin SuDO ilə məhdudlaşdırılması

Istifadəçiləri əməliyyat sisteminde olan fayllara və əmrlərə məhdudiyyət təyin etmək üçün SuDo paketindən istifadə edə bilərik. Paketi yükləyək.

```
cd /usr/ports/security/sudo
make config
make install clean
- Port ünvanına daxil oluruq.
- Əmrlə aşağıdakı şəkildə göstərilən modulları
seçirik.
- Yükləyirik.
```

**Qeyd:** Susmaya görə SuDo-nun quraşdırma faylı "/usr/local/etc/sudoers" ünvanında yerləşir. Heç bir zaman bu faylda öz redaktorundan başqa redaktorla dəyişiklik etməyin. Yəni "visudo" (SuDO üçün spesifik editor) ilə dəyişiklik edib, yadda saxlayaraq çıxış edin. Redaktor sintaksisi "vi" ilə eynidir. Çünkü "visudo" avtomatik olaraq, "vi"-yə müraciət edir. "vi" əmri haqqında növbəti bölmələrimizdə ətraflı danışacaqıq.



Misal olaraq, cavid adlı istifadəçi üçün 3 ədəd SuDo məhdudiyyəti əlavə edək.

<b>root</b> ALL=(ALL) ALL	- Susmaya görə " <b>root</b> " istənilən ünvandan istənilən əmri istifadə edə bilər.
<b>cavid</b> ALL= /usr/bin/less /var/log/messages	- " <b>cavid</b> " adlı istifadəçi istənilən ünvandan " <b>less</b> " əmri ilə " <b>/var/log/messages</b> " faylinə baxa bilər.
<b>cavid</b> ALL= /usr/bin/ee /etc/rc.conf	- " <b>cavid</b> " adlı istifadəçi istənilən ünvandan " <b>ee</b> " redaktoru ilə " <b>/etc/rc.conf</b> " faylini redakte edə bilər.
<b>cavid</b> ALL= /etc/rc.d/inetd restart	- " <b>cavid</b> " adlı istifadəçi istənilən ünvandan " <b>inetd</b> " daemonunu restart edə bilər.
<b>cavid</b> 192.168.192.151 = /etc/rc.d/sshd restart	- " <b>cavid</b> " adlı istifadəçi " <b>192.168.192.151</b> " IP ünvanlı mənbədən " <b>sshd</b> " daemonu yenidən işə sala bilər.
<b>%admin</b> ALL=(ALL) ALL	- " <b>admin</b> " qrupuna daxil olan hər kəs hər bir əmri yerinə yetirə bilər.

# BÖLÜM 3

## Vacib əmrlər, Swap idarə edilməsi, portlar, paketlər, arxivlər, sətir əməliyyatları

- / Sistemdə həmişə istifadə ediləcək vacib əmrlərin detallı açıqlanması
- / İslək sistemdə Swap həcminin artırılması
- / Portlar, paketlər və onların idarə edilməsi
- / Fayl/qovluqların arxivlənməsi, sıxılması və grep sətir preprocessoru

Başlıq gündəlik tələbdə olan vacib əmrlərin detallı açıqlanmasını, faylların hissələrə bölünməsi, SWAP-ın idarə edilməsi, üçüncü tərəfin hazırladığı port və paketlərin müxtəlif üsullarla yüklənməsi və onların gündəmdə qalmalarını, faylların arxivləşdirilməsi və sıxılmasını, sətirlərlə aparılan işlərin avtomatlaşdırılmasını açıqlayır.

# Sistemdə həmişə istifadə ediləcək vacib əmrlərin detallı açıqlanması

Əməliyyat sistemində işə başlamazdan önce müəyyən əmrləri bilmək lazımdır ki, işimizi görə bilək. Bu əmrlərdən **cd**, **ls**, **df**, **du**, **pwd**, **ee** və **vi** başlanğıcda öyrənilməli olanlardır.

**tty** seanslara baxmaq istəsək, **/etc/ttys** faylinə daxil olmaq lazımdır.

Orada Dial-Up, SSH və Grafik interfeysler üçün **tty** sayı təyin olunub. Öz **tty** virtual alətinizə baxmaq üçün **tty** əmrini daxil etməyiniz kifayətdir.

Bir ünvandan digərinə nüsxə çıxaranda nüsxələnən hər bir məlumatın siyahı şəklində, haradan-haraya getdiyini görmək istəsək, aşağıdakı əmri yığmaq kifayətdir.

```
cp -apvf * .. /FreeBSD && cd .. /FreeBSD && ls -lh
```

**ls -lh**

- Çap olunan informasiyanın bayt və ya gigabaytlı çap olunmasını istəsək, aşağıdakı kimi edirik.

Əgər FTP serverdə istifadəçilərə banner çıxmاسını istəsək, iki fayl yaratmalyıq. Qeyd: FTP qurulması şəbəkə başlığıımızda göstəriləcək.

**/etc/ftpwelcome**  
**/etc/ftpmotd**

- Burada girişdən əvvəl mesaj çıxır.  
- Burada girişdən sonra mesaj çıxır.

Əgər əməliyyat sistemində olan istənilən istifadəçi üçün banner təyin etmək istəsək, '/etc/motd' faylına çap ediləcək məzmunu əlavə edirik. İstifadəçi sistemə daxil olduqdan sonra faylin içindəki məlumat istifadəçinin ekranında əks olunacaq.

## Top əmrin açılınması

Top əmrini aşağıda göstərilən qaydada CLI-dan daxil edirik və hər bir önəmlı sütunda rəqəm ardıcılılığı təyin edirik. Təyin edilən rəqəmləri ardıcılıqla açıqlayıraq.

# top

```
1.last pid: 1218; 2.load averages: 0.10, 0.09, 0.07 3.up 0+00:12:06 03:24:09
4.18 processes: 1 running, 17 sleeping
5.CPU: 0.0% user, 0.0% nice, 0.4% system, 0.0% interrupt, 99.6% idle
6.Mem: 12M Active, 8480K Inact, 32M Wired, 20K Cache, 14M Buf, 923M Free
7.Swap: 1024M Total, 1024M Free
```

1. Sistem tərəfindən təyin olunmuş son Process ID-nin PID-i
2. Sistemdəki CPU-nun son yüklənmə rəqəmi. Cox qeyri-dəqiq bir prinsiplə işləyir.  
**0.10** - bu rəqəm prosessorun son dəqiqə ərzində yüklənməsini bildirir.  
**0.09** - bu rəqəm prosessorun son 5 dəqiqə ərzində yüklənməsini çap edir.  
**0.07** - bu rəqəmsə prosessorun son 15 dəqiqə ərzində yüklənməsini bildirir.
3. Bu bölüm isə sistemin nə qədər müddət işlədiyini göstərir,  
burada sistem > "**0+00:12:06**" 12 dəqiqə işləyir.  
Bu isə > "**03:24:09**" sistemdə olan vaxtdır.
4. Sistemdə hal-hazırda işləyən proseslərin sayı **18**-dir. **1** ədəd işlek proses və **17** yatmış proses var.  
Yatmış proseslər müəyyən bir mənbədən girişi gözləyirlər.

**Qeyd:** Bəzi istifadəçilər sistemdə həddən artıq proses işə salıb, sistemi "çökdürə" bilərlər.  
Buna "**forkbombing**" deyilir.

5. **CPU:** sistem tərəfindən müxtəlif tipli proseslərin emalına sərf edilən, mövcud olan CPU vaxtinin faizlərlə göstəricisidir. Burada 5 müxtəlif tipli proses göstərilir. "**user**", "**nice**", "**system**", "**interrupt**", "**idle**"

**user**

- Bu, hər gün ya "root", ya da digər istifadəçilər tərəfindən işə salınan proseslərdir.

<b>nice</b>	- Bu prosesler istifadəçi prosesləridir, hansılarına ki, prioritet verilib. ( <b>nice</b> - prioritet vermək üçün istifadə olunur.)
<b>system</b>	- Bu bölüm isə, kernelde olan FreeBSD kernel və istifadəçi proseslərinin faizlərlə ümumi CPU vaxtını göstərir.
<b>interrupt</b>	- Yarımçıq müraciətlərin emalı üçün sistem CPU-dan neçə faiz sərf edir? ( <b>IRQs</b> - Interrupt Requests)
<b>idle</b>	- Burada isə sistem neçə faiz CPU-dan istifadə etmir? CPU boşdur.

6. **MEM**: Bu hissə sistemin fiziki RAM istifadəsi haqqında məlumat verir. FreeBSD RAM istifadəsini müxtəlif tipli kateqoriyalara böölür.

**"Active", "Inact", "Wired", "Cache", "Buf", "Free"**

<b>Active</b>	- İstifadəçi prosesləri tərəfindən istifadə olunan RAM-in ümumi sayı.
<b>Inact</b>	- Programın işi bitdikdə onun istifadə etdiyi yaddaş hissəsi buraya keçir.
<b>Cache</b>	- Diskdən götürülmüş verilənlər isə, burada yerləşdirilir. (Əgər program yenidən işə salınmalıdırsa, o, disk əvəzinə RAM-dan çağırıla bilər.)
<b>Wired</b>	- Ayrılmış yaddaş kernel-daxili strukturunda sistem çağırışları üçün istifadə olunur. Wired yaddaş heç vaxt swap-lanmış və page-lənmiş ola bilməz.
<b>Buf</b>	- Yaddaş buferinin həcmi göstərir. Bu bufer az vaxt önce diskdən oxunmuş verilənləri çap edir. Burada olan informasiyanın müəyyən bir qismi Active-dən, Inact-dan və Cache-dən olur. Tam ayrılmış kateqoriya deyil.
<b>Free</b>	- Bu sistem tərəfindən istifadə olunmayan boş RAM-in həcmidir.

7. **SWAP**: Bu isə sistemdə olan ümumi swap-in həcmini və nə qədər istifadə edildiyini göstərir.

Aşağıdakı sütunların strukturu isə top əmrin çıxışında olan prosesin strukturudur ki, biz də onu açıqlayırıq.

PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	C	TIME	WCPU	COMMAND
1211	root		1	20	0	68016K	5564K	select	0	0:01	0.00% sshd

<u>PID</u>	-	Hansı identifikasiator altında proses işləyir? PID
<u>USERNAME</u>	-	Hansı istifadəçi adı tərəfindən proses işə salınıb?
<u>THR</u>	-	Hal-hazırda işləyən daemonun sayı.
<u>PRI</u>	-	Hansı prosesə nə qədər prioritet verildiyini göstərir.
<u>NICE</u>	-	Eyni ilə prosesə prioritet verildiyini göstərir.
<u>SIZE</u>	-	Sistemin bu proses üçün nə qədər yer ayırdığını göstərir.

<u>RES</u>	- Hal-hazırda bu prosesin nə qədər memory-nin içinde olduğunu bildirir.
<u>STATE</u>	- Bu sütun prosesin hal-hazırda nə iş gördüğünü göstərir.
<u>C</u>	- Sistem prosessorun neçənci nüvəsini istifadə edir? <b>Qeyd:</b> Əgər fiziki prosessor sayı çoxdursa, onun da sayı ardıcılıqla olacaq. Məsələn: "5" olsa, bizim sistemdə 2 ədəd fiziki Quad prosessor var. 5 ikinci CPU-nun 1-ci nüvəsidir.
<u>TIME</u>	- Prosesin CPU-dan istifadə etdiyi ümumi vaxtı bildirir.
<u>WCPU</u>	- Bu proses üçün mərkəzi CPU-dan neçə faiz istifadə olunur? Prioritet üçün ən çox istifadə olunur. " <b>nice</b> "
<u>COMMAND</u>	- Burada isə işləyən programın adı olur.

Aşağıda, top əmri parametrlə işə salındıqda çox yararlı məlumatın rahat əldə edilməsini göstərir.

**top -osize** - Proseslərin istifadə etdiyi ümumi RAM həcmiini çap edir.

**top -mcpu** - İşləyən prosesləri CPU-ya görə çeşidləyir.

'top' əmrini daxil etdikdən sonra '?' işarəsini yazsaq, bütün imkanlar əks olunacaq.

**top -s 5** - 5 saniyə intervalı ilə dəyişiklikləri çap edin.

**top -U cavid** - Yalnız "**cavid**" istifadəçisinin proseslərini çap edin.

**top -S** - Digər proseslər kimi, sistem proseslərini də çap edin. Susmaya görə Swap və Page ekrana çıxmır, bu, onları çap edir.

**top -d 10** - Görüntünü **10** dəfə yeniləyin, sonra çıxın.

**top -b** - '**batch**' rejimi, bu rejimdə terminaldan daxil olan istənilən məlumata məhəl qoyulmur.

**top -b -d 12000 > myprocesslog &** - **Qeyd:** Əgər biz sistem resurslarının gün ərzində kim tərəfindən istifadə olunduğunu bilmək istəsək, onları jurnallayıb fayla yaza bilərik. Göstərilən misaldə top ekranı '**-b**' təhlükəsiz rejimdə, ekranı '**-d**' ilə, **12000** dəfə 2 saniyə intervalı ilə '**myprocesslog**' adlı faylinə arxa fonda işləyərək yazacaq.

**Qeyd:** top əmri daxil edildikdən sonra onlayn help-i açmaq üçün "?" simvolunu daxil etmək lazımdır.

## vmstat əmrinin açıllanması

**vmstat** əmri virtual yaddaşın statistikalarından hesabat almaq üçün istifadə edilir. CLI-dan vmstat əmrini daxil edirik və çıxışda əldə etdiyimiz nəticəni açıqlayırıq.

# **vmstat**

<b>procs</b>	<b>memory</b>	<b>page</b>	<b>disks</b>	<b>faults</b>	<b>cpu</b>	
r b w	a v m	f r e	f l t   r e   p i   p o	f r   s r   d a 0   d a l	i n   s y   c s	u s   s y   i d
1 0 0	2 0 4 M	9 0 4 M	2 4   0   0   0	2 5   0	0   0	1   6 0   2 3 5
						0   0   1 0 0

VMSTAT 6 sekiyadan ibarətdir.

"**Prosesler(proc),memory,page,disks,faults,cpu**"

procs - "r" proseslərin sayı, hansı ki, CPU vaxtını gözləyərək bloklanır. Bu proseslər işləməyə hazırlıdır, amma CPU-dan icazə ala bilmirlər. Əgər bu rəqəm böyükdürsə, CPU yüklüdür.  
- "b" proseslərin sayı, hansı ki, sistem INPUT/OUTPUT-u gözləyərək, disk yetkisini gözləyərək bloklanır. Bu proseslər öz verilənlərini aldığı anda yerinə yetiriləcək. Əgər bu rəqəm böyükdürsə, demək, diskimiz doludur.  
- "w" proseslərin sayı, hansı ki, yerinə yetiriləndir, amma tam çıkış olmuş hesab edilir. Əgər sistemdə daim çıxarılmış proseslər çox olursa, demək, bizim yaddaş gördüyüümüz işə uyğun deyil.

memory - "avm" istifadə olunan virtual yaddaşın orta rəqəm hesabı. Əgər bu rəqəm həddən artıq çoxdursa, demək, sistem SWAP-ı həddən artıq çox istifadə edir.  
- "fre" istifadəyə mümkün olan yaddaşın həcmi. Əgər bu rəqəm həddən artıq azdırısa, demək, sistemdə yaddaş çatışır.

page - "flt" uğursuz pageing-in rəqəmləri, hansı ki, lazımı informasiya RAM-da olmayıb, PAGE-dən və DISK-dən götürülməli idi.  
- "re" page-lərin sayı, hansı ki, düzəldilmiş və ya yenidən CACHE-dən istifadə olunmuşdur.  
- "pi" qısa görünüşdə: bu, RAM-dan SWAP-a köçürülecek page-lərin sayıdır.  
- "po" qısa görünüşdə: bu, SWAP-dan RAM-a köçürülecek page-lərin sayıdır.  
- "fr" saniyədə azad olunan page-lərin sayı.  
- "sr" saniyədə skan olunan page-lərin sayı.

## disks

- "**da0**" - Göstərilən hər iki ad disklerin adlarıdır. Ancaq diskin altında hər hansı bir rəqəmin olması şərtidir.
- "**da1**" - Saniyə ərzində diskdə aparılan əməliyyatların sayını çap edəcək.

**Qeyd:** Əgər bizdə disk sayı həddən artıq çox olsa, "**vmstat**" onların hamisini çap etməyəcək. Nəzərdə tutulub ki, "**vmstat**" maksimum 80 sütunu çap edə bilsin. Əgər bizdə disk həddən artıq çox olsa və hamisini görmək istəsək, səhifəni '**-n**' opsiyası ilə bölmək olar. "**vmstat -n 2**"

## faults

- "**in**" son 5 saniyə ərzində olan sistem gecikmələrinin sayı.
- "**sy**" son 5 saniyə ərzində olan sistem çağırışlarının sayı.
- "**cs**" vmstat keçidinin son təyinatlı uyğun rəqəmi.

## CPU

- "**us**" istifadəçilər nə qədər CPU-dan istifadə edib? (faizlə)
- "**sy**" sistem nə qədər CPU-dan istifadə edib?
- "**id**" nə qədər CPU istifadəsiz olub?

Aşağıdakı əmr ilə nüvə tərəfindən istifadə edilən yaddaşı çap edə bilərsiniz.

```
# vmstat -z
```

## **gstat açıqlanması**

Sərt diskin giriş-çıxış statusunu çap edir.

**gstat** - Bütün sərt disklerin INPUT və OUTPUT-u haqda ətraflı informasiya verir.

dT: 0.020s w: 0.000s									
L(q)	ops/s	r/s	kBps	ms/r	w/s	kBps	ms/w	%busy	Name
0	0	0	0	0.0	0	0	0.0	0.0	fd0

## **systat açıqlanması**

Sistem statistikalarını ekrana çap edir.

**systat -iostat - CPU** və sərt disk istifadəsini davamlı olaraq çap edin. (**numbers, bars, kbpt**)

- numbers** - Diskin I/O statistikasını rəqəmlərlə çap edin.
- bars** - Diskin I/O statistikasını graph-la çap edin.
- kbpt** - Hər tranzaksiyanı kilobaytlarla çap edin.  
(Kilobyte per transaction)

**systat -ifstat** - Bütün şebəkə kartlarının Trafiklə (input və output), həcmi (KB, MB) və ümumi statusunu çap edir.

**systat -netstat** - Şebəkə statusunu yüklenmə aralıqları ilə çap edir.  
**all** - Bütün net statistikasını çap edir.

**systat -pigs** - Susmaya görə olan systat çıxışını çap edir.

**systat -icmp** - Yalnız ICMP paketlərinin statusunu çap edir.

**systat -ip** - ICMP statistikası ilə eynidir, IP və UDP statistikasını çap edəcək.

**systat -tcp** - Yalnız TCP paketlərinin statusunu çap edir.

**systat -swap** - SWAP-in sərt diskdə istifadə olunan statusunu çap edir.(Bloklarla)

**systat -vmstat** - Tam detalllı şəkildə həddən artıq böyük açıqlamalarla, **Real Memory** və **Virtual Memory**, sərt disklərin **I/O** statusu haqqında məlumat çap edir.  
**boot** - Sistem boot olduqdan sonra olan bütün statistikaları çap edir.  
**run** - Bu əmr daxil edilən andan işləyən statistikanı çap edir.  
**time** - Təyin edilmiş vaxt intervalı (saniyelərlə) ilə statistikanı yeniləyir.  
**zero** - Hal-hazırkı statistikanı reset edin.

**Qeyd:** systat-ın tam imkanlarını görmek üçün əmri daxil etdikdən sonra "**SHIFT+:**" və **help** sözünü daxil etsək, istifadə edilə biləcək bütün sütunları görə bilərik.

## Split

Test üçün bir fayl yaradın və içində 3000 sətirlik bir məlumat yazın.

İşinizi asanlaşdırmaq üçün **writer.sh** adlı skript yaradıb içində aşağıdakı sətirləri əlavə etsəniz, faylı əldə etmiş olacaqsınız. Ancaq **writer.sh** faylini yerinə yetirilən etməyi unutmayın.

```
#!/usr/local/bin/bash

for i in `seq 1 2000`
do
    echo "$i, Salam necesen?" >> /root/splitfile
```

<code>echo "\$i, Cox saq ol, sen necesen" &gt;&gt; /root/splitfile</code>	
<code>done</code>	
<code>split splitfile</code>	- split əmri splitfile faylini hər biri 1000 sətir olan ayrı-ayrı fayllara ayırır.
<code>split -l 3 splitfile</code>	- Əmr splitfile faylini hər birində 3 sətir olan çoxlu hissələrə ayıracaq.
<code>split -b 100M access.log access.log</code>	- Həcmi <b>11GB</b> olan <b>access.log</b> faylini hər biri <b>100MB</b> olmaqla çoxlu hissələrə bölgəcək. Eyni ilə <b>-k</b> kilobayt, sadəcə rəqəmin yazılması isə baytdır.
<code>split -l 3 splitfile bolunenler</code>	- Splitfile faylini hər birində <b>3</b> sətir olan və bölünənlər adı ilə başlayan fayllara bölgəcək.
<code>split -l 3 -a 1 splitfile F</code>	- Eyni ilə hər birində 3 sətir olan fayllara ayıracaq, ancaq yaratdığı fayllar yalnız 2 simvollu adla olacaq.
<code>split -l 3 -d splitfile F</code>	- Eyni ilə hər birində 3 sətir olan fayllara ayıracaq, ancaq yaratdığı fayllar yalnız <b>-d</b> rəqəmlü suffixlə yaranacaq.
<code>split -n 2 -a 1 -d splitfile F</code>	- <b>-a 1</b> simvollu və <b>F</b> adı ilə başlayacaq, <b>-n 2</b> iki yerə bölgəcək splitfile faylini, <b>-d</b> isə adlar rəqəmlərlə başlayacaq.
<u>CPU temperaturunun monitoring edilməsi</u>	
<code>/usr/ports/sysutils/lmmon</code>	- CPU temperaturuna baxmaq üçün portu yükleyirik.
<code>lmmon -i /dev/smb0</code>	- Bu əmrlə baxırıq.
<code>/usr/ports/sysutils/consolehm</code>	- CPU temperaturuna baxmaq üçün portu yükleyirik.
<code>chm -I /dev/smb0</code>	- Bu əmrlə baxırıq.

```
/usr/ports/sysutils/healthd
```

- CPU temperaturuna baxmaq üçün portu yükleyirik.

```
cpuset -C -c -l 0,2 -p 1701
```

- Əmrin sayesində 1701 PID-li prosesi 1-ci prosessorun 2-ci core-unda işə salırıq.

# İşlək sistemdə Swap həcminin artırılması

İlk olaraq SWAP haqqında nəzəri məlumat vermək istəyirəm ki, bütün suallara aydınlıq getirək. Ümumiyyətlə, SWAP dedikdə virtual yaddaş nəzərdə tutulur. Bu virtual yaddaş HDD-dən əldə edilən müvəqqəti ayrılmış bir hissədir, hansı ki, RAM-a da oxşayır. Əvvəlki nəzəriyyəyə əsaslanısaq, SWAP həcmi RAM-in 2-yə vurulmasından alınmalıdır. Ancaq bu məntiq özünü doğrultmadı, çünki 1TB-lıq RAM olan serverlər yarandı və 1TB RAM-ı 2-yə vurduqda 2TB SWAP yaratmaq ehtiyacı ortaya çıxdı. Təsəvvür edin ki, sistem özü verilənlər bazası ilə birlikdə 1TB-lıq diskdə işləyir və 2TB-lıq SWAP həcm diskdə boş yerə sərf edilir. Hal-hazırda SWAP həcmi təyinatı inzibatçı tərəfindən hesablanıb tələbə uyğun olaraq təyin edilir. Ləp köhnə UNİX və Linux əməliyyat sistemlərində virtual yaddaşa yüksələn məlumatlar ya sərt olaraq HDD-də, ya da sərt olaraq RAM-da saxlanılırdı. Ancaq təsəvvür edin ki, sizin proses ata prosesdir və onun 4 ədəd bala prosesi mövcuddur. Bu bala proseslərdən 2-si işlək vəziyyətdə, 2-si yatmış vəziyyətdə olarsa, köhnə prinsiplə, yəni swapping prinsipi ilə bu proseslərin statusu nəzərdən keçirilməyəcək və sərt olaraq ya RAM-a, ya da RAM limitinə çatarsa, birbaşa SWAP-a atılması idi. Bu, düzgün məntiq deyildi, çünki bu prosesləri hansısa bir identifikatora görə təyin etsək, onların SWAP və ya RAM-da saxlanması təyinatı çox asanlaşacaq. Ona görə paging məntiqi yarandı, hansı ki, prosesin RAM-da və ya SWAP-da olacağı təyinatını onun statusuna görə müəyyən edir. Yəni PİD-də işlək vəziyyətdə olarsa, RAM-da işləməlidir ki, sürətli nəticə versin. Həmçinin bu proses sonra yatmış statusu alarsa, o, RAM-da boşuna yer tutmalı deyil və səliqəli şəkildə SWAP-a köçürülməlidir.

**PAIGING baş verir:**

FreeBSD işleyen programın müəyyən hissəsini atır PAIGING-ə. Bu işi ona görə sürətləndirir ki, sistem böyük yük altında işleyəndə prosesin istifadə olunmayan bitləri nə vaxtsa lazım olanadək page-də saxlanıla bilsin. FreeBSD o zaman işlek code üçün RAM-dan istifadə edir. Bu, ən çox verilənlər bazasında istifadə olunur, verilənlər bazasının start kodunu o, işləməyənədək swap-da saxlayır.

**SWAPPING baş verir:**

Swapping-də isə əgər bir proses üçün fiziki RAM-da yer çatmırsa, həmin anda yerinə yetirilməyən prosesi o, tam olaraq SWAP-a yerləşdirir. Yenidən bu proses lazım olduqda isə, o, tamam başqa bir proseslə həmin prosesi çağırır. Bu, ilk olaraq bir az adı görünə də, elə deyil. Çünkü sistemə müraciət sayı çox olanda bu iş prinsipi sistemi tamam çökdürə bilər və ya fiziki yaddaş artırmasına gətirib çıxarar.

**Qeyd:** Kernel-də "md" modulunun olmasını dəqiq bilməliyik. Hələ ki siz FreeBSD kernelin necə kompilyasiya edilməsi və yüklənmə qaydasını bilmirsiniz. Növbəti başlıqlarımızda bunun haqqında ətraflı məlumat əldə edəcəksiniz. Ancaq istənilən halda "md" modulu GENERIC kernel-də susmaya görə pseudo alətlər bölümündə mövcud olur.

```
device    md      # Memory "disks"
```

İndi isə işimizə başlayaq.

Ardıcılıq:

1. `dd if=/dev/zero of=/usr/swap0 bs=5M count=64`

`if` - əgər /dev/zero-sa

`of` - getsin /usr/swap0 faylinə

`bs` - bayt size (bayt həcmi 5 megabayt)

`count` - sayı 64 dəfə(5\*64=320 edir)

2. `chmod 0600 /usr/swap0`

3. `/etc/rc.conf` faylinin sonuna aşağıdakı sətri əlavə edirik.

`swapfile="/usr/swap0"`

4. `swapinfo -h`

    - Əmr swap slice-i çap edir.

    "-h" human readable(normal oxunuş) deməkdir.

Ya **reboot** edirik, ya da anında işletmək üçün 5-ci sırada olan əmri işə salırıq.

5. **mdconfig -a -t vnode -f /usr/swap0 -u 0 && swapon /dev/md0**

Və CLI-dan real seansımızda çalışdırıq.

**mdconfig -a -t vnode -f /usr/swap0 -u 0 && swapon /dev/md0**

- a - Memory diskini qəbul edin.
- t - Memory diskinin tipini seçin.
- f - Hansı fayldan oxuyacaq?
- u - Hansı unite 0-ci (Yəni /dev-də md0-ci device yaradacaq)

**Qeyd:** Əgər biz sistemə 1024 megabayt RAM vermişiksə, onda bizim SWAP 2024 megabayt olmalıdır. İndiki halda isə biz ümumi swap-ı top əmrinin nəticəsində görə bilərik.

Swap: 2330M Total, 2330M Free

Ya da '**swapinfo -h**' əmri ilə aşağıdakı ümumi nəticəni əldə etməliyik.

Total	2385960	0B	2.3G	0%
-------	---------	----	------	----

Bir neçə lazımlı əmri qısaca açıqlayaq:

- |                         |  |
|-------------------------|--|
| <b>last</b>             | - Bütün qoşulan istifadəçiləri göstərir.                         |
| <b>last -h 10.0.0.1</b> | - '10.0.0.1' IP ünvandan gələn bütün qoşulmaları çap edir.       |
| <b>which</b>            | - Sistem əmrinin hansı sistem ünvanından çağırıldığını göstərir. |

# Portlar, paketlər və onların idarə edilməsi

Əməliyyat sistemine üçüncü tərəfin yazdığı program təminatını yüklemek istədikdə bizim köməyimizə portlar və paketlər çatacaq. Bu başlıqda biz portların və paketlərin istifadə qaydalarını ətraflı araşdıracaqıq.

**Qeyd:** Nəzərə alın ki, port və paketlərdən istifadə edib hansısa program yüklemek üçün siz artıq serverin internetə çıxışını təmin etməlisiniz. Lakin bu mövzu şəbəkə quraşdırılmalarını qabaqlayır və öncə şəbəkə başlığını oxuyub, quraşdırmadan sonra bu mövzuya qayıtmaq lazımdır.

**Qeyd:** Əgər programı CD/DVD-dən yükleyirsinizsə, release-lərin özləri avtomatik olaraq CD və ya DVD-ni mount edəcək. Yenə də özümüz mount etsək, mütləq '/**dist**'-ə eləməliyik, çünki **sysinstall** (FreeBSD8.4) və **bsdconfig** (FreeBSD9.3 və FreeBSD10.1) susmaya görə '/**dist**' qovluğununa müraciət edir.

İki metodika ilə port kolleksiyasını yükleyək. Ancaq bunun üçün minimal formada yüklenmiş sistem olmalıdır ki, orada ports olmasın. Ya da özünüz **/usr/ports** qovluğunu əlinizlə silməlisiniz.

## Birinci metodika

FreeBSD 6.0-cı release-dən başlayaraq, artıq portsnap sistemin özündə susmaya görə olur.

**/etc/portsnap.conf**

- Portsnap-in default quraşdırma faylı.

**sysinstall -> Configure -> Distributions -> ports** - Sysinstall ilə də yükləyə bilərik.

**portsnap fetch**

- Bu əmr Ports kolleksiyasını "**/var/db/portsnap**" qovluğuna yükleyir. Yəni əgər biz başqa serverdə "**portsnap fetch**" əmri daxil etmişiksə, həmin serverdən **/var/db/portsnap** qovluğunu öz serverimizdə eyni ünvana atsaq və aşağıdakı əmrləri yığsaq, kifayətdir.

**portsnap extract**

- Yeni port kolleksiyasını "**/usr/ports**"-a yükleyir və açır.

**portsnap update**

- Yüklənmiş "**/usr/ports**"-da olan port kolleksiyasını yeniləyir.

**Qeyd:** Əgər portsnap-da belə bir "Fetching public key from isc.portsnap.freebsd.org... failed" səhv çıxsa, o zaman siz hansısa bir proxy-dən çıxırsınız və ekrana mesaj çıxır. Cavabı bu əmrlə əldə eləmək olar. '**fetch -vvv http://update3.freebsd.org/10.1-RELEASE/amd64/pub.ssl'**'

## İkinci metodika

**mkdir /usr/ports**

- Minimal yükləmişiksə, bu, qovluq olmayıcaq, ona görə də qovluğu yaradırıq.

**pkg\_add -r cvsup-without-gui**

- CVSUP-ı qrafik interfeysiz yükleyirik (FreeBSD8.4).

**pkg install fastest\_cvsup-0.3.0**

- FreeBSD10.1 üçün ən yaxın CVSUP reposoların tapılması üçün bu paketi yükleyirik.

**pkg install binary-cvsup-static-16.1h** - FreeBSD10.1 maşın üçün cvsup yükleyirik.

```
fastest_cvsup -c all
```

- FreeBSD10.1 maşın üçün cvsup repos serverlərin bütün siyahısını sizə çap edəcək. Əldə etdiyimiz nəticədən ən yaxın olanı seçirik. Mənim seçimim **cvsup6.ru.freebsd.org** idi. Bu seçimi **/root/supfile** faylımızda qeyd edirik.

**/root/ports-supfile** faylı yaradırıq və içində aşağıdakı məzmunu əlavə edirik:

```
*default host= cvsуп6.ru.freebsd.org  
*default base=/var/db  
*default prefix=/usr  
*default release=cvs tag=.  
*default delete use-rel-suffix  
*default compress  
ports-all
```

```
cvsup -L 2 /root/ports-supfile
```

- Burada "**/root/ports-supfile**" faylına müraciət edilir və quraşdırmasına uyğun olaraq müraciət edilir. '-L' 2 tam detallı şəkildə hər yenilənən paket haqqda informasiya çap edin. Ancaq **-h cvsup. FreeBSD.org** yazaraq yenilənmə mənbəyini əlimizlə təyin edə bilərdik.

**Qeyd:** Unutmayın ki, bu metod daha çox vaxt alır. cvsup əmrində qoşulma yetkisi ilə bağlı aşağıdakı səhv çıxsa, fikir verməyin və əmri yenidən daxil edin.

```
Rejected by server: Access limit exceeded; try again later  
Will retry at 13:53:30
```

## Portların daxili strukturu və yükləmə qaydaları

```
whereis lftp
```

- İstədiyimiz programın əsas düzgün ünvanını tam göstərir.

```
cd /usr/ports/ftp/lftp/  
ls -CF
```

- LFTP ftp clientin port ünvanına daxil oluruz.  
- İçindəkiləri siyahılıayıraq.

Makefile

distinfo

files/

pkg-descr

pkg-plist

**Makefile**

- "make" əmri paketin necə kompilyasiya olunması haqda informasiyani bu fayldan oxuyur.

**distinfo**

- Program portlardan endiriləndə onun checksumu SHA256 və ya MD5-də bu fayldan oxunur ki, portun düzgünlüyü yoxlanılsın. Çünkü endirilən paket serverimizə tam çatmaya bilər. Bunun sayəsində onun həmin paket olduğu təsdiq edilir.

**files/**

- Bu qovluqda yüklemək istədiyimiz program üçün "patch"-lar olur ki, paketi fix-ləyir və ya təhlükəsizlik problemini həll edir.

**pkg-descr**

- Bu faylda paket haqda açıqlama və adətən paketi yazmış saytin ünvanı olur.

**pkg-plist**

- Bu fayl paketdən yüklenən faylları identifikasiya ələyir və local fayl sistemdə hansı ünvanlarda yükləyəcəyini təyin edir. Həm də öz əmrlərinin çağırıldığı ünvanların linkini yazar.

**make build**

- Paketi rəsmi saytından endirin və "**build**" edin (yəni kompilyasiya).

**make install**

- Paketin binar fayllarını və sənədlərini yükləyin. Bu, paketin sıxılmış man səhifələrini, registerlərini və fayllarını fayl sistemdə, lazımi ünvanlarına yerləşdirir.

**make clean**

- "**build**"-in nəticəsində yaranan faylları təmizləyir.

**Qeyd:** "**make install**" əmri işə düşən kimi paketin susmaya görə olan qovluğunda "**work**" adlı başqa qovluq yaradır. Və "**/usr/ports/distfiles**"-dan özünə aid olan paketi bu qovluqda açır, "**make clean**"-də install-in işi bitəndə work qovluğunu silir.

**make checksum**

- Paketin checksum uzunluğu "**distinfo**" faylı ile müqayisə edilib yoxlanılır.

**make configure**

- Quraşdırma porsiyasının build olması üçün bu əmrə müraciət edin.

**make deinstall**

- Paketi silin.

**make package**

- Yüklənmiş mənbə kodu kompilyasiya edib, yerləşdiyi ünvanda hazır paket düzəldir.  
Məs: "**lftp-4.3.1.tbz**". Yəni biz bu hazır paketi artıq "**pkg\_add**"-la yükləyə bilərik.

**make fetch**

- Port ünvanında yerləşdiyimiz paketin sıxlımsız vəziyyətdə olan mənbə kodu "**/usr/ports/distfiles**" ünvanına endirin.

**make fetch-recursive**

- Bu paketin özünü və onun asılılığında olan bütün paketləri "**/usr/ports/distfiles**" ünvanına endirin.

**make readmes**

- "**/usr/ports**" ünvanında bu əmr daxil ediləndə bütün paketlər haqqında informasiyanı təşkil edən **README.html** adlı fayl yaradılır.

**Qeyd:** Həm ümumi '**/usr/ports**' ünvanı haqda, həm də hər bir port üçün bütün portların öz ünvanlarında '**README.html**' adlı fayl yaradır. Əgər bir port haqqında readme yaratmaq istəyirsizsə, onda mütləq həmin portun öz ünvanına daxil olub, orada bu əmri daxil etməyiniz lazımdır.

**make search name='apache'**

- Bu əmr "**/usr/ports**" ünvanında daxil edildikdə "**apache**" adlı bütün portları axtarış edir.

**make install clean -C /usr/ports/www/nginx**

- Burada make-ə deyirik ki, '**-C**' get **makefile**-i '**nginx**' portunun qovluğundan oxu, birbaşa oradan da kompilyasiya et və yüklə.

```
make -j5 all install clean
```

- Yüklədiyimiz port maksimum eyni vaxt üçün 5 sistem CPU istifadə edə bilər.

```
make -DBATCH install clean
```

- Yükləmək istədiyiniz port-un bütün asılılıqlarında olan portları susmaya görə qəbul edin və soruşmayın.

```
make all-depends-list
```

- Yükləmək istədiyiniz port-un bütün asılılığında olan portların siyahısını çap edin.

```
/usr/ports/Mk/bsd.port.mk
```

- make əmri üçün lazım olan bütün arqument siyahısını bu faydan oxuya bilərsiniz.

**Qeyd:** Biz portları müəyyən quraşdırımlar üçün susmaya görə olduğu kimi təyin edə bilərik.  
Məs: Əgər bizim sistemimizdə istifadə olunacaq bütün bazalar "MySQL"-dirsə, biz onu susmaya görə təyin edə bilərik. Yəni, artıq hər dəfə baza ilə işləyən program yüklenikdə, o, bizdən soruşmayacaq, hansını istəyirik, avtomatik olaraq özü seçəcək. Bunun üçün onu '/etc/make.conf'-a WITH\_MYSQL=YES əlavə etməliyik. Artıq hər dəfə portlardan program yüklenikdə o, make.conf-u oxuyacaq və sonra qərar verəcək.

**Qeyd:** Əgər biz program təminatına tam inanrıqsız və bütün xəbərdarlıqlara baxmayıaraq, onu yüklemək istəyiriksə, aşağıdakı qaydada edirik.

```
make install DISABLE_VULNERABILITIES=YES
```

- Portun təhlükəli boşluqlarına fikir vermədən yükləyəcək.

Seçilmiş Portun gündəmdə qalması üçün müxtəlif program paketlərinən istifadə edə bilərik.

Məsələn: **portupgrade**. Aşağıda onu yükləyib açıqlayıraq.

```
cd /usr/ports/ports-mgmt/portupgrade
```

- Portupgrade-in port ünvanına daxil oluruq.

```
make install clean
```

- Və yükləyirik.

```
portupgrade
```

- R bash - "bash" paketine aid olan portu və paketi yeniləyirik. Yəni avtomatik olaraq, bu porta aid olan portu internetdən çəkib yeniləyirik. Sonra köhnə versiyanın portunu rezerv nüsxə edir, silir və yeni versiyanın portunu kompilyasiya edib paketini sistemə yükləyir.

Bütövlükde müəyyən bir port skeletini yeniləmək üçün növbəti paketdən istifadə edirik.  
**cd /usr/ports/ports-mgmt/portmanager** - Portun ünvanına daxil oluruq.  
**make install clean** - Portu yükleyirik.

**portmanager -u** - Sistemdə olan bütün yüklənmiş portları yeniləyin.

**portmanager ftp/lftp** - "lftp" clientin portunu yeniləyin. Yeni "lftp"-nin seçilmiş portunu yeni versiyaya yeniləyir. Bu da o deməkdir ki, sistemə lftp-nin ən yeni versiyası yüklənir.

### **Yüklənmiş portların audit edilməsi**

Aşağıda göstərilən program təminatından istifadə edirik.

**Qeyd:** Aşağıda sadalanan əmrlərin hamısı artıq **FreeBSD10.1** və **FreeBSD9.3**-də **pkg** paketinin üstünə miqrasiya edilmişdir.

**pkg audit -F** - Bu əmr FreeBSD10.1 və FreeBSD9.3-də bütün paketləri audit edəcək.

**cd /usr/ports/ports-mgmt/portaudit** - Portun ünvanına daxil oluruq. Portaudit paketi öz bazasını **"/usr/local/etc/periodic/security"** qovluğunda saxlayır. Portaudit bu qovluqda yerləşən faylda bütün təhlükəsizlik yenilənmələrini saxlayır. Bu faylin yenilənməsi ilə **freebsd.org**-un təhlükəsizlik komandası məşğul olur.

**portaudit -Fda** - Bu əmrlə ən yeni təhlükəsizlik bazası serverə yüklenir və sistemdə olan portlar o baza ilə müqayisə olunub, auditin nəticəsi çap olunur.

**portaudit -a** - Yüklənmiş audit bazası ilə sistemdə olan bütün portları audit edir və çap edir.

**portaudit -f /usr/ports/mail/roundcube** - Seçdiyimiz konkret portun boşluqlarını audit edib çap edir.

**Qeyd:** Bu əmrin nəticəsində port-un özü yeni ola bilər. Ancaq yüklenmiş paket köhnə olsa, nəticə bəlli olmur.

**Qeyd:** Hər dəfə yeni port yükləndikdə, əgər biz "make clean" əmri yiğmamışıqsa, həmin paketin müvəqqəti faylları sistemdə yer tutacaq. Onları avtomatik silmək üçün "portsclean" əmrindən istifadə edirik. "portsclean" paketi avtomatik olaraq "portupgrade" paketinin daxilində gəlir.

**portsclean -D**

- İstənilən program paketinin asılılığında olan paketləri əgər başqa program təminatı tərefindən istifadə olunmursa, "/usr/ports/distfiles" qovluğundan silin.

**portsclean -DD**

- Heç bir program paketinin asılığında olmayan paket varsa, onu "/usr/ports/distfiles"-dan silin.

**portsclean -C**

- Sistemdə olan bütün portların silinməmiş "work" qovluqlarını silir.

**Qeyd:** Heç bir paketə linklənməyən paketləri tapmaq istəsək, "pkg\_cutleaves" paketindən istifadə etmək lazımdır.

**cd /usr/ports/ports-mgmt/pkg\_cutleaves  
make install clean**

- Paketin ünvanına daxil olub,  
- yükləyirik.

**pkg\_cutleaves -lc**

- Əlaqəli olmayan bütün paketləri açıqlamaları ilə çap edin.

**pkg\_cutleaves**

- Asılılıqda olmayan və silinməsi lazım olan paketi göstərin. "Keep" saxlayın və ya "Delete" silin.

**Qeyd:** Artıq FreeBSD10.1 və FreeBSD9.3 üzərində **pkg\_info**, **pkg\_add**, **pkg\_delete** əmləri demək olar ki, eyni opsiyalarla **pkg info**, **pkg add/install** və **pkg delete** əmləri ilə əvəz edilmişdir.

```
pkg_info -a
```

- **pkg\_delete** emri ile silmek lazım olan programın düzgün versiyasını gösterir. Hem de burada silmek istediğiniz programın diğer hansı programlar tərəfindən tələb edildiyini görə bilərsiniz. Qeyd: Artıq FreeBSD10.1 və FreeBSD 9.3-də bu emr **pkg info** olmuşdur.

```
pkg_info | less
```

- Bütün paketlərin ad və versiyasını çap edir. 'less' page edir.

```
pkg_info -I curl*
```

- "curl" adı ilə başlayan bütün paketləri çap edir.

```
pkg_info curl*
```

- 'curl' adı ilə başlayan paket haqqında detallı informasiya verir.

```
pkg_info -d curl*
```

- 'curl' adı ilə başlayan paketin yalnız 'Description:' bölümünü çap edir.

```
pkg_info -c curl*
```

- 'curl' adı ilə başlayan paketin yalnız 'Comment:' bölümünü çap edir.

```
pkg_info -f curl*
```

- 'curl' adlı paketin içinde olan faylları siyahiya alır.

```
pkg_info -L curl* | less
```

- 'curl' adlı paketin içinde olan faylların tam ünvanlarını siyahiya alır.

```
pkg_info -i bash* | less
```

- 'bash' adlı paketin install skriptini çap edir.

```
pkg_info -k bash* | less
```

- 'bash' adlı paketin deinstall skriptini çap edir.

```
pkg_info -R libiconv* | less
```

- Hansı paketler 'libiconv' paketini tələb edir?

```
pkg_info -r bash* | less
```

- 'bash' paketinin tələb etdiyi bütün paketləri çap edir.

```
pkg_info -W apachectl
```

- 'apachectl' emri hansı program paketi tərəfindən sistemdə əmələ gəlməsini bildirir.

**Qeyd:** `pkg_add`-la portsnap-in fərqi ondan ibarətdir ki, portsnap program haqqında olan məlumatı daha ətraflı və daha dəqiq çatdırır.

`pkg_add -r -f apache13`

- Paketlərin içinde seçdiyimiz programı şəbəkədən yükleyir. (düzgün versiya)  
(`-f "force"` – məcburi yüklə)

`pkg_add mysql-server-5.1.55.tbz`

- '`mysql`' paketini yerləşdiyimiz qovluğun içindən götürərək yükleyirik.

`pkg_add -r -K apache22`

- '`-r`'-lə internetdən çəkdiyimiz `apache22`-ni '`-K`' ilə yerli qovluğumuzda yadda saxlayıb, sonra yükleyirik.

`pkg_add -v hal-0.5.14_12.tbz`

- '`hal-0.5.14_12.tbz`' adlı paketi yerli qovluqdan '`-v`'-ilə verbose rejimi ilə yükleyir.

`pkg_add -M memdump-1.01.tbz`

- '`-M`' opsiyası paketi yüklemək əvəzinə onun daxilini '`/var/tmp`' ünvanına "`instmp.xxxx`" adı ilə yazar.

`pkg_delete -n dbus-0.93_3`

- Yoxlayaq görək, '`dbus`' paketini sildikdə, nələr baş verə bilər? Yəni hansı paketlərin işləməsi üçün bu paket tələb olunur?

`pkg_delete apache-2.2.15_9`

- Göstərilən versiyalı apache-i silir.

`pkg_delete -i akode*`

- Hər bir '`akode`' adı ilə gələn paketi sildikdə '`-i`' opsiyası ilə təsdiqlənmə menyusu çıxacaq.

`pkg_delete -v xmunes*`

- '`xmunes`' adı ilə başlayan bütün paketləri verbose modda siləcək.

`pkg_delete -f metacity*`

- '`metacity`' adlı paketi heç bir dependency-sinə baxmayaraq, tamamilə silin.

`pkg_delete -a`

- Diqqətli olun, əmr bütün yüklenmiş paketləri silir.

```
pkg_delete -r gstreamer*
```

- 'gstreamer' adlı paketi bütün asılılıqları ile birlikdə tamamilə silir.

```
pkg_version -v
```

- Yüklenmiş bütün paketlerin gündemde olub-olmadığını yoxlayıb çap edir.

```
pkg_version -v -s cvsup
```

- Yalnız yüklenmiş cvsup paketinin gündemde olduğunu yoxlayır.

```
pkg_version /tmp/INDEX-7
```

- Susmaya göre "pkg\_version" emri "/usr/ports/" ünvanında olan "INDEX-6" faylından paketlerin gündemini yoxlayır. Ýgər biz başqa index faylini istəsək, məs., "INDEX-7" freebsd.org saytından endirib, tmp qovluğundan yoxlaya bilərik.

**Qeyd:** Hər bir yüklenən paketin aktiv opsiyalarını port bazasının içində görmək olar.

```
less /var/db/ports/nginx/options
```

- Burada 'nginx' paketinin aktiv opsiyalarını çap edirik.

Siz həmçinin ISO DVD nüsxənin içindən də aşağıda göstərilən ardıcılıqlarda getsəniz, müəyyən qisim paketləri yükleyə bilərsiniz. Ancaq diskı özünüz mount etməyin, çünki sistem özü bunu etməyə çalışır.

```
sysinstall -> Configure -> Packages -> CD/DVD -> www -> apache22
```

# Fayl/qovluqların arxivlənməsi, sıxılması və grep sətir preprocessoru

## Tar, Gzip, Bzip2, Zip

İmkanlar:

Qovluq və faylları bir faylda sıxıb saxlamaq.  
GZIP-lə başlayaqq. Gzip tipli faylları sıxaqq və açaqq.

İşimiz:

**forgzip** faylini GZIP-i istifadə edərək sıxaqq

**gzip -c forgzip > forgzip.gz**

- **forgzip** faylini sıxıb **forgzip.gz** faylinə yazacaqq.

**gzip myfile**

- "myfile" faylini sıxin və adını dəyişin "myfile.gz"-yə.

**gzip -v myfile**

- "myfile" faylini sıxin və adını dəyişin "myfile.gz"-yə, "-v" verbose mod.

```
gzip -tv myfile.gz
```

- "-v" verbose eləmək üçün '-t' təyin edir ki, file gizp-lə sıxılıb, ya yox.

```
gzip -lv file3.txt.gz
```

- "-v" verbose eləmək üçün, '-l' sıxılmış file haqqında detallı məlumat çap edir.

```
gzip -rv /home/salman
```

- "/home/salman" ünvanında olan bütün faylları sıxın, həmçinin həmin qovluğun içində olan digər qovluqların içində olan bütün faylları (yalnız faylları) sıxacaq. '-r' recursive (folder, subfolder)

```
gzip -l myfile
```

- "myfile" adlı faylı ən aşağı səviyyədə '-l'-ə sıxın (tez vaxtda az sıxır).

```
gzip -9 myfile
```

- "myfile" adlı faylı ən yuxarı səviyyədə '-9'-ə sıxın (gec vaxtda çox sıxır). Qeyd: Gzip susmaya görə 6-ci səviyyədə sıxır.

Gzip ilə sıxılmış faylin içini çap edək(sıxılmış faylı açmadan)

```
zcat forcompress.gz
```

- Bu əmr forcompress.gz faylini açmadan adı cat əmri kimi sadəcə faylin kontentini çap edir.

**Gunzip** və ya **Gzip**-i istifadə edərək faylı decompress edək.

```
gunzip forcompress.gz
```

- decompress edirik.

İndi isə eyni işi **BZIP** üçün edək.

```
Bzip2 istifadə edərək forcompress faylini sıxaq.
```

```
bzip2 -c forcompress > forcompress.bz2
```

- forcompress.b2 faylinı sıxırıq.

```
bunzip2 və ya bzip2 istifadə edərək forcompress.gz faylini açaq.
```

```
bunzip2 forcompress.bz2
```

- forcompress.b2 faylini açmadan sadəcə çap edirik.

```
bzip2 -d forcompress.gz
```

- faylı açırıq.

Eyni vaxtda Bzip2 və Gzip ilə sıxaraq arxiv işini də TAR ilə görək. Arxiv yaradaq (ora bir və ya bir neçə qovluq və ya fayl ataq).

Arxiv olmayan forcompress adlı qovluğu və içinde olan bütün məlumatları öncə arxiv edək. Sonra isə sıxaraq arxiv edək.

**tar -cvf forcompress.tar forcompress**

- **forcompress** adlı qovluğu bütövlükdə **forcompress.tar** adlı arxivə əlavə edirik.

**tar -cvf various.tar \***

- Yerləşdiyimiz ünvanda olan bütün məlumatları **various.tar** faylinə arxiv edirik.

**tar -czvf forcompress.tar.gz forcompress**

- **forcompress** adlı qovluğun içindəki bütün məlumatları GZIP-lə sıxdıqdan sonra onları **forcompress.tar.gz** adlı faylda arxivləşdiririk.

**tar -cjvf forcompress.tar.bz2 forcompress**

- **forcompress** adlı qovluğun içindəki bütün məlumatları BZIP-lə sıxdıqdan sonra onları **forcompress.tar.gz** adlı faylda arxivləşdiririk.

Tar və compress etdiyimiz faylların açılmasına baxaq:

**tar -xjvf forcompress.tar.bz2**

- BZIP-lə sıxılmış **forcompress.tar.bz2** adlı faylı həm açırıq, həm də arxivdən çıxarıraq.

**tar tvf all.tar**

- 'all.tar' faylinin (-t yalnız **tar** faylı) kontentini siyahıya alın.

**tar tzvf myfiles.tgz**

- 'myfiles.tgz' adlı, 'z' gzip-də sıxılan və 't' **tar**-da arxivlənən faylin məzmununu çap edin.

**tar rvf archive.tar myfile**

- 'myfile' faylini "archive.tar" faylinə əlavə edin.

BZIP və GZIP ilə gördüğümüz bütün işləri, həmçinin LZOP adlı programla da görə bilərik. Ancaq LZOP susmaya görə sistemdə yüklenmiş olmur və biz onu sistemə yükləməliyik.

**pkg install lzop**

- lzop-u yükleyirik.

```
tar -cf - *.txt | lzop > myfiles.tar.lzo
```

- Yerləşdiyimiz ünvanda bütün "txt" genişlənməli faylları tar-la arxivləyib ötürürük "lzop"-a və o da öz növbəsində "myfiles.tar.lzo" faylına ötürür.

```
lzop -d < myfiles.tar.lzo | tar -xf -
```

- "myfiles.tar.lzo" faylıni lzop-la açıb ötürürük tar-a ki, o da öz növbəsində arxivini açır.

```
lzop -v myfile
```

- "myfile" faylıni silmədən "myfile.lzo" faylına sıxın.

```
lzop -U myfile
```

- "myfile" faylıni "myfile.lzo" faylına sıxın və sonra orijinal faylı silin.

```
lzop -t file1.lzo
```

- "file1.lzo" faylıni '-t' opsiyası ilə test edin.

```
lzop --info file1.lzo
```

- "file1.lzo" faylı haqda məlumat verin.

```
lzop -l file.lzo
```

- "myfile.lzo" faylinin sıxılması haqqında məlumatı çap edin.

```
lzop --ls file.lzo
```

- Sıxılmış "file.lzo" faylinin məzmununu 'ls -l' kimi çap edin.

```
lzop -dv file.lzo
```

- "file.lzo" faylinin özünü saxlayaraq açın.

**cat** əmrinin **STDOUT**-a çap edilməsində bir neçə gərəkli misallar çəkək.

```
cat test
```

- test adlı faylı tam olaraq ekranə çap edin.

```
cat test > new
```

- test adlı faylin içini çap edin və nəticəni "new" adlı fayla yazın, ancaq öncə new faylini təmizləyin, sonra yazın.

```
cat test >> new
```

- test adlı faylin içini çap edin və nəticəni "new" adlı faylin sonuna yazın.

```
cat -s test
```

- test faylında "-s" təkrarlanan bir neçə boş sətri bir sətir kimi sayacaq.

```
cat -n test
```

- test faylında bütün sətirlər və boş sətirləri "n" rəqəmlərlə çap edin.

```
cat -b test
```

- test faylında boş sətirləri çıxmaqla bütün sətirləri "-b" rəqəmlərlə çap edin.

## Grep - Line Processor

İmkanlar:

1. Sətirdə üst-üstə düşən simvolların analizi
2. RegExe-i dəstəkləyir (POSIX və EGREP)
3. İnformasiyanı standart girişdən dəstəkləyir:
  - a. STDIN '<'
  - b. File (bir və ya bir neçə)
  - c. Kanal
4. Axtarış susmaya görə reqistr-ə hissiyyatlıdır.

Adı text fayl yaradıb "**grep**" istifadə edərək axtarış edək. Yaratdığımız faylin içində **unix** sözünü simvolların böyük və kiçikliyinə görə fərqli düzərək bir neçə sətirdə yazın ki, testlərimizin hamısında nəticə əldə etmiş olasınız.

```
grep "Unix" grep.compress
```

- Faylda **Unix** sözünü axtarıraq, reqistr-ə hissiyyatlıdır. Qeyd: Hərflərin böyük və kiçikliyi tam dəqiq yazılmalıdır.

```
grep -i "Unix" grep.compress
```

- Öncəki sətirlə eyni işi burda reqistr fərqi olmadan edirik.

```
grep -vi "Unix" grep.compress
```

- Faylda **Unix** sözü olmayan sətirləri çap edəcək.

```
grep "2007" grep.compress
```

- Faylin hər sətrində '2007' sözünü tapdıqda çap edəcək.

```
grep "[0-9]" grep.compress
```

- 0-dən 9-dək taplığı rəqəmlərdən birini gördükdə çap edəcək (yalnız onlardan 1-i olduğu halda).

**grep -i "Unix[0-9]\$" grep.compress** - registr fərqi olmadan əvvəli "Unix" sözü və sonu 0-dan 9-dək rəqəmlərdən biri olduqda çap edəcək.

**grep -Ri ftp /usr/local/etc /etc | less** - "-R" recursiv olaraq "ftp" sözünü "-i" böyük-kicikliyindən asılı olmayaraq, "/usr/local/etc" və "/etc" qovluqlarında olan fayllarda axtarış edin.

**grep -Rin ftp /etc /usr/local/etc | less** - "-R" rekursiv olaraq, "ftp" sözünü "-i" böyük-kicikliyindən asılı olmayaraq, "-n" tətilan sətirləri rəqəmi ilə göstərin, "/usr/local/etc" və "/etc" qovluqlarında olan fayllarda axtarış edəndə.

**grep -Ril FTP /etc** - "R" rekursiv olaraq, "-i" registr-e hissiyatlı olmadan "-l" fayl adları da daxil olmaqla, "/etc" qovluğunda "FTP" adını axtarış edin.

**grep -v "#" /etc/ttys | less** - "/etc/ttys" faylında qarşısında "#" simvol olmayan sətirləri çap edin.

**egrep -i "Unix[0-9]\*\$" grep.compress** - registr fərqi olmadan əvvəli "Unix", sonra istənilən simvol və sonu 0-dan 9-dək rəqəmlərdən biri olduqda çap edəcək.

RegEx təyinatlarından bir neçəsinin açıqlanması:

- \* - 0 və ya sonsuz simvol
- ? - 0 və ya 1 simvol
- + - 1 və ya bir neçəsinin üst-üstə düşməsi

**Qeyd:** Əgər quantifiers(təyinatlar) istifadə edirikse, **grep**-in əvəzinə '**egrep**' istifadə etmək daha məsləhətdir.

**egrep -i "Unix[0-9]+\$" grep.compress** - registr fərqi olmadan əvvəli Unix sözü olaraq və sonu 1 və ya bir neçə simvol olanı çap edəcək.

**egrep -i "Unix[0-9]+"** grep.compress - registr fərqi olmadan əvvəli Unix sözü olaraq və sonra 1 və ya bir neçə simvol olanı çap edəcək.

```
grep "Sep 19" /var/log/messages > messages.20070919
      - /var/log/messages-faylından görüyü hər
      "Sep 19" sözlü sətri messages.20070919
      faylına yazacaq.
```



# BÖLÜM 4

SHELLin işləmə prinsipi, terminalda qısa keçidlər, CRON, istifadəçi üzərində əməliyyatlar, Vi redaktoru, sistem RAID-ləri, sərt disklerin şifrələnməsi

- / SHELL, onun işləmə prinsipi, terminal qısa keçidləri, CRON
- / İstifadəçilərin yaradılması, silinməsi və deaktiv edilməsi
- / Vi mətn redaktoru və vim
- / Fayl sistəmlə praktik işlər
- / Disklerin bölünməsi və sistem RAID-ləri
- / Sərt disklerin şifrələnməsi

Başlığımızda SHELL-lə işləmə qaydaları, istifadəçiyə hansısa bir shell-in təyin edilməsi və hər shell mühitinə aid olan xüsusiyyətlərin açıqlanmasından danışılır. Hər bir SHELL mühitinə xas olan terminalda asan istifadə etmək üçün qısa əmr ardıcılıqları mövcud olur və bu başlıqda onlar açıqlanır. Müəyyən vaxt aralığında hansısa bir işin görülməsi üçün planlayıcının iş prinsipi barədə məlumat verilir. İstifadəçi və qrupların idarə edilməsi nəzərdən keçirilir. Bütün UNIX/Linux əməliyyat sistemlərində olan vi tekst redaktoru açıqlanır. Diskin bölüşdürülməsi strukturu, fayl sistemin yoxlanılması və program təminatı vasitəsilə raidlərin qurulması barədə danışılır. Həmçinin disklerin fərqli üsullarla şifrələnməsi açıqlanır.

# **SHELL, onun işləmə prinsipi, terminal qısa keçidləri, CRON**

Əməliyyat sistemi üzərində istifadə edilən istənilən əmrlər və əməliyyatlar hansıa **SHELL** mühiti üzərində yerinə yetirilir. Hər bir shell-in susmaya görə özünə məxsus işləmə prinsipi, komfortu və diskomfortu ola bilər. Ancaq siz seçdiyiniz hər bir SHELL-i öz tələbinizə uyğun olaraq dəyişdirə bilərsiniz.

## **cshell faylları**

`/usr/share/skel/` qovluğunda CSHELL mühiti üçün tələb edilən bütün nüsxə faylları mövcud olur. Sistemə hər dəfə yeni istifadəçi əlavə edildikdə, avtomatik olaraq bu istifadəçi üçün şablon fayllar `/usr/share/skel/` ünvanından götürülüb istifadəçinin ev qovluğuna nüsxələnir. Hər bir istifadəçi sistemə giriş etdikdə isə, onun ev qovluğunda yaranmış SHELL mühiti faylları təyinata əsaslanaraq xüsusi ardıcılıqla işə düşür. Aşağıda səliqə ilə həmin SHELL fayllarının təyinatları açıqlanır.

### **dot.login**

- Bu faylda olan əmrlər CSH mühitinə giriş edən kimi işə düşəcək.

### **dot.tcshrc yada dot.cshrc**

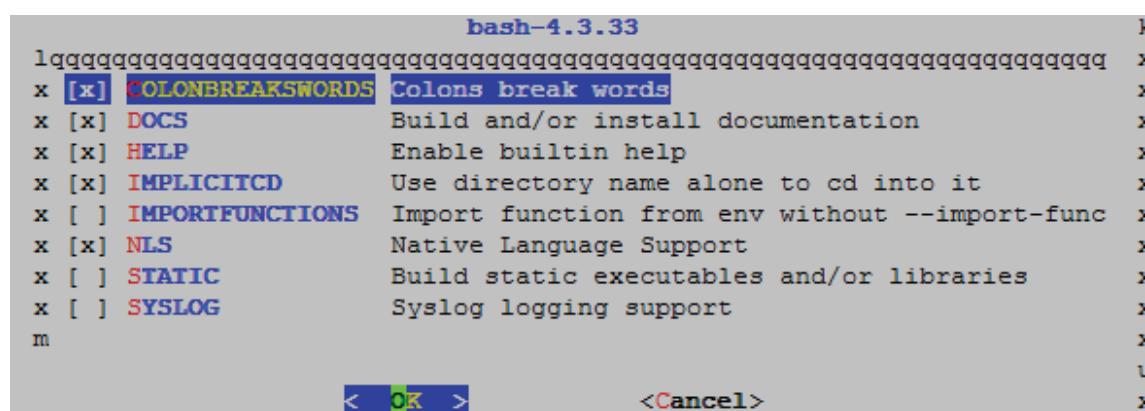
- Bu fayllarda olan əmrlər CSH işə düşən kimi işləyəcək.

<b>dot.history</b>	- CSH mühitində işe salınan bütün əmrlər bu faylda saxlanılacaq.
<b>dot.mailrc</b>	- 'mail' əmri üçün quraşdırımlar bu faylda olur.
<b>dot.mail_aliases</b>	- 'mail' əmri üçün quraşdırımlar bu faylda olur.
<b>dot.rhosts</b>	- Bu fayl '/etc/inetd.conf'-da quraşdırılmış 'rcp' və 'rlogin' servislərinin istifadəçiləri şifrəsiz müəyyən qisim əmrlər istifadə edəndə işə düşür. Burada olan əmrlər həmin əmrlər olur.
<b>echo \$0</b>	- Hansı SHELL mühitini istifadə etdiyimizi çap edir.
<b>echo \$\$</b>	- İstifadə etdiyimiz SHELL mühitinin PID-ni çap edir.

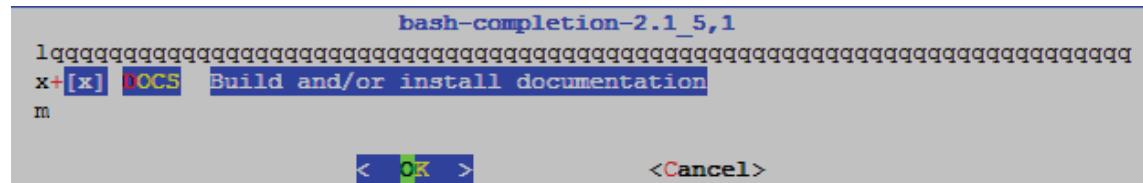
### BASH Shell mühitinin yüklenməsi və quraşdırılması

FreeBSD əməliyyat sisteminin üzərində susmaya görə BASH olmur. Ona görə də, öncə biz onu yükleməliyik.

<b>cd /usr/ports/shells/bash</b>	- BASH Shell-in port ünvanına daxil oluruq.
<b>make config</b>	- Lazımı modulları seçirik.



<code>make install clean</code>	- Yükleyirik.
<code>cd /usr/ports/shells/bash-completion/</code>	- Komfort için emrlərimizin avtotabulyasiyasını da əlavə edək.
<code>make config</code>	- Lazımi modulları seçirik.



<code>make install</code>	- Yükleyirik.
---------------------------	---------------

**Qeyd:** Yüklənmə bitdikdən sonra aşağıdakı sətirlər çap ediləcək. Mütləq yazılanları uyğun olaraq yerinə yetirmək lazımdır.

To enable the bash completion library, add the following to your `.bashrc` file:

```
[[ $PS1 && -f /usr/local/share/bash-completion/bash_completion.sh ]] && \
    source /usr/local/share/bash-completion/bash_completion.sh
```

See `/usr/local/share/doc/bash-completion/README` for more information.

<code>cd /usr/share/skel</code>	- Ünvana daxil oluruq ki, BASH üçün skelet faylları yaradaq.
---------------------------------	--

`ee` əmri susmaya görə FreeBSD əməliyyat sistemində olan mətn redaktorudur. Yəni bu əmri gördükdə təeccübəlməyin. Məsələ burasındadır ki, CSH mühitindən istifadə etdikdə siz ingilis əlifbası ardıcılılığında gələn əmrlərin axtarısını tarixçədə `Up` və `Down` düymələri ilə axtarır tapa bilərsiniz. Ancaq bu BASH-da susmaya görə olmur.

<code>ee dot.inputrc</code>	- Faylı açırıq və içünə aşağıdakı sətirləri əlavə edirik.
<code>"\e[A": history-search-backward</code>	- Bash-History Down düyməsi aktiv olsun.
<code>"\e[B": history-search-forward</code>	- Bash-History Up düyməsi aktiv olsun.

```

ee dot.bashrc

alias ll='ls -lh'
alias la='ls -A'
alias l='ls -CF'
set path= '(/sbin /bin /usr/sbin /usr/bin /usr/local/sbin /usr/local/bin $HOME/bin)'

set      EDITOR  ee
set      PAGER   more
set      BLOCKSIZE      K

[[ $PS1 && -f /usr/local/share/bash-completion/bash_completion.sh ]] && \
source /usr/local/share/bash-completion/bash_completion.sh

```

**Qeyd:** Susmaya görə `.bashrc`-ni istifadəçinin profiline əlavə eləmək üçün '`dot.bash_profile`'-a "`test -f ~/.bashrc && . ~/.bashrc`" sətrini əlavə etmək lazımdır.

#### /usr/share/skel/dot.bash\_profile

- Bu faylda olan əmrlər o zaman işə düşür ki, ya BASH mühitinə keçid olunur, ya da istifadəçinin SHELL-i susmaya görə BASH-dır. Mütləq fayla `test -f ~/.bashrc && . ~/.bashrc` sətrini əlavə edin ki, `.bashrc` işə düşsün.

#### /usr/share/skel/dot.bash\_logout

- Bu fayla əlavə edilən əmrlər istifadəçi `logout` olan kimi işə düşəcək.

#### /usr/share/skel/dot.bash\_history

- Bu fayl hər bir istifadəçinin istifadə etdiyi tarixçələrin saxlanması üçün istifadə olunacaq.

Beləliklə, sistemə əlavə edilən yeni istifadəçi də **SHELL** olaraq **bash** (`/usr/local/bin/bash`) təyin edilsə, bu quraşdırmaları mənimşəyəcək.

Bash shell sessiya avtomatik bağlandıqda `".bash_logout"` faylini işə salır.

Bash işə düşəndə hər istifadəçinin ev qovluğunda olan `".bash_history"` faylini RAM-a yazır. Bu fayl susmaya görə `"$HISTFILE"` dəyişəninə mənimşədilib. **BASH** sessiyasında işləyən

müddətədək history-lər RAM-dan oxunur, seansı tərk edən kimi isə, yeni history-lərlə birlikdə RAM-dan çıxarılıb geriyə ".bash\_history" faylına yazılır.

```
echo $HISTFILE $HISTSIZE $HISTFILESIZE - ".bash_history" faylinin history həcmini və  
history faylinin həcmini çap edin.  
/home/cavid/.bash_history 500 500
```

```
history 5
```

```
!!
```

```
!544
```

- History-dən ən son **5** əmri çap edin.
- Sonra əmri yerinə yetirin.
- **544 ID** altında yerləşən əmri çağırırıq.

Biz əmrlər tarixçəsini "fc" əmri ilə də dəyişə bilərik.

```
fc 544
```

- **544** nömrəli **ID** altında duran tarixçədə dəyişiklik edin. Vi rejimində bir ekran açılacaq. Dəyişikliyi etdikdən sonra yadda saxlayıb çıxan kimi dəyişdiyimiz əmr işə düşəcək.

```
fc -e /usr/bin/ee 544
```

- **544** nömrəli ID altında duran history-ni "**ee**" redaktorla redaktə edin.

```
Ctrl+r
```

```
mail root < /etc/hosts
```

- Tarixçədə sətrə görə axtarış edir.
- "root" istifadəçi adına görə /etc/hosts faylinin məzmununu göndəririk.

## BASH dəyişənləri

```
echo $BASH
```

- '**BASH**' əmrinin tam ünvanını çap edəcək.

```
echo $BASH_COMMAND
```

- Hal-hazırda yerinə yetirilən əmri çap edəcək.

```
echo $BASH_VERSION
```

- BASH-ın versiyasını çap edəcək.

```
echo $COLUMNS
```

- BASH terminalının simvollarla eninin sayını çap edəcək.

```
echo $DISPLAY
```

- X11 server olsa, display-in hansı ID ilə açıldığını çap edəcək.

<code>echo \$EUID</code>	- Hal-hazırkı istifadəçi üçün Effective User ID-ni çap edəcək.
<code>echo \$FCEDIT</code> <code>echo \$GROUPS</code>	- ' <b>fc</b> ' əmri tərəfindən istifadə olunan işləri çap edəcək. - Hal-hazırkı istifadəçinin üzv olduğu əsas qrupu çap edəcək.
<code>echo \$HISTCMD</code>	- Hal-hazırkı əmrin history rəqəmini çap edir.
<code>echo \$HISTFILE</code>	- History faylinin yerləşdiyi yeri çap edir.
<code>echo \$HISTFILESIZE</code>	- History faylinin sətir sayına görə ümumi tutum rəqəmi.
<code>echo \$HOME</code>	- Hal-hazırkı istifadəçinin ev qovluğu.
<code>echo \$HOSTNAME</code>	- Hal-hazırkı maşının adı.
<code>echo \$HOSTTYPE</code>	- Hal-hazırkı maşının HOST tipi.
<code>echo \$LESSOPEN</code>	- LESS açan bir dəyişəndir.
<code>echo \$LINES</code>	- Terminalın hal-hazırkı sətir sayını çap edir.
<code>echo \$LOGNAME</code>	- Hal-hazırda daxil olmuş istifadəçinin adını çap edir.
<code>echo \$MACHTYPE</code>	- İstifadə olunan maşının OS tipini çap edir (i686 ya X64).
<code>echo \$MAIL</code>	- Hal-hazırkı istifadəçinin mailbox qovluğununu çap edir.
<code>echo \$MAILCHECK</code>	- Mail yoxlanışı üçün istifadə olunan vaxt intervalını çap edir (60 saniyə susmaya görə).
<code>echo \$OLDPWD</code>	- İşlədiyimiz qovluqdan öncəki qovluğu çap edir.
<code>echo \$OSTYPE</code>	- Hal-hazırkı OS-un hansı ad altında identifikasiya olmasını çap edir.

<code>echo \$PAGER</code>	- man əmrinin səhifələri page eləməsi üçün istifadə olunan əmr (more əmri ilə).
<code>echo \$PATH</code>	- Sistemdə istifadə olunan sistem əmrlərinin ünvanlarını çap edir.
<code>echo \$PPID</code>	- Hal-hazırda bash tərəfindən işə salınan əmrin Process ID-si.
<code>echo \$PRINTER</code>	- Susmaya görə yüklənmiş olan Printeri çap edir. Hansı ki, ' <b>lpqr</b> ' və ya ' <b>lpq</b> ' əmrlərindən istifadə olunur.
<code>echo \$PROMPT_COMMAND</code>	- İşə salınan əmri adı ilə çap edir və işə salır. Məs: ' <b>PROMPT_COMMAND=ls ; echo \$PROMPT_COMMAND</b> '
<code>echo \$PS1</code>	- <b>SHELL Prompt</b> -un istifadə etdiyi işaretni çap edir. Məs: <b>PS1=`date`</b> CLI-da solda tarix görsənəcək.
<code>echo \$PWD</code>	- Hal-hazırda yerləşdiyimiz qovluğu çap edir.
<code>echo \$RANDOM</code>	- Bu dəyişəni çağırıldığda, 0-la <b>32767</b> arasında olan təsadüfi bir rəqəm generasiya olacaq.
<code>echo \$SECONDS</code>	- <b>SHELL</b> mühit işə salınmağa başlayandan istifadə olunan müddət (saniyələrlə).
<code>echo \$SHELL</code>	- Hal-hazırkı shell-in tam ünvanını çap edir.
<code>echo \$SHELLOPTS</code>	- Aktiv olan <b>SHELL</b> opsiyalarını çap edir.
<code>echo \$SHLVL</code>	- İstifadə etdiyimiz SHELL iç-içə olaraq neçənci SHELL mühitiidir.
<code>echo \$TERM</code>	- İstifadə etdiyimiz terminalın tipini çap edir.
<code>echo \$TMOUT</code>	- Bu dəyişənə mənimsdilən rəqəm mühitinə təyin

edilən vaxtın bitməsidir. Susmaya görə heç nə yoxdur. Ancaq təyin edilən vaxt saniyelərlə olur. Əgər vaxt bitdişə, sessiya atacaq.

`echo $UID`

- Hal-hazırkı istifadəçinin istifadə elədiyi UID-i çap edəcək.

`echo $USER`

- Hal-hazırkı istifadəçinin adı.

Əgər istifadəçimiz öncədən sistemdə BASH yox, digər shell-ə təyin edilmişsə və həmin istifadəçi artıq BASH-la işləmək istəyirsə, onda sizin köməyinizə chsh əmri çatacaq. Aşağıda misallarla göstərilir.

`chsh -s /usr/local/bin/bash`

- Bu əmrlə hal-hazırkı istifadəçinin **\$SHELL**-ni BASH-la dəyişirik.

`chsh -s /usr/local/bin/bash cavid`

- **cavid** adlı istifadəçinin **\$SHELL**-ni BASH-la dəyişirik.

**Qeyd:** BASH mühitində hər hansı yerinə yetirilən iş bitdikdən sonra statusu yoxlamaq istəsək, '`echo $?`' əmrindən istifadə etməliyik. Əgər cavab 0(sıfır)-sa **true**(doğru), eger 1(bir)-se **false**.

**Qeyd:** **\$SHELL**-i dəyişdikdən sonra sistemə yenidən **logon olmaq** (yenidən daxil olmaq) lazımdır ki, yeniliklər işə düşsün.

**Qeyd:** '`~`', simvolu **\$SHELL**-də istifadəçinin **\$HOME**-dir.

**Qeyd:** '`printenv`' – istifadəçinin hal-hazırkı **\$SHELL** dəyişənlərini çap eləyir.

**Qeyd:** '`#`' - superuser işaretəsidir.

**Qeyd:** '`$`' - adı istifadəçi işaretəsidir.

## Terminal qısa keçidləri

Əməliyyat sistemi üzərində klaviatura vasitəsilə olan qısa keçidlər inzibatçının işini çox sürətləndirir və qısa keçidləri bilməsi onun işini çox asanlaşdıracaq. Bu alt başlıqda gündəlikdə lazım olanların əksəri açıqlanır.

### Console

**Fn+PageUp** FreeBSD console-da yuxarı və aşağı axtarış etmək üçün istifadə olunur. İşə düşdükdən sonra **PageUP** və **PageDown** düymələri ilə idarə edə bilərsiniz. Ancaq səhifələrə baxışı bitirdikdən sonra çıxış üçün yenidən **Fn+PageUp** düymələrini sıxmaq lazımdır.

Bütün aşağıda yazılılanlar BASH terminal SHELL-də işləyir. Ancaq hər hal üçün digər shell tipləri üçün də qeyd edilib.

#### Öncə **ctrl** haqqında

**Ctrl + a**

- Sətrin əvvəlinə keçid. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + b**

- Bir simvol əvvələ qayıdır. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + c**

- Proqrama SIGINT siqnalı ötürür. Adətən, mövcud işi dayandırır. ([csh](#), [zsh](#))

**Ctrl + d**

- Kursorun altında duran simvolu silir. (delete düyməsinin analogu) ([cisco](#), [csh](#), [zsh](#))

**Ctrl + e**

- Sətrin sonuna keçid. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + f**

- 1 simvol irəli gedisi. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + k**

- Kursordan sağa doğru sətrin sonundakə silir. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + l**

- Ekranı silir, clear əmrinin analogu. ([csh](#), [zsh](#))

**Ctrl + r**

- Tarixçədə axtarış, axtarış təkrarı. (Yəni axtarışı sətirləyir.) Ya da inkremental axtarış. ([zsh](#))

**Ctrl + j**

- Axtarışı dayandırır və təpilən əmin üstündə dəyişiklik etmək imkanı yaradır. Əgər axtarış yerinə yetirilməyibse, return əminə analoqdur. ([zsh](#)-da əmr yerinə yetirir)

**Ctrl + t**

- Kursorun altında olan simvolu bir simvol öncəki ilə əvəz edir. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + u**

- Kursordan sola doğru sərin əvvəlinədək bütün simvolları silir. ([cisco](#), [csh](#)-də, zsh üçün tam səri silir)

**Ctrl + w**

- Kursordan sola doğru olan sözün əvvəlinədək bütün simvolları silir. ([cisco](#), [csh](#), [zsh](#))

**Ctrl + xx**

- Sətrin əvvelinə və axırına gedib qayıdır. Cisco üçün **ctrl + u**. (**csh**)

**Ctrl + x @**

- Host adına mümkün ola biləcək artırmanı göstərir (adlar /etc/hosts faylından götürülür).

**Ctrl + z**  
**Ctrl + x; Ctrl + e**

- Hal-hazırkı işi müvəqqəti dayandırır. (**csh**, **zsh**)  
- Daxil edilən sətirdə dəyişiklik üçün **\$EDITOR** açır. Dəyişikliklərin yadda saxlanılmasından sonra əmr yerinə yetirilməyə yollanır. Əgər dəyişən təyin edilməyibsə, sistem redaktoru açılır. (FreeBSD üçün emacs-a müraciət edəcək.  
**pkg install emacs24-24.4\_6,3**)

#### İndi isə Alt imkanlarına baxaq

**Alt + <**

- Əmr tarixçəsində ilk əmrə kecid. (**zsh**)

**Alt + >**

- Əmr tarixçəsində ilk əmrə kecid.

**Alt + ?**

- Əmrin bütün mümkün ola biləcək əlavələrini göstərir (tab-tab analoqudur)  
(**csh**, **zsh** üçün **which string**)

**Alt + \***

- Əmrin bütün mümkün ola biləcək əlavələrini CLI-a çap edir.

**Alt + /**

- Faylin adını artırmağa çalışır  
(Tabulyasiyanın analoqudur)

**Alt + .**

- Öncəki əmrin son arqumentini yerləşdirir.  
(!\$ analoqu, ancaq yoxlanış üçün :p etmək lazım deyil)

**Alt + b**

- Kursoru 1 simvol sola çekir (**cisco**, **csh**, **zsh**)  
- Kursor altındakı simvolu böyük edir, qalanlarını isə sözün sonunadək kiçik. (**cisco**, **csh**, **zsh**)

**Alt + c**

- Kursor yerləşən yerdən sağa doru olan sözü sonadək silir. (**cisco**, **csh**, **zsh**)

**Alt + d**

- Kursoru bir söz qabağı aparır (**cisco**, **csh**, **zsh**)  
- Kursorun yerləşdiyi simvoldan sonadək bütün simvolları kiçik edir. (**cisco**, **csh**, **zsh**)

**Alt + f**

- Kursorun altında olan sözlə öncəkinin yerini dəyişir. (**zsh**)

**Alt + l**

- Kursorun yerləşdiyi simvoldan sözün sonunadək bütün simvolları böyür. (**cisco**, **csh**, **zsh**)

**Alt + t**

**Alt + u**

**Alt + back-space**

- Kursor yerleşdiyi ünvandan sözün əvvəlinədək bütün simvolları silir.([cisco](#), [csh](#), [zsh](#))

**TAB imkanlarına baxaq:**

**Tab + Tab**

**(string)T**

**(dir/)T**

**\*T**

**~T**

**\$T**

**@T**

**=T**

- Əmrin artırılması. Əgər boş sətirdə etsək, bütün mümkün ola biləcək əmrlərin siyahısını çap edəcək.  
- Bütün mümkün ola biləcək artırımları çap edəcək.  
- dir adlı qovluğun bütün alt qovluqlarını çap edəcək.  
- - Gizlilərdən başqa bütün alt qovluqları çap edəcək.  
- **/etc/passwd** faylında olan bütün istifadəçiləri çap edəcək. İstifadəçi adını tamamlamaqla onun ev qovluğuna keçmək olar. Misal üçün, **~cavid/** - cavid adlı istifadəçinin ev qovluğu.  
- Sistem dəyişənləri üçün bütün siyahını tamamlayır.  
- **/etc/hosts** faylında olan host adlarını tamamlayır.  
- Mövcud qovluğu siyahı halına salır. **ls** kimidir.

## CRON - System Scheduler

CRON inzibatçının əvəzedilməz alətidir və həmişə inzibatçının işinə yarayacaq. Misal üçün, sistemdə hansısa bir skript yazılı bilər ki, o, ildə, ayda, həftədə, gündə, saatda, dəqiqlidə bir dəfə və ya bir neçə dəfə işə düşməlidir. Məhz bu məqamlarda bizim köməyimizə CRON çatır.

İmkanları:

1. Qlobal **/etc/crontab** faylini və istifadəçi bazalı **/var/cron/tabs** qovluğunda olan fayllarını işə salır.
2. '**crontab**' – istifadəçilərin **cron** cədvəlini idarə etməyə kömək edir.
3. CRON yazılıma strukturunun açıqlanması aşağıdakı kimi olacaq:
  - a. 'm' - dəqiqli - **0-59** - ardıcılılığı ilə bölünür. '**\*/2**' – hər iki dəqiqdən bir işə salır.
  - b. 'h' - saat - **0-23** ardıcılığı ilə bölünür. Məsələn: '**0,2,5,11,13**' - göstərilən saatlarda işə salır.
  - c. 'dom' - Ayın günləri - **1-31**
  - d. 'm' - Aylar - **1-12**
  - e. 'dow' – Həftənin günləri - **0-6 (0 = Sunday)** Məsələn: '**1,3,5**' || '**Mon,Wed,Fri**'
  - f. '**user**' - sistemdə olan hansı istifadəçi adından işə salınacaq.
  - g. '**command**' - **script/command** fayl içinde işə düşəcək əmr və ya əmrin özü.
4. İstifadəçilər üçün CRON təhlükəsizlik kontrolunu aşağıdakı fayllarla edə bilərik:
  - a. '**/etc/cron.allow**' - Yalnız bu faylda olan istifadəçilər cron işə sala bilər.

b. '/etc/cron.deny' - Bu faylda olan istifadəçilərə cron işə salınması qadağandır.

**Qeyd:** Hansı faylların istifadə edilməsini sizin siyaset əsasında təyin edir?

**Qeyd:** Əgər siyasetinizdə bəzilərindən başqa hər kəsə izin varsa, onda istifadə olunur: 'cron.deny'

**Qeyd:** Əgər siyasetinizdə bəzilərindən başqa hər kəsə qadağandırsa, onda istifadə olunur: 'cron.allow'

5. Cron gördüyü iş daxilində 'MAILTO', 'PATH' dəyişənləri dəstəkləyir.

**Qeyd:** Əgər /etc/crontab faylında 'PATH' dəyişəninə işə salacağımız əmr təyin olunmayıbsa, onda işə salacağımız əmin tam yolunu yazmaq lazımdır. Misal üçün, yazacağınız kodlar BASH-da işə düşməlidirsə, CRON işləməyəcək. Çünkü BASH-ın binar faylı /usr/local/bin ünvanında yerləşir və PATH dəyişənidə bu ünvan mövcud deyil.

6. Hər dəqiqə işə salınacaq iş varsa, onlar üçün yoxlanış edir.

7. CRON jurnalları göstərilən ünvanda saxlanılır: '/var/log/cron'

Hər bir istifadəçi üçün crontab fayllar mövcuddur ki, onlarla da cronlar təyin edə bilərsiniz. Bu faylin kontrolu birbaşa superuser tərəfindən və hər bir istifadəçinin öz faylinə özü tərəfindən olur.

İşimiz:

1. '/etc/crontab'-in susmaya görə olan quraşdırılmaları açıqlayaq.

**Qeyd:** 'periodic' sistemdə vaxtaşırı işə salınacaq işi yerinə yetirir, bu qovluqda: '/etc/periodic'

Öz növbəsində '/etc/periodic' işə bütün görəcək işini bu faylda olan quraşdırılmalardan oxuyur: '/etc/defaults/periodic.conf'

Əgər bize lazım olan hansısa işin periodic işləməsini istəsək, '/etc/defaults/periodic.conf' faylini '/etc/periodic.conf' faylinə nüsxələyib özümüzə uyğun olaraq '/etc/periodic.conf' dəyişikliklərimizi etməliyik.

2. Xırda bir CRON yazaq.

a.'/etc/crontab'-a əlavə edək, '\*/2 \* \* \* \* root uptime | awk '{print \$1,\$2,\$3}' >> /home/cavid/`date +%F`.uptime'

**Qeyd:** İşin script-dən oxunmasını isteyirsinizsə, onda əmrləri fayla yazıb, sonra da faylin yolunu crona yazırıq.

```
ee /home/cavid/uptime.sh  
#!/bin/sh  
uptime | awk '{print $1,$2,$3}' >> /home/cavid/`date +%F`.uptime  
#END
```

**chmod 700 /home/cavid/uptime.sh** - Faylı yerinə yetirilən edirik.

b. '/etc/crontab'-a əlavə edirik, '\*/2 \* \* \* \* root /home/cavid/uptime.sh'

3. Scripti istifadəçi adından cronla işə salaq: 'cavid'

a. **cavid** istifadəçi adı ilə daxil olub işə salırıq: 'export EDITOR=ee'

b. **cavid** istifadəçi konsolunda işə salırıq: 'crontab -e'

b1. 'crontab -e -u cavid' - Eynilə 'root' istifadəçi adından 'cavid' istifadəçisinin cronunu təyin edə bilərik.

c. Sətri əlavə edirik: '\*/2 \* \* \* \* /home/cavid/uptime.sh'

Və root istifadəçi adı ilə 'cd /var/cron/tabs/;cat cavid' ünvanında yaranan 'cavid' adında fayla baxırıq.

**crontab -e** - istifadəçilər üçün şəxsi crontab faylıdır.

Aşağıdakı CRON-la deyirik ki, 'cavid' adlı istifadəçiye 1-ci gündən 5-ci gündək, saat 8:15-də 'stats.txt' faylı mail-lə yollanacaq.

**15 8 \* \* Mon,Tue,Wed,Thu,Fri mail cavid < /var/project/stats.txt**

Bu cron yanvar, aprel, iyul, oktyabr aylarının 1-ci günü sistemdə bütün "doc" fayllarını axtarır və 'documents.txt' faylına yazar.

**\* \* 1 1,4,7,10 \* find /doc | grep .doc\$ > /var/sales/documents.txt**

**crontab -eu cavid** - 'cavid' adlı istifadəçi üçün yeni cron əlavə edirik. (Yalnız root edə bilər.) '-u' istifadəçi, '-e' editor rejimə keçid edin.

**crontab -l** - İstənilən istifadəçi öz cron faylinin məzmununa '-l' (list) opsiyası ilə baxa bilər.

```
crontab -l -u cavid
```

- cavid adlı istifadəçinin crontab faylinə root istifadəçi adından baxmaq istəyirik.

```
crontab -r
```

- İstənilən istifadəçi öz crontab faylini bu əmrlə silə bilər. '-r' remove

**Qeyd:** Cron-un başqa bir variantı '**anacron**' var. Anacron da öz növbəsində işləri planlaşdırmaq üçündür. Amma crondan fərqi ondan ibarətdir ki, əgər CRON-da qoyduğumuz planlanmanın işə düşmə vaxtında server sönübsə, onda server yenidən işə düşəndə, o, start olmayıcaq. Ancaq anacron buna baxır və onu işə salır. Anacron susmaya görə sistemdə olmur, onu yükləmək lazımdır. '**pkg install anacron**'

```
pkg install anacron
```

- anacron-u yükleyirik.

```
ee /etc/crontab
```

- anacron-u crontabdan işə düşməsi üçün fayla artırırıq.

```
0      0      *      *      *      root   /usr/local/sbin/anacron
```

Həmçinin **/etc/crontab** faylında periodic əmrləri söndürməliyik. Bunun üçün aşağıdakı sətirlərin qarşısına komment təyin edirik. (Bunu anaCRON tələb edir).

```
#1      3      *      *      *      root    periodic daily
#15     4      *      *      6      root    periodic weekly
#30     5      1      *      *      root    periodic monthly
```

```
/etc/rc.conf
```

- Faylin sonuna aşağıdakı sətri əlavə edirik ki, yenidənyüklənmədən sonra işləsin.

```
anacron_enable="YES"
```

Manualları [anacron\(8\)](#) və [anacrontab\(5\)](#)-də var.

# İstifadəçilərin yaradılması, silinməsi və deaktiv edilməsi

Əməliyyat sistemində istifadəçilərin əlavə edilməsi və silinməsi işlərini mütləq bilmək lazımdır. Bu başlıqda istifadəçilərin necə əlavə edilməsi və silinməsini araşdıracaqıq.

```
adduser
- İstifadəçiləri əlavə etmək üçün əmrdir. Test üçün
bir istifadəçi əlavə edib çıkışına baxaq.

Username: faxri
Full name: Faxri Iskandarov
Uid (Leave empty for default):
Login group [faxri]:
Login group is faxri. Invite faxri into other groups? []:
Login class [default]:
Shell (sh csh tcsh bash rbash nologin) [sh]: bash
Home directory [/home/faxri]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password: şifre
```

```
Enter password again: şifre_təkrar
Lock out the account after creation? [no]:
Username : faxri
Password : *****
Full Name : Faxri Iskandarov
Uid       : 1003
Class     :
Groups    : faxri
Home      : /home/faxri
Home Mode :
Shell     : /usr/local/bin/bash
Locked    : no
OK? (yes/no): yes
adduser: INFO: Successfully added (faxri) to the user database.
Add another user? (yes/no): no
Goodbye!
```

**chpass username**

- İstifadəçi haqqında bütün informasiyanı interaktiv rejimdə dəyişmək olur (şifrə mümkün deyil).

**pw useradd yeni -s /bin/tcsh**

- Yeni istifadəçi əlavə edib və ona **tcsh** mühiti təyin edirik (ancaq istifadəçi şifrəsiz yaranacaq).

**pw usermod yeni -s /usr/sbin/nologin**

- İstifadəçini passiv etmək üçün **nologin** ünvanından istifadə edirik. Lakin unutmayaq ki, mühitin ünvanını tapmaq üçün '**which**' əmrindən istifadə etmək lazımdır.

**pw useradd -n faxri -u 0 -g 0 -o**

- Sistemə yeni **faxri** adlı **root** hüquqlu istifadəçi əlavə edirik.

**pw groupadd faxri**

- **faxri** adlı group-u sistemə əlavə edirik.

**Qeyd:** Sistemdə olan sərt adlandırılmış və dəyişməz qruplar. **nobody** '65534' ID-si ilə işləyir. **nogroup** '65535' ID-si ilə işləyir.

<b>rmuser username</b>	- username adlı istifadəçinin silinməsi əmrini veririk.
<b>rmuser -y username</b>	- İstifadəçini sildikdə bütün suallara yes cavabı verin.
<b>rmuser -v username</b>	- İstifadəçini sildikdə daha da detallı informasiya çap edir.

**Qeyd:** İstifadəçi silindikdə '**rmuser**' bu məlumatları konsola ötürmür və hamısı silinir. İstifadəçi crontabı, növbədə işləyən bütün cronlar, bütün proseslər silinir, tmp-dən istifadəçiye aid olan bütün fayllar silinir, istifadəçinin qrupu silinir.

**pwd\_mkdb /etc/master.passwd** - Əmrələ sistemdə olan şifrə bazasını yeniləyirik.

**Qeyd:** **/etc/login.conf** faylında istifadəçiye aid olan şifrə və digər susmaya görə olan siyasetləri dəyişmək olur. (Məs: **minpasswordlen=8:\**). Dəyişikliyin həmin an işləməsi üçün bu əmri yığırıq. "**cap\_mkdb /etc/login.conf**"

<b>id username</b>	- Əmr istifadəçinin hansı ID-yə mənsub olduğunu çap edir.
<b>finger -l username</b>	- İstifadəçi haqqında tam detallı informasiya çap eləyir.
<b>touch -t 8001031301 messages</b>	- <b>messages</b> jurnal faylinin son dəyişmə tarixini <b>1980-ci il 03.10</b> və saat <b>13:01</b> -ə çevirir.
<b>touch -r maillog messages</b>	- maillog jurnal faylinin tarixini messages faylinə yazır.
<b>uname -a</b>	- OS versiyasını və platformanı göstərir.
<b>uptime</b>	- Serverin dayanmadan işlədiyi müddəti göstərir.
<b>shutdown -p now</b>	- Söndürmək üçün. (-h tarixçəni saxlayıb söndür) -p power
<b>/etc/shells</b>	- Sistemdə istifadə oluna biləcək bütün SHELL mühitləri faylda göstərilir.

# Vi mətn redaktoru və vim

Ümumiyyətlə, FreeBSD əməliyyat sistemində susmaya görə öncədən olan və nisbətən istifadəçiye rahat mətn redaktoru **ee** olur. Ancaq istənilən UNIX/Linux platformalı serverdə Vi mətn redaktoru mövcud olur və sintaksis hər yerdə eynidir. Ona görə də hər bir UNIX/Linux inzibatçının Vi ilə işləmə qabiliyyətinin olması şərtidir.

## Vi redaktoru

Redaktorla fayla daxil olduqda ilk dəfə ESC-siz bütün əmrlər işləyir. Lakin növbəti hər bir yeni əmrə keçid üçün ESC-dən istifadə mütləqdir. Gəlin redaktorun sintaksisini açıqlayaq. Hansısa bir məzmuna sahib olan faylı “**vi filename**” sintaksislə açın və aşağıda göstərilən əmrləri ardıcıl test edin.

<b>a</b>	- Yazı yazmaq üçün əmrdir.
<b>x</b>	- Hər bir simvolu tək-tək silir.
<b>dd</b>	- Bütün sətri silir.
<b>"SHIFT+?" yada /</b>	- Söz axtarışı üçün <b>n - next</b> . Axtardığım növbəti söz
<b>ESC, SHIFT+:, q!</b>	- Yadda saxlamadan çıxış (Ardıcıl əmrlər kombinasiyası).
<b>ESC, SHIFT+:, wq!</b>	- Yadda saxlayaraq çıxış.
<b>ESC, SHIFT+:, w /tmp/newfile.txt</b>	- Çıxanda faylı başqa adla yadda saxlayın. (Başqa sessiya açıb yoxlayın.)
<b>Ctrl+f</b>	- Bir səhifə aşağı düş.
<b>Ctrl+b</b>	- Bir səhifə yuxarı qalx.

<b>Ctrl+d</b>	- Yarım səhifə aşağı düş.
<b>Ctrl+u</b>	- Yarım səhifə yuxarı qalx.
<b>:1</b>	- Faylin 1-ci sətrinə, yəni əvvelinə qayıt.
<b>SHIFT+G</b>	- Faylin son sətrinə düş.
<b>SHIFT+H</b>	- Kursoru olduğumuz səhifənin əvvelinə qaytar.
<b>SHIFT+M</b>	- Kursoru olduğumuz səhifənin ortasına apar.
<b>SHIFT+L</b>	- Kursoru olduğumuz səhifənin sonuna apar.
<b>-</b>	- Kursoru öncəki sətrin əvvelinə apar.
<b>SHIFT+\$</b>	- Sətrin sonu.
<b>SHIFT+^</b>	- Kursoru sətrin əvvelinə apar. Ya da '0' sıfır. (Eyni işi görür.)
<b>SHIFT+(</b>	- Kursoru öncəki abzasın əvvelinə apar.
<b>SHIFT+)</b>	- Kursoru sonrakı abzasın əvvelinə apar.
<b>SHIFT+{</b>	- Kursoru öncəki paraqrafın əvvelinə apar.
<b>SHIFT+}</b>	- Kursoru sonrakı paraqrafın əvvelinə apar.
<b>w</b>	- Kursoru növbəti sözə apar.
<b>SHIFT+W</b>	- Kursoru növbəti sözə apar.
<b>b</b>	- Kursoru öncəki sözə apar.
<b>SHIFT+B</b>	- Kursoru öncəki sözə apar.
<b>e</b>	- Kursoru növbəti sözün sonuna apar.
<b>SHIFT+E</b>	- Kursoru növbəti sözün sonuna apar.
<b>h</b>	- Bir simvol sola.
<b>k</b>	- Bir sətir yuxarı.
<b>j</b>	- Bir sətir aşağı.
<b>l</b>	- Kursoru <b>1</b> hərf sağa çək.
<b>/search_string</b>	- Axtarış edilən sözü ' <b>search_string</b> '-in yerinə yazırıq.
<b>/?</b>	- Öncə axtarış edib tapdığımız sözü çap et.
<b>SHIFT+I</b>	- Teksti yerləşdiyimiz sətrin əvvelindən yazmağa başlayacaq.
<b>SHIFT+A</b>	- Teksti yerləşdiyimiz sətrin sonundan yazmağa başlayacaq.
<b>SHIFT+O</b>	- Yazmaq üçün yerləşdiyimiz sətirdən əvvəldə bir boş sətir aç.
<b>SHIFT+S</b>	- Hal-hazırkı sətri sil və yeni boş sətir aç.

### vi tekstdə dəyişiklik

<b>i</b>	- Hal-hazırkı simvolun öündən yazmağa başla.
<b>SHIFT+I</b>	- Teksti hal-hazırkı sətrin əvvelindən yazmağa başla.
<b>a</b>	- Teksti hal-hazırkı simvoldan sonra yazmağa başla.
<b>SHIFT+A</b>	- Teksti hal-hazırkı sətrin sonundan yazmağa başla.
<b>o</b>	- Bu sərin altından yeni sətir açın və yazmağa başla.

<b>SHIFT+O</b>	- Bu sətrin üstündən yeni sətir aç və yazmağa başla.
<b>s</b>	- Hal-hazırkı simvolu sil və yenisini yaz.
<b>SHIFT+O</b>	- Hal-hazırkı sətri sil və yenisini yaz.
<b>c?</b>	- Replace üçün idi, amma sınaqdan keçmədi.
<b>SHIFT+C</b>	- Kursordan sonra sərin sonunadək sil.
<b>r</b>	- Hal-hazırkı simvolu daxil etdiyim simvolla əvəz et.
<b>SHIFT+R</b>	- Kursordan sonra hər simvolu daxil edilən simvollara əvəz et.

#### vi Delete Paste

<b>x</b>	- Kursorun altında olan simvolu ardıcıl olaraq sağa doğru olanları hər istifadədə sil.
<b>SHIFT+X</b>	- Kursorun altında olan simvolu ardıcıl olaraq sola doğru olanları hər istifadədə sil.
<b>SHIFT+D</b>	- Kursordan sonra olanları sərin sonunadək kəs.
<b>SHIFT+Y</b>	- Hal-hazırkı sətri nüsxələ.
<b>p</b>	- <b>CUT</b> və ya <b>copy</b> olmuş sətri kursordan sonra yerləşdir.
<b>SHIFT+P</b>	- <b>CUT</b> və ya <b>copy</b> olan sətri kursordan öndə yerləşdir.
<b>u</b>	- Bir əmr öncə etdiyimi geri qaytar. <b>UNDO</b> ( <b>VIM</b> -də " <b>CTRL+Z</b> ")
<b>.</b>	- Son istifadə etdiyimiz əmri özünə mənimsət.
<b>J</b>	- Yerləşdiyimiz sərin sonuna növbəti sətri əlavə et.

#### vi rəqəmlərlə

<b>7cw</b>	- Növbəti <b>7</b> sözü sil və yenisini yaz. ( <b>8-ci</b> sözü silməyəcək.)
<b>d5d</b>	- Hal-hazırkı sətir də daxil olmaqla növbəti <b>5</b> sətri sil.
<b>3p</b>	- Öncə kəsdiyimiz mətni kursordan sonra 3 dəfə yerləşdir.
<b>9db</b>	- Kursordan öndə olan 9 sözü kəs.
<b>10j</b>	- Kursoru 10 sətir aşağı endir.
<b>y2)</b>	- Mətni kursordan sonra 2 başlığın sonunadək nüsxələ.
<b>5CTRL+F</b>	- 5 səhifə irəli get.
<b>6J</b>	- Növbəti 6 sətri yan-yana düz.
<b>vi +25 /tmp/services</b>	- 25-ci sətirdən başlayaraq ' <b>/tmp/services</b> ' faylini aç.
<b>vi + /tmp/services</b>	- ' <b>/tmp/services</b> ' faylinin sonuncu sətrindən başlayaraq faylı aç.
<b>vi +/tty /tmp/services</b>	- ' <b>/tmp/services</b> ' faylini ilk " <b>tty</b> " sərinin taparaq aç.
<b>vi -r /tmp/services</b>	- ' <b>/tmp/services</b> ' faylini crash rejimdə bərpa edərək aç.
<b>view /tmp/services</b>	- ' <b>/tmp/services</b> ' faylini yalnız oxuma rejimində aç.
<b>vi protocols rc.conf services</b>	- <b>3</b> faylı da birdən aç.
<b>:n</b>	- Növbəti faylı aç.

<b>:prev</b>	- Öncəki faylı aç.
<b>:wn</b>	- İşlədiyim faylda yadda saxlayıb, ardınca sonrakı faylı aç.
<b>:n!</b>	- Yadda saxlamadan növbəti faylı aç.
<b>:!date</b>	- Sistem əmri date-i vi-dan işə sal.
<b>:!!</b>	- Öncə işə salduğumuz sistem əmrini işə sal.
<b>:20</b>	- Faylin 20-ci sətrinə get.
<b>:5,10w abc.txt</b>	- 5-ci sətirdən 10-dək olan aralığı 'abc.txt' faylinə yaz.
<b>:e abc.txt</b>	- Bu fayldan çıxın və 'abc.txt' faylini redaktə etməyə başla.
<b>::r abc.txt</b>	- 'abc.txt' faylini mövcud fayla yerləşdir.
<b>:s/UNIX/FreeBSD</b>	- Faylda ilk təpilən 'UNIX' sözünü 'FreeBSD' sözü ilə əvəz et.
<b>:s/UNIX/FreeBSD/g</b>	- Faylin kursor yerləşən sətrində bütün 'UNIX' sözlərini 'FreeBSD' sözü ilə əvəz et.
<b>:%s/UNIX/FreeBSD/g</b>	- Bütün faylda olan 'UNIX' sözlərini "FreeBSD" sözü ilə əvəz et.
<b>:g/FreeBSD /p</b>	- Faylda 'FreeBSD' sözü olan bütün sətirləri çap et.
<b>:g/UNIX/s//FreeBSD/gp</b>	- Bütün əvvəlində boşluq olan 'UNIX' sözlərini təpib, boşluğun yerinə 'FreeBSD' sözünü yaz.

#### vi EX commands

<b>:set</b>	- Bütün mümkün ola bilən opsiyaları çap et.
<b>:set</b>	- Susmaya görə olan quraşdırımlardan başqa nələrsə dəyişibsə, onları çap et.
<b>:set number</b>	- Faylda olan sətirləri rəqəmləyin.(Söndürmək üçün ' <b>:set nonu</b> ')
<b>:set list</b>	- Sətrin sonlarına '\$' simvolu və əvvəlinə isə '^I' simvolu əlavə et.
<b>:set wm</b>	- Vi-in səbəbi WrapMargin.

Ekrani bir neçə hissəyə bölüb redaktə etmək üçün isə, vim redaktorundan istifadə etmək rahatdır. Ancaq əməliyyat sisteminin üzərində susmaya görə vim olmur və biz onu yükləməliyik.

<b>pkg install vim</b>	- Bu, həddən artıq uzun vaxt alacaq.
<b>vim rc.conf</b>	- Faylı açırıq.
<b>:split services</b>	- Ekranı horizontal olaraq iki yerə bölcək. <b>BASH</b> varsa, faylin adını yazdıqda tab işləyir.
<b>:vsplit protocols</b>	- Ekranı vertikal olaraq iki hissəyə böllür. (Bunu oxuya biləcək həddədək eləmək olar.)
<b>'Ctrl+WW'</b>	- Bölünmiş ekranlar arasında kecid.
<b>:q</b>	- Yadda saxlamadan çıkış eynidir.
<b>:wq</b>	- Yadda saxlayaraq çıkış eynidir.

# Fayl sistemlə praktik işlər

## FDisk istifadəsi

**Qeyd:** Slice hissə deməkdir.

Tarixən bütün PC-BIOS-lar MBR(Master Boot Record) yazı ilə **32** bitlik platforma üzərində olurdu. Bunun da müəyyən limitləri var idi, bu, bütün disk bölmərinin maksimum həcmi 2TB-a qədər və 4 ədəd Primary Partition-a qədər dəstəkləyirdi. Extended disklərin sayəsində bu həcmi artırmaq olurdu. Ancaq bu həddi "**guid partition table (gpt)**" hesabına aşmaq oldu. Bunun sayəsində diskləri "**9,4 ZB**"-a qədər istifadə eləmək oldu. FreeBSD slice-ları yaradıb və bölmək üçün "**fdisk**" utiliti istifadə edir, "**bslabel**" utiliti isə hər bir hissə həddindən kənar işləmək üçün istifadə olunur. Hər iki aletin də qrafik rejimdən idarəsi mümkündür.

**sysinstall** -> **Configure** -> **Fdisk** -> Və diskimizi seçirik.(FreeBSD8.4)  
**bsdconfig** -> **Disk Management** -> Diskimizi seçirik (FreeBSD9.3 və FreeBSD10.1)

**Qeyd:** Unutmayın ki, "**sysinstall**"-la "**fdisk**"-i istifadə edəndən sonra mütləq "**w**" əmrini daxil edin, əks halda etdiyiniz quraşdırılmalar yadda qalmayacaq.

**Qeyd:** "**U**" əmri etdiyimiz dəyişiklikləri yalnız o halda geri qaytarır ki, "**w**" write-i önce edilməmişdir.

- U** - write olunmayıbsa, son dəyişiklikləri geri qaytar.
  - D** - Seçilən hissəni sil.
  - T** - Partition Slice tipini seç. Məs: **165** (FreeBSD), **6** (DOS FAT16), **7** (NTFS), **130** (Linux swap), **131** (Linux)
  - S** - Seçdiyimiz slice bootable olsun.
  - C** - İstifadə olunmayan yerdən yeni slice yarat.
  - W** - Etdiyimiz dəyişiklikləri yadda saxla.
  - A** - Göstərilən diskı ilk slice-a bütövlükdə mənimsət (məsləhət görülür ki, yalnız ilk yüklemədə bunu istifadə edək).
  - Z** - Seçdiyimiz slice-in həcminə müxtəlif tiplərdə baxmaq olur. (**KB**, **MB**, və **GB**)
  - G** - Diskin Geotermiyasını dəyişmək olar(toxunmaq məsləhət deyil).
  - Q** - Dəyişiklik edilsə belə, "w" opsiyası istifadə olunmayıbsa, bu çıxışda heç nə dəyişmir.
- fdisk -p da0** - Quraşdırma faylı formatında slice cədvəl informasiyasını çap et.
- fdisk -s da0** - **da0** diskini haqqında ümumi şəkildə informasiya çap et.
- fdisk da1** - **da1** diskini haqqında məlumatı tam şəkildə çap et.
- fdisk -BI /dev/da1** - **/dev/da1** adlı diskimizi inisializasiya edirik.
- fdisk -i da2** - CLI-dan **da2** diskini slice-lara ayırmaq olur. (İnteraktiv rejimdə suallara cavab verməliyik. ENTER sıxaraq sonadək davam etsək, nəticədə ilk slice yaradılmış olacaq).

### BSDLabel istifadəsi

**Qeyd:** Öncədən nəzərə alın ki, **bsdlabel**-ı planlaşdırılmış şəkildə '**sysinstall**' vasitəsilə etmək ən düzgün yoldur.

**Qeyd:** BSDLabel diskini yalnız slice ayırdıqdan sonra görür.

**Qeyd:** BSDLabel-ı istifadə etmədən önce label-lərin işləməsi üçün kernel-in geom dəyişənini "**sysctl kern.geom.debugflags=16**" təyin etmək lazımdır.

**BSDLabel** eyni zamanda həm qrafik rejimlə, həm də command line ilə işləyir.

**sysinstall -> Configure -> Label** -> və diskimizi seçirik OK (FreeBSD 8.4)

**bsdconfig -> Disk Management** -> və diskimizi seçib OK düyməsini sıxırıq (FreeBSD9.3, FreeBSD10.1)

- C** - partiton yaratmaq üçün (Həcmi block, **M**-megabayt, **G**-gigabaytlə verə bilərik.)
- U** - write etməmişiksə, etdiyimiz dəyişiklikləri geri qaytar.
- N** - File sistem yarananda əlavə newfs opsiyalarını da təyin et.
- M** - Təyin etdiyimiz file siteminə mount nöqtəsini dəyişmək olar.
- D** - Seçdiyimiz hissəni silir.

**Qeyd:** Bütün etdiyimiz dəyişikliklərdən sonra "w" write əmrini yiğmasaq, bütün edilən dəyişikliklər silinəcək.

- W** - Etdiyimiz dəyişiklikləri yadda saxla.
- Q** - Yadda saxlamadan çıxış.

**bslabel dals1** - BSD slicenin partition informasiyasını çap edir (**Qeyd:** Bu, yalnız slisi-i 'bslabel -w' əmri ilə yazıldıqdan sonra işləyir, yəni orada ən azı 1 label olmalıdır ki, çap edilsin).

**bslabel -n -w dals4** - "-w" Yazılmış label-in nəticələrini çap et, "-n" amma write etmə.

**bslabel -w dals4** - "dals4" standart label-i disk1-ə yazırıq.  
**bslabel -e dals4** - "dals4" label-ni vi editorda açıb FSTYPE\_ni ("4.2BSD" etmək üçün)

**newfs -N /dev/dals4** - Formatlananda nəticəni çap et, amma write etmə.

**newfs /dev/dals4** - UFS File sistemi disk 1, slice 4-ə yazın.

**newfs /dev/da0** - **da0** diskinin ufs ilə formatlanması

**mount /dev/da0 /newdisc** - **da0** diskini newdisc ünvanına mənimşədirik.

Bütün quraşdırımaların sistemin yadında saxlanması üçün, **/etc/fstab** faylinin içində aşağıdakı sintaksislə sətri əlavə edirik:

<b>/dev/da0</b>	<b>/newdisc</b>	<b>ufs</b>	<b>rw</b>	<b>1</b>	<b>1</b>
-----------------	-----------------	------------	-----------	----------	----------

**mount -a** - Sistemi yenidənyüklənmə etmədən **/etc/fstab** faylinin içində yazılın bütün quraşdırımaları işə salır.

**Qeyd:** Susmaya görə FreeBSD sistemi '/' slice-i **read/write** rejimində istifadə edir. Dəyişmək istəsək, **/etc/rc.conf** faylinə **root\_rw\_mount="NO"** sətrini əlavə etmək lazımdır.

## FreeBSD ext2

FreeBSD əməliyyat sisteminə **ext2** fayl sistemi mount edib istifadə etmək mümkündür. Bunun üçün xüsusi program təminatı yüklenməlidir.

```
cd /usr/ports/sysutils/e2fsprogs
```

- **ext2** FS-in dəsteklənməsi üçün bu paketi yükleyirik.

```
make install all
```

```
mkfs.ext2 -v /dev/dals1
```

- **dals1** diskinin 1-ci slice-nda "**ext2**" formatında format et. "**-v**" verbose.

```
mkfs.ext2 -v -c /dev/dals1
```

- "**/dev/dals1**" slice-nda "**-c**" bad blokları yoxlanış et və "**-v**" verbose et.

```
mount -t ext2fs /dev/dals1 /disk
```

- "**/disk**" qovluğuna "**/dev/dals1**" slice-ni "**-t**" (tipi) **ext2fs**-lə mount et.

```
df -HT
```

- Diskləri "**-H**" (insan tərəfindən oxunula bilən formatda) Megabayt və Gigabayt tipli göstər, "**-T**" file sistem tiplərini də göstər.

```
tune2fs -l /dev/dals1 | less
```

- "**/dev/dals1**" disk haqda bütün atributları çap et, "**-l**" diskdə qeyd olunan susmaya görə olan dəyişənləri çap et.

```
dumpe2fs -h /dev/dals1 | less
```

- "**/dev/dals1**" disk haqda bütün atributları çap et, "**-h**" ancaq superblock haqqında informasiyani çap et.

```
tune2fs -c 31 /dev/dals1
```

- "**/dev/dals1**" FS-in sərt yoxlanışı maksimum "**-c**" 31 dəfə olsun.

```
tune2fs -c -1 /dev/dals1
```

- "**-c**" "**-1**" FS-in sərt yoxlanışını, ümumiyyətlə, söndür.

<code>tune2fs -i 10 /dev/dals1</code>	- "dals1" FS-nə yoxlanışı "-i" 10 gündən sonra et.
<code>tune2fs -i 1d /dev/dals1</code>	- "dals1" FS-nə yoxlanışı "-i" 1 gündən sonra et.
<code>tune2fs -i 3w /dev/dals1</code>	- "dals1" FS-nə yoxlanışı "-i" 3 həftədən sonra et.
<code>tune2fs -i 6m /dev/dals1</code>	- "dals1" FS-nə yoxlanışı "-i" 6 aydan sonra et.
<code>tune2fs -i 0 /dev/dals1</code>	- Vaxtdan asılı yoxlanışı söndürürük.

**Qeyd:** FreeBSD xəbərdarlıq edir ki, istismarda **ext2** file sistemi istifadə eləmək məsləhət deyil.

Əgər **ext4** və **ext3** fayl sistemləri FreeBSD əməliyyat sisteminə mount eləmək istəsəniz, onda portlardan `/usr/ports/sysutils/fusefs-ext4fuse` yükleməlisiniz və `/boot/loader.conf` faylinə `fusefs_load="YES"` sətrini əlavə etməlisiniz ki, sistem yenidənyüklənməsində avtomatik modul yüklənsin.

<code>mkfs.ext4 /dev/dal</code>	- <b>dal</b> diskimizə <b>ext4</b> fayl sistem yazırıq.
<code>ext4fuse /dev/dal /mnt/</code>	- Sonra <b>da</b> <code>/mnt</code> qovluğuna mount edirik.

**Qeyd:** **e2fsprogs** 3-cü tərif program yükündikdə, həmçinin fayllara atributların təyin edilməsi və onlara baxış keçirilməsi üçün **chattr** və **lsattr** adlı binary fayllar da yaranır.

#### File atributlarının dəyişdirilməsi

FreeBSD əməliyyat sisteminde ext2 və ext3 file sistemi yaratmaq üçün xüsusi 3-cü tərif "**e2fsprogs**" program yükündikdə, eynilə fayllar üzərində atributların təyinatı funksionallığı da yaranır. Bu alt başlıq həmin atributların imkanlarını açıqlayır.

<code>lsattr -aR /tmp/   less</code>	- Rekursiv olaraq linux atributlarını çap edir.
--------------------------------------	---

Atributlar:

- a** - Ancaq əlavə eləmək;
- c** - Sıxılmış;
- d** - Dump olmasına;
- i** - Dəyişməz, bu opsiya ilə təyin olunan faylı silmək, ad dəyişmək və ya link etmək olmaz;
- j** - Verilənlərin jurnallanması;

- s** - Təhlükəsiz silinmə;
- t** - Aşağı qarışqlıq olmasın;
- u** - Silinməzdır;
- A** - Yenilənmə vaxtı yoxdur;
- D** - Qovluqla sinxron şəkildə yenilənmə;
- S** - Sinxron yenilənmə;
- T** - Direktiv iyerarxiyanın başı;

Atribut təyin etmək üçün "+", silmək üçün isə "-" simvolundan istifadə edilir.

Bu atributları biz "**chattr**" əmri ilə dəyişə bilərik.

Məs: **chattr +i test**

Məs: **chattr -t test**

## Fayl sistemin yoxlanılması

İşlədiyiniz mühitdən asılı olaraq, elə ola bilər ki, server otağının mərkəzi UPS sistemi və ya serverin ayrıca UPS-i mövcud deyil. UPS olmayan hallarda əgər diskin üzərində iş gedən müddədə işıqlar keçərsə, fayl sistem zədələnəcək və yoxlanışdan keçmədən sistem işləməyəcək. Bunun üçün fayl sistemi yoxlamaq və bərpa imkanına malik olan faylları bərpa etmək və ya tam zədələnənləri silmək lazımdır. Aşağıda bütün üsullar açıq şəkildə nümayiş etdirilir:

```
fsck_ufs /dev/da0s1e
fsck_ufs -B /dev/da0s1f
fsck_ufs -B -f da0s1e
```

- "/**dev/da0s1e**" alətini ufs yoxlanış edir.
- '**-B**' arxa fonda file sistemi yoxlanış et.
- "**fsck\_ufs**" özü avtomatik '**da0s1e**' ünvanını tapıb '**-B**' arxa fonda '**-f**' force rejimdə yoxlanış edəcək, hətta təmiz olsa belə.
- "**-d**" debugging rejimdən başqa olan bütün əmrləri çap et.
- Əgər hansısa faylı düzəldə bilirsə, bütün suallara "**yes**" cavabı verin.
- "**msdosfs**" file sistemi yoxlanış et.
- Diskləri '**-h**' insan tərəfindən oxunula bilən və '**-i**' istifadə olunan inode-ların siyahısını çap et.
- '**-h**' insan tərəfindən oxunula bilən və '**-l**' daxili file sistemimizi çap edin. Yəni əgər biz nfs və ya SMBFS-lə share mount eləmişiksem, onlar çap edilməyəcək.

```
fsck_ufs -d /dev/da0s1a
```

```
fsck_ufs -y /dev/das01a
```

```
fsck_msdosfs /dev/dals2
df -hi
```

```
df -hl
```

<code>df -ht ufs</code>	- "-h" human readable və "-t" tipi 'ufs' olan file sistemi çap et.
<code>df -h /home/</code>	- "-h" human readable-da yalnız "home" slice-ni çap et.
<code>du -h /home/</code>	- "-h" human readable-da "/home" silce-nin istifadə etdiyi disk tutumunu açıqlayaraq, çap et.
<code>du -sh /home/file</code>	- "-h" human readable-da və "-s" seçilmiş fayl üçün "/home/file" faylinin həcmini çap et.
<code>du -sch /home /var</code>	- '-h' human readable-da "/home" və "/var" slice-nin ayrı həcmi və "-c" ikisi birlikdə olan ümumi həcm.
<code>find /var -maxdepth 1 -type d -exec du -sh {} \;</code>	- Tap "/var" qovluğunda "-maxdepth" dərinliyində 1-ci səviyyə qovluqların hamısında, əgər "-type" tipi "d" qovluqdursa, nəticələrin həcmini çap edin. Yəni "/var" qovluğunda olan yalnız bütün 1-ci səviyyə qovluqların həcmini çap edəcək.

# Disklərin bölünməsi və sistem RAID-ləri

## Disklərin bölünməsi

FreeBSD ilk yüklenməsində sistemin işlədiyi diskin driver tipini təyin etmək üçün kernelə müraciət edir. Kernel isə öz növbəsində kompilyasiya edilmiş driver siyahısından uyğun adı seçib diskə qaytarır. Disklər sistemimizdə /dev ünvanında virtual alət olaraq yerləşir və aşağıdakı struktur ilə açıqlanır:

- da** - SCSI|SATA\USB Mass storage alətləri
- ad** - IDE mass storage
- fla** - Flash drives
- cd** - SCSI|SATA cd-roms
- acd** - IDE cd-roms
- fd** - floppy

Əməliyyat sistemində istifadə etdiyimiz disklərin yoxlanılması və test edilməsi üçün gözəl utilit mövcuddur, hansı ki, sayəsində diskin I/O(Input/Output)-nu yoxlamaq mümkündür. Bu dd-dir. Başlığımızda önce test üçün yeni diskı sistemə əlavə edəcəyik və onu format edib sistemimizə mount edəcəyik.

Bir disk bir neçə slice-a bölək

**dd if=/dev/zero of=/dev/da4 bs=1k count=1** - Diski tamam boşaldırıq.

**fdisk -BI da4**

- "-B" boot codu sector 0-a, "-I" inisializasiya edin.

**bslabel -B -w da4s1 auto**

- "-B" boot code "**/boot/boot**" ünvanından oxunub **da1s1** diskinə yazılıcaq. Və disk düzülüşünü "**auto**"-dan alın, yəni "disktab"-dan, hansı ki, simvolları "**/etc/disktab**" faylından alın.

**bslabel -e da4s1**

- da4s1 adlı diskdə FSTYPE-i 4.2BSD təyin edirik.  
Nəticə aşağıdakı kimi olacaq.

```
# /dev/dals1:  
8 partitions:  
#      size     offset   fstype   [fsizze bsize bps/cpg]  
a: 41929571          16   4.2BSD       0     0     0  
c: 41929587          0     unused       0     0  # "raw" part, don't edit
```

**mkdir -p /1**

- Kök ünvanda **/1** adlı qovluq yaradırıq.

**newfs /dev/da4s1a**

- Ardıcılılığı bütün slice-lar üçün edirik.

**mount /dev/da4s1a /1**

- Slice-ı mount edirik.

Sonda **/etc/fstab** faylına aşağıdakı sətri əlavə edirik ki, sistemin yenidən yüklenməsində işləsin.

<b>/dev/da2s1a</b>	<b>/1</b>	<b>ufs</b>	<b>rw</b>	<b>0</b>	<b>0</b>
--------------------	-----------	------------	-----------	----------	----------

# Sistem RAID-ləri

Bu alt başlığımızda əməliyyat sisteminin imkanları ilə program səviyyəsində RAID qurub test edəcəyik. Başlamazdan önce bir neçə nəzəri hissələri açıqlamaq istərdim. RAID müxtəlif növlərə malikdir və onlardan ən gündəmdə olanları **RAID0**, **RAID1** və **RAID5**-dir. RAID-lər haqqında <https://ru.wikipedia.org/wiki/RAID> linkindən daha da ətraflı oxuya bilərsiniz.

## GEOM\_STRIPE quraşdırılması( RAID0 )

```
klldload geom_stripe
```

- **geom\_stripe** modulunu sistemdən çağırırıq.  
Susmaya bu modul yüklenmiş olmur.  
Ya kernel-dən çağrırmalı, ya da KERNEL-in  
daxilində kompilyasiya etməlisiniz.

```
mkdir /raid0
```

- Mount üçün **/raid0** adlı qovluq yaradırıq.

```
gstripe label -v st0 /dev/dal /dev/da2
```

- **gstripe** əmri **st0** adlı disk çağrıır və bu diskə  
2 ədəd da1 və da2 diskinin həcmini tikiir.

```
bsdlabel -wB /dev/stripe/st0
```

- **bsdlabel** əmri diskdə ilk label yaratdı (**st0a**).  
Ancaq istəsək, label yaratmadan da birbaşa  
diskimizi format edə bilərik.

```
newfs /dev/stripe/st0
```

- **st0**-i format edirik. Sonra istədiyimiz ünvana  
mount edirik.

```

mount /dev/stripe/st0 /raid0           - mount edirik verdiyimiz ünvana

echo 'geom_stripe_load="YES"' >> /boot/loader.conf
                                         - modulu startupa əlavə edirik, ya da kernel-imizi
                                         "options GEOM_STRIPE" ilə kompilyasiya edirik.

ee /etc/fstab
Ctrl+u
Ctrl+e
/dev/stripe/st0      /raid0 ufs    rw   2     2
                                         - Virtual diskı startupa əlavə edirik.

                                         - Enter sıxaraq yeni sətrə keçid alıb aşağıdakı
                                         sətri əlavə edirik və yadda saxlayaraq çıxırıq.

```

**Qeyd:** Eynilə logic "st.." disklerini belə mirror(güzgü)etmək olar, ancaq mount olunmadan önce.

RAID1 quraşdırılması

**sysctl kern.geom.debugflags=17** - debugging rejimi aktiv edirik.

Həmçinin kernel dəyişənlərinin sistemin yenidən yüklənməsindən sonra avtomatik olaraq işə düşməsini istəyirikse, **/etc/sysctl.conf** faylinın sonuna aşağıdakı sətri əlavə edirik.

**kern.geom.debugflags=17**

**kldload geom\_mirror** - Disklər üçün güzgü işini görən modulu çağırırıq.

**gmirror load** - Öncəki əmrələ eyni işi görür. Mirror modulunu yükleyirik (kldload-la eynidir).

**/boot/loader.conf** - Modulun sistem yenidənyüklənməsindən sonra işləməsi üçün fayla **geom\_mirror\_load="YES"** sətri əlavə edirik. Ya da kernel-imizi "options GEOM\_MIRROR" opsiyası ilə kompilyasiya edirik.

**gmirror label -vnb round-robin gm0 /dev/da3** - Güzgü modulu **round-robin** alqoritmi ilə **gm0** adlı virtual güzgü diskinə 1-ci disk da3-u tikir.

<code>gmirror insert gm0 /dev/da4</code>	- <code>gmirror gm0</code> diskinə güzgülənmə üçün 2-ci diski əlavə edir.
<code>gmirror configure -a gm0</code>	- ' <code>gm0</code> ' alətinin bütün disklerimiz üçün güzgü rolu oynayacağını elan edirik.
<code>gmirror status</code>	- Güzgülənmiş diskləri çap edəcək.
<code>newfs /dev/mirror/gm0</code>	- <code>gm0</code> güzgü diskini formatlarıyıq.
<b>Qeyd:</b> Mirror-a əlavə etdiyimiz disklərin həcmi eyni olmalıdır.	
<code>mkdir /guzgu</code>	- <code>/guzgu</code> adlı qovluq yaradırıq mount point.
<code>mount /dev/mirror/gm0 /guzgu</code>	- <code>gm0</code> güzgü diskini ' <code>/guzgu</code> ' ünvanına bağlayırıq.
<b>Qeyd:</b> Əgər sistem yenidən yüklənmədən sonra qalxmasa və bu simvol çıxsa, -> " <code>mountroot&gt;</code> " düymə ilə reboot olub "6"-ci seçimi edirik. Və aşağıdakı əmrləri daxil edirik.	
<code>load geom_mirror</code>	- Modulu yükleyir.
<code>boot</code>	- Sistemi yükleyir.
Əgər diskimizin biri sıradan çıxsa, əvvəl adını dəqiqləşdiririk ki, hansıdır. Sonra çıxardırıq.	
<code>gmirror forget gm0</code>	- Mirror-u passiv edib disk çoxarıraq.
<code>gmirror insert gm0 /dev/dal</code>	- Mirror-a yeni disk əlavə edirik.
<code>gmirror list</code>	- Tərkibində olan diskləri və həcmərini göstərir.
<code>gstat</code>	- Disklərin saniyələrlə giriş-çixışını həcmə göstərmək üçün utilitidir.
<code>cd /guzgu</code>	- Test üçün ünvana girib, <code>gstat</code> əmrini başqa seansda daxil edərək buraya informasiya yazıb disklərin <code>input/output</code> -na baxaq.
<code>jot 10000000 &gt; file</code>	- Müəyyən bir informasiya yazdıqda eyni vaxtda ' <code>gstat</code> ' əmrinin çıxışına digər seansda baxın.

```
cat '/dev/zero' > /guzgu/file
```

- Həmçinin bu əmr ilə də fayla yazıl test edə bilərsiniz.

Sonda isə '/guzgu' diskini umount edib, disklər haqqında informasiyani '/etc/fstab'-da və '/boot/loader.conf'-da komentariya edirik. Sonra da reboot edirik.

Artıq həmin diskləri sistemdə ayrı-ayrı qovluqlara mount edirik və eyni informasiyanın hər ikisində olduğunu görürük.

### Raid10 qurulması

Gəlin indi də virtual yaratdığımız iki ədəd - **st0** və **st1** diskimizi güzgü rejimində işə salaq.

```
kldload geom_stripe  
kldload geom_mirror
```

- **geom\_stripe** modulunu çağırırıq  
- Disklər üçün güzgü işini görən modulu çağırırıq.

```
gstripe label -v ss0 /dev/dal /dev/da2
```

- gstripe **ss0** adlı disk çağrıır və bu diskə **2** ədəd **dal** və **da2** diskinin həcmi tikiir.

```
gstripe label -v ss1 /dev/da3 /dev/da4
```

- gstripe **ss1** adlı disk çağrıır və bu diskə **2** ədəd **da3** və **da4** diskinin həcmi tikiir.

```
gmirror label -vnb round-robin gm0 /dev/stripe/ss0
```

- Güzgü modulu **round-robin** alqoritmi ilə **gm0** güzgü diskinə **ss0** logical diskı əlavə edir.

```
gmirror insert gm0 /dev/stripe/ss1
```

- Güzgü modulu **round-robin** alqoritmi ilə **gm0** güzgü diskinə **ss1** logical diskı əlavə edir.

**-b** - Balans.

**-n** - Komponentlərin avtomatik sinxronizasiyasını dayandırır.

**-v** - Görülən işi consola yollayın.

```
gmirror configure -a gm0
```

- **gm0** diskı bütün disklərimiz üçün güzgü rolunu oynayacaq.

```
newfs /dev/mirror/gm0
```

- **gm0** güzgü diskini formatlarıyıq.

<code>mkdir /raid10</code>	- Mount edəcəyimiz qovluğu yaradırıq.
<code>mount /dev/mirror/gm0 /raid10</code>	- gm0-ı raid10 ünvanına mount edirik.
<code>/etc/fstab</code>	- Disklərimizin startup faylına aşağıdakı sətri əlavə edirik ki, güzgü diskimiz sistemin yenidən yüklənməsində avtomatik işə düşsün.
<code>/dev/mirror/gm0</code>	<code>/raid10</code>
	<code>ufs rw 2 2</code>

### CCDCONFIG vasitəsilə RAID konfiqurasiyası

`ccdconfig` - birləşdirilmiş disklər driver üçün konfiqurasiya utilitiidir.

CCD vasitəsilə RAID0 yaradaq.

`ccdconfig ccd0 32 0 /dev/da1 /dev/da2 /dev/da3 /dev/da4`

- RAID util, `/dev/ccd0` virtual alet altına **4** ədəd disk tiki və onları `raid0` edir.

`ccdconfig -g > /etc/ccd.conf`

- '`rc`' bu configi oxuyur və CCD-ləri inisializasiya edir.

`mkdir /raid0`

- **RAID0** üçün qovluqları yaradırıq.

`newfs /dev/ccd0`

- Yeni yaratdığımız virtual diskimizə UFS fayl sistem yazırıq.

`/etc/fstab`

- Disk StartUP fayla aşağıdakı sətri əlavə edirik ki, sistemin yenidənyüklənməsindən sonra avtomatik işə düşsün.

`/dev/ccd0 /raid0 ufs rw 2 2`

`/boot/loader.conf`

- `geom_ccd` modulu kernel-də öncədən kompilyasiya edilmədiyinə görə, aşağıdakı sətri StartUP modul faylına əlavə edirik ki, sistem yenidənyüklənməsin-dən sonra avtomatik yüklensin.

`geom_ccd_load="YES"`

`mount -a`

- Diskimizi `/etc/fstab` faylından oxuyaraq mount edirik.

`ccdconfig` vasitəsilə RAID1 yaradaq.

`mkdir /raid1`

- raid1-i mount etməyimiz üçün qovluq yaradırıq.

`ccdconfig -U`

- Əgər lazımdırsa, yenidən yüklənmə edin və Single-User rejimə keçin.

`ccdconfig ccd0 32 CCDF_MIRROR /dev/da[5-8]` - 4 disk arasında güzgü edirik.

`ccdconfig -g >> /etc/ccd.conf`

- Həmçinin **RAID1** haqqında olan quraşdırmaları '**rc**' burdan oxuyur. Əgər aşağıdakı sətir faylda olmazsa, sistemin yenidənyüklənməsində diskler mount olmayıcaq və tək istifadəçili rejimə kecid alacaqsınız.

## GVINUM - Logical Volume Manager control program

Adından göründüyü kimi, logik disklerin idarə edilməsi üçün program təminatıdır. Dəstəklədiyi RAID siyahısı aşağıda göstərilir:

**RAID0** - Disk bölüşdürülməsi

**RAID1** - Güzgülənmə

**RAID5** - Cütlükə növbələşmə

**RAID10** - Bölüşdurmə və Güzgülənmə

`/dev/gvinum`

- Obyektlər bu ünvanda saxlanılır. Disk adları və iyerarxiyaları.

Verilənlər bazasının quraşdırma faylları hər bir 'vinum' diskdə saxlanılır.

`newfs /dev/gvinum/VOL_NAME`

- Bu strukturu istifadə edərək, diskimizə File System yazırıq.

**Qeyd:** GVINUM modulu susmaya görə kerneldə yüklenmiş olmur. Yəni əgər biz '**gvinum**' diskini '**/etc/fstab**' faylına əlavə etsək, o, sistemin yenidənyüklənməsindən sonra işləməyəcək. İşləməsi üçün aşağıdakılari edirik:

`/boot/loader.conf`

- Sistemin Modul StartUP faylına **GEOM\_VINUM** modulunu əlavə edirik.

`geom_vinum_load="YES"`

- gVinum modulu yükənsin.

### Birləşmiş həcm yaradaq (RAID0)

**gvinum create** - Müəyyən bir fayl yaradır və ora lazımi diskləri əlavə edib, yadda saxlayır.  
(Aşağıdakı sətirləri ora əlavə edirik):

```
drive a1 device /dev/dal
drive a2 device /dev/da2
volume volconcat01
plex org concat
sd length 7168m drive a1
sd length 7168m drive a2
```

**/dev/gvinum/volconcat01**

- Artıq bu adda alət var və ona 'newfs'-lə fayl system yazmaq olar.

**gvinum printconfig**

- gvinum-un quraşdırmasını çap edə bilir.

**newfs /dev/gvinum/volconcat01**

- GVINUM diskı UFS file system-ə format edirik.

**mkdir /volconcat**

- **volconcat** adlı qovluq yaradırıq ki, GVINUM formatlanmış aləti ona mount edək.

**mount /dev/gvinum/volconcat01 /volconcat**

- **volconcat01** alətini **/volconcat** ünvanına mount edirik.

**/etc/fstab**

- Disk StartUP fayla əlavə edirik ki, yenidən yüklənmədə işləsin.

**/dev/gvinum/volconcat01 /volconcat**

ufs        rw                  2                  2

**gvinum lv**

- Əmrlə diskin statusuna baxırıq.

**gvinum rm -r volconcat01**

- Əmrlə yaradılmış disk volume silinir.

**Qeyd:** **rm** əmri disk mount olanda işləməyəcək, mütləq disk 'umount' eləmək lazımdır.

**Qeyd:** gvinum quraşdırmasını sıfırlamaq üçün '**gvinum resetconfig -f**' əmrini daxil etmək lazımdır.

## GVINUM RAID1

```
gvinum mirror /dev/da1 /dev/da2
```

- **da1** və **da2** disklərini güzgü rejimində işləməsi üçün aktivləşdiririk.

```
newfs /dev/gvinum/gvinumvolume0
```

- Avtomatik ad verilmiş **gvinumvolume0** adlı virtual diskimizə fayl system yazırıq.

```
mount /dev/gvinum/gvinumvolume0 /mnt
```

- Virtual diskimizi **/mnt** ünvanına mount edirik.

## GVINUM RAID10

Birləşmiş disklər arasında RAID1 edirik.

```
gvinum mirror -s -n data /dev/da3 /dev/da4 /dev/da5 /dev/da6
```

-**s** stripesize

-**n** virtual diskimizə **data** adı veririk.

```
newfs /dev/gvinum/data
```

- **data** adlı virtual diskimizə fayl system yazırıq.

## GVINUM RAID5

Məqsədimiz **da1**, **da2** və **da3** diskləri arasında **raid5** yaratmaqdır. Birləşdirici sərhədin həcmi -**s 493k** olmaqla -**n raid5** adlı virtual disk yaradırıq.

```
gvinum raid5 -n raid5 -s 493k /dev/da7 /dev/da8 /dev/da9
```

**mkdir /raid5** - Virtual diskı mount etmək üçün qovluq yaradırıq.

```
mount /dev/gvinum/raid5 /raid5/
```

- **raid5** adlı virtual diskimizi yaratmışımız **/raid5** ünvanına mount edirik.

Əgər sizin disklərin hansısa sıradan çıxarsa, aşağıdakı səhvi görəcəksiniz:

```
sd name myraid5vol.p0.s2 drive gvinumdrive2 len 32538s driveoffset 265s
```

# Sərt disklərimizin şifrələnməsi

FreeBSD sərt disklərin şifrələnməsini dəstəkləyir. Şifrələnmə üçün iki üsuldan istifadə edə bilərik. Siz bu işi GBDE və ya GELİ ilə edə bilərsiniz. Bu imkanların hər ikisi də faylları ayrılıqda şifrələmir, bütövlükdə diski şifrələyir.

## GBDE – GEOM Based Disk Encryption (Geom bazalı disk şifrələnməsi)

**GBDE** vasitəsilə şifrələnmə işini görmək üçün onu ya modul vasitəsilə çağırmaq, ya da kernelin içində əlavə etmək lazımdır.

Kernelin içində istifadə ələmək istəsəniz, aşağıdakı sətri kernel faylinə əlavə edib kompilyasiya etməlisiniz:

**options GEOM\_BDE**

Modulla çağırmaq istəsəniz, aşağıdakı əmri işə salmalısınız:

**kldload geom\_bde**

Həmcinin StartUP-da işleməsi üçün **/boot/loader.conf** faylinə aşağıdakı sətri əlavə etmək lazımdır:

**geom\_bde\_load="YES"**

İndi isə biz **da1** adlı yeni diskimizi bütövlükdə GBDE vasitəsilə şifrələyib **/gbdecrypt** qovluğuna mount edəcəyik. Siz həmcinin **/home** və **/var/mail** qovluqlarını da şifrələyə bilərsiniz, ancaq bunun ardıcılıq proseduru daha çətindir və bu başlığımızda açıqlanmır.

```
mkdir /etc/gbde
```

- Bloklanacaq GBDE faylları üçün qovluq yaradırıq. Bloklanmış faylda GBDE şifrələnmiş diskin hissəsinə yetki almaq üçün informasiya saxlayır. Bu fayla yetki olmazsa, diskdə olan şifrələnmiş verilənləri əllə müraciət edilmədən GBDE deşifrə edə bilməyəcək. Hər bir şifrələnmiş disk hissəsi ayrı blok fayldan istifadə edir.

```
gbde init /dev/dal -i -L /etc/gbde/dal.lock
```

- gbde diskə işləməyə başlamazdan önce onu inisializasiya etmək lazımdır. Inisializasiya yalnız bir dəfə olur. Əmrlə redaktor açılacaq, hansı ki, şablonda fərqli konfiqurasiya parametrlərini təyin edə bilərsiniz. **UFS1** və **UFS2** fayl sistemləri ilə işlədikdə **sector\_size** parametrini həmişə **2048** təyin edin. Vİ redaktordan çıxıldıqda sizdən şifrə soruşulacaq. Bu şifrə verilənlərin qorunması üçün istifadə ediləcək. Bu şifrə çox çətin təyin edilməlidir.

**gbde init** əmrini yığmaqla diskiniz üçün bloklama faylı yaranır, hansı ki, bizim halda **/etc/gbde/dal.lock** adında oldu. Bloklama fayllar **/etc/rc.d/gbde start** skripti tərəfindən düzgün tanınması üçün faylların sonu mütləq .lock genişlənməsi ilə bitməlidir.

**Qeyd:** Bloklama faylların nüsxəsi şifrələnmiş diskin özündə yerləşməlidir, çünkü bu faylin nüsxəsi heç kəsin əlinə keçməli deyil. Fayl itərsə, informasiya bərpası da çox çətin olacaq. Hətta rəsmi yazarlar özləri belə bunu etmirlər.

```
gbde attach /dev/dal -l /etc/gbde/dal.lock
```

- Şifrələnmiş disk sistemə qoşaq. Bir az önce yaratdığımız şifrəni daxil edirik. **ls /dev/dal\*** əmrini daxil etsəniz, **/dev/device\_name.bde** quruluşlu yeni disk yaranacaq.

```
mkdir /gbdecrypt
```

- Şifrələnmiş diskimizi mount edəcəyimiz qovluğu yaradırıq.

```
newfs -U -O2 /dev/dal.bde
```

- Diskimizə fayl sistem yazırıq. **Qeyd:** Fayl sistem mütləq **.bde** genişlənməli diskə yazılımalıdır.

<code>mount /dev/dal.bde /gbdecrypt/</code>	- Diskimizi mount edirik.
<code>df -H   grep bde</code>	- Şifrlənmiş diskimizə baxaq
<code>/dev/dal.bde 5.1G 8.2k 4.7G 0% /gbdecrypt</code>	<code>0% /gbdecrypt</code>
<code>fsck -p -t ffs /dev/dal.bde</code>	- Diskimiz <code>/etc/fstab</code> faylında avtomatik işə düşməsi üçün göstərilə bilməz və buna görə də əlimizlə yoxlanış edirik.
 Bundan əlavə, diskin sistem StartUP-da işə düşməsi üçün aşağıdakı sətirləri <code>/etc/rc.conf</code> faylına əlavə etmək lazımdır. Ancaq yenə də sistem qalxdıqda şifrəni əllə konsoldan daxil etməlisiniz.	
<code>gbde_autoattach_all="YES"</code>	
<code>gbde_devices="dal"</code>	
<code>gbde_lockdir="/etc/gbde"</code>	
<code>gbde detach /dev/dal</code>	- Qorunan diskin ayrılması üçün əmri daxil etməniz kifayətdir.

GBDE sektorlarının tərkibini 128 bitlik AES ilə şifrləyir. Hər bir sektor fərqli AES açarla şifrlənir.

#### GELİ vasitəsilə disklerin şifrlənməsi

GELİ-də olan üstün xüsusiyyətlər aşağıdakılardır:

- **CRYPTO(9)** infrastrukturunu istifadə edir və avadanlıq səviyyəsində şifrlənmə görən kimi onu istifadə edir.
- Fərqli şifrlənmə alqoritmləri dəstəkləyir. Hal-hazırda **AES**, **Blowfish** və **3DES**.
- Kök diskin şifrlənməsini dəstəkləyir, ancaq sistem qalxanda şifrəni daxil etmək lazımdır.
- İki bir-birindən asılı olmayan şifrlənmə açarını dəstəkləyir(misal üçün, əsas açar və şirkətin açarı).
- Sektor-Sektor şifrlənməsinə görə sürətli işləyir.
- Əsas açarların arxivləşməsi imkanı. Tələb yaranarsa, bu açar silinə bilər və gələcəkdə verilənlərə yetkini arxiv açarlarından bərpa edə bilərsiniz.
- Birdəfəlik açarlarla fayl sistemin şifrlənməsi imkanı.

İşə salmaq üçün kerneli aşağıdakı sətirlərlə kompilyasiya etmək lazımdır:

```
options GEOM_ELI
device crypto
```

Ya da **/boot/loader.conf** faylinə aşağıdakı sətri əlavə etmək lazımdır:  
**geom\_eli\_load="YES"**

**kldload geom\_eli**

- Həmçinin konsoldan çağırıq ki, işimizi davam etdirə bilək.

Aşağıda göstərilən proses açar faylin generasiyasını açıqlayır, hansı ki, şifrələyici üçün əsas açarın hissəsi olacaq. Diskimizi **/gelicrypt** qovluğuna mount edəcəyik. Bu açar faylin köməkliyi ilə təsadüfi verilənlərin yığımı alınır, hansı ki, onunla əsas açar şifrələnir. Bundan əlavə, o, kod sözə şifrələnəcək. Provayder sektorunun həcmi **4kB** olacaq.

Əsas açar, kod sözə qorunacaq. Açar fayl üçün verilənlər isə **/dev/random** virtual aletindən alınır. Tərəfimizdən yaradılan şifrələyici provayder həcmi **/dev/da2.eli - 4kB**.

```
# mkdir /gelicrypt                                - Mount edəcəyimiz qovluğu yaradırıq
# dd if=/dev/random of=/root/da2.key bs=64 count=1
# geli init -s 4096 -K /root/da2.key /dev/da2
Enter new passphrase: şifre
Reenter new passphrase: şifre_təkrar
```

Nəticədə bizə aşağıdakı əmr çap olunacaq:

Metadata backup can be found in **/var/backups/da2.eli** and  
can be restored with the following command:

**# geli restore /var/backups/da2.eli /dev/da2**

Açar fayl və kod sözün eyni vaxtda istifadə edilməsi mütləq deyil. Əsas açarın müdafiəsi üçün bu üsullardan hansısa biri istifadə edilə bilər.

**cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2**

- Bu üsulla bir neçə açar faylından istifadə edə bilərik.

**# geli attach -k /root/da2.key /dev/da2** - Generasiya edilmiş açarı provayder ilə əlaqələndirək. Bu halda yaradılmış disk aletinin adı **/dev/da2.eli** olacaq.

Enter passphrase: **şifre**

Fayl sistemi yaradaq və sonra diskimizi **/gelicrypt** qovluğuna mount edək:

```
# dd if=/dev/random of=/dev/da2.eli bs=1m
# newfs /dev/da2.eli
# mount /dev/da2.eli /gelicrypt
```

```
df -H | grep eli
/dev/da2.eli      5.2G    8.2k    4.8G    0%    /gelicrypt
```

- şifrelənmiş diskimizə baxırıq.

```
umount /gelicrypt
geli detach da2.eli
```

- Diskimizi sistemdən ayırırıq.  
- Provayderi deaktiv edirik.

StartUP skriptin istifadə edilməsi üçün **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə etmək lazımdır. Burada **da2** diski geli provayder kimi qurulub və **/root/da2.key** açar faylı ilə əlaqələnmişdir, ancaq kod sözü sistem işə düşdükdə daxil ediləcək(qeyd edək ki, bu, yalnız **geli init** inisializasiyasında -P açarı istifadə edildikdə mümkün olur):

```
geli_devices="da2"
geli_da2_flags="-p -k /root/da2.key"
#geli_tries=""
#geli_default_flags=""
#geli_autodetach="YES"
#geli_swap_flags="-a aes -l 256 -s 4096 -d"
```

**Qeyd:** Əgər swap-ı şifreləmək istəsəniz, sadəcə **/etc/fstab** faylında, swap bölünmüş diskin sonuna şifreləmə tipi olaraq **.eli**, ya da **.bde** yazmağınız kifayətdir.

Aşağıdakı sətirlərdəki kimi:

```
/dev/da0p3.eli    none        swap      sw      0      0
/dev/da0p3.bde    none        swap      sw      0      0
```

Sistem yenidənyüklənməsindən sonra **swapinfo -h** əmrini daxil etdikdə aşağıdakı nəticəni əldə etmiş olacaqsınız:

Device	1K-blocks	Used	Avail	Capacity
/dev/da0p3.bde	1016752	0B	993M	0%

# BÖLÜM 5

Qrafik interfeysin qurulması,  
kitabxanaların idarə edilməsi, Kernel  
kompilyasiyası, sistem control, İNETD,  
DevFS, DevD, sistemin yenilənməsi

- / FreeBSD X11 quraşdırılması, paylaşılmış kitabxanaların idarə edilməsi və GPART fayl sistem genişlənməsi
- / Kernel kompilyasiya edilməsi, **SYSCTL** idarəedilməsi, **INETD**
- / CheckSUM(İnformasiya bütövlüyü), Device File System(Devfs), DEVD quraşdırılması
- / İstifadəçilərin sistemə əlavə edilməsinin avtomatlaşdırılması
- / FreeBSD əməliyyat sisteminin, mənbə kodlarının və portların fərqli üsullarla yenilənməsi

*Başlığımız əməliyyat sistemimizdə X11-in, yeni qrafik rejimin yüklenməsi və qurulmasını açıqlayır. Sistemdə olan kitabxanaların idarə edilməsi izah olunur. Əgər bir diskdə fayl sistem müəyyən qismə yazılsarsa və fiziki diskdə boş yer qalarsa, başlıq həmin boş yerin mövcud qovluğa yenidən format edilmədən yazılmasını açıqlayıır.*

*Kernelin işləməsi, kernel modullarının idarə edilməsi və kernelin yenidən kompilyasiya edilməsi qaydası, SYSCTL vasitəsilə sistem resurslarının kerneldə təyin edilməsi və idarə edilməsi, sistem qosulan yeni alətlərin sistemdə hansı iş prinsipi ilə avtomatik tanınması barədə danışılır. Həmcinin əməliyyat sistemimizin fərqli üsullarla yenilənməsi qaydaları haqda məlumat verilir.*

# FreeBSD X11 quraşdırılması, paylaşılmış kitabxanaların idarəedilməsi və GPART fayl sistem genişlənməsi

## FreeBSD X11 quraşdırılması

FreeBSD əməliyyat sistemi üzərində GUI-nin quraşdırılması üçün X11 quraşdırılın işə salmaq lazımdır. Bu başlığımızda məhz bu işi görəcəyik.

```
pkg install xorg
```

- **Xorg** paketini yükleyirik.  
**Xorg** öz ardınca (7.4 versiyadan başlayaraq elədir) asılılıqlarında "**hal**" və "**dbus**" yükleyəcək. Ancaq işə düşməsi üçün onları startup-a əlavə etmək lazımdır.  
(yəni **/etc/rc.conf**)

```
/etc/rc.conf
hal_enable="YES"
dbus_enable="YES"
```

- Hald və DBUS-i startup-a əlavə edirik.  
- Ancaq avtomatik işləmə bəzi hallarda və bəzi avadanlıqlarda işləməyə bilər.  
- Bunun üçün əl ilə işə salmaq lazımdır.

X11 quraşdırmaq çox addımlı bir işdir. Ona görə də quraşdılmalarımıza başlayaqq.

```
xorg -configure
```

- Öncə quraşdırma faylı yaradırıq (root istifadəçi ilə edirik). Bu halda **/root** qovluğunda da xorg.conf.new skelet faylı yaranacaq.

```
xorg -config xorg.conf.new
```

- Sonra isə yaranan quraşdırma faylini test edirik. (Yaxşı halda ancaq qara ekran çap olunur.)

```
xorg -config xorg.conf.new -retro
```

- Kohnə yoxlanış variantı, əgər aq-qara setka və mouse kursoru "X" kimi əmələ gəldisə, demək, hər şey işlədi. Testin bitməsini yoxlamaq istəsəniz, "**Ctrl+Alt+Backspace**" eyni anda sıxın.

**Qeyd:** 7.4 və daha yuxarı versiyalarda işləməsi üçün istənilən terminalda bu əmri sıxın: =>  
**setxkbmap -option terminate:ctrl\_alt\_bksp**  
Ya da "x11-input.fdi" adlı fayl yaradıb bu ünvanda saxlayıraq "**/usr/local/etc/hal/fdi/policy**"

```
ee /usr/local/etc/hal/fdi/policy/x11-input.fdi
```

- Fayla aşağıdakı sətirləri əlavə edirik.  
(Mülləq reboot-a ehtiyac var)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbOptions" type="string">terminate:ctrl_
alt_bksp</merge>
    </match>
  </device>
</deviceinfo>
```

```
setxkbmap -model pc102 -layout fr
```

- Burada "**pc102**" tipli klaviaturanı "**fr**(fransız)" dilində təyin edirik.

```
/usr/local/share/X11/xkb/rules/base.lst
```

- Bütün klaviatura tiplərinə bu faylda baxa bilərik.

```
ee xorg.conf.new
```

- Və **X** faylini öz zövqümüzə görə qururuq.  
(Horizontal və vertikal uzunluq, monitorun tezliyi).

```

Section "Monitor"                                - Yalnız bir bölümde olan müqayisədir.
    Identifier      "Monitor0"
    VendorName     "Monitor Vendor"
    ModelName      "Monitor Model"
    HorizSync      30-107
    VertRefresh    48-120

EndSection

Option      "DPMS"                            - Əgər monitorumuz DPMS dəstekləyirsə, DPMS
                                                opsiyasını "Monitor" bölümünə əlavə etmək
                                                lazımdır. DPMS -> Video kart ilə monitor arasında
                                                olan program idarəsidir. (Yəni enerjiyə
                                                qənaət quraşdırması).

Section "Screen"                            - Rəngin dərinliyi "Screen" bölümündə təyin edilir.
    Identifier      "Screen0"
    Device         "Card0"
    Monitor        "Monitor0"
    DefaultDepth   24          # "DefaultDepth" rəngin dərinliyi.
    SubSection     "Display"
        Viewport    0 0
        Depth       24
        Modes       "1024x768"    # Ekran həcmi.

    EndSubSection
EndSection

/var/log/Xorg.0.log                          - Xorg jurnal faylları bu tipdə yaranır.

Quraşdırma bitdikdən sonra faylı "Xorg" oxuduğu ünvana nüsxələmək lazımdır.

cp xorg.conf.new /etc/X11/xorg.conf          - Ya da "cp xorg.conf.new /usr/local/etc/X11/xorg.conf"

startx                                         - Nəticədə qrafik interfeysi işə salırıq.
                                                # Testin bitməsini yoxlamaq istəsəniz,
                                                "Ctrl+Alt+Backspace" eyni anda sıxın.

```

Gnome interfeysi yüklemek istəsək, işləməsi üçün mütləq "Linux type file System" procfs-i mount etmək lazımdır.

Bunun üçün aşağıdakılari edirik.

```
ee /etc/fstab  
## linux type file system.  
proc          /proc      procfs  rw      0      0
```

```
pkg install gnome3-3.14.2
```

- Disk Startup faylımiza procfs-i əlavə edirik.

```
ee /etc/rc.conf  
gdm_enable="YES"  
gnome_enable="YES"
```

- Gnome qrafik interfeysi yükleyirik.

- Startup-a əlavə edirik.  
- Gnome Display Manager

**Qeyd:** Serverimizə X-dən istifadə edən istifadəçinin yetkisini idarə eləmək istəsəniz, '**xhost**'-dan istifadə edin.

```
xhost  
xhost +  
  
xhost -
```

- İzin verilmiş host-ları siyahilayın.  
- Access control-u tamam bağlayın ki, hər kəs qoşula bilsin.  
- Access control-u işə salın, yalnız qeydiyyatda olan istifadəçilər daxil ola bilərlər.

```
xhost remotemachine
```

- Uzaq maşını access control siyahısına əlavə edirik.

```
export DISPLAY=localhost:0  
xcalc &
```

- **Display-i localhost:0-a** təyin edirik.  
- Uzaq kalkulyatoru client maşınınında açırıq.

## Paylaşılmış kitabxanaların idarəedilməsi

Əməliyyat sistemimizdə həddən artıq kitabxanalar var, hansı ki, müəyyən proqramlar tərəfindən istifadə edilir. Bu kitabxanaların düzgün idarə edilməsi və təyinatı inzibatçılar üçün çox önemlidir. Yüklənmiş hansısa bir proqramın istifadə etdiyi kitabxanaların siyahısını öyrənmək tələbi istənilən an yaranı bilər.

```
ldconfig -r
```

- Sistemdə istifadə olunan bütün kitabxanaları çap edəcək.

**Qeyd:** Əgər biz sistemə yeni shared library qovluq əlavə etmişikse, onu **ldconfig**-ə əlavə etmək lazımdır. **ldconfig** quraşdırma sətrini "**/etc.defaults/rc.conf**" ünvanından götürmək olar.

<b>ee /etc/rc.conf</b>	- Startup faylimizə aşağıdakı sətri əlavə edirik
<b>ldconfig_paths="/usr/lib/compat /usr/local/lib /usr/local/lib/compat/pkg"</b>	
<b>ldconfig_local_dirs="/usr/local/libdata/ldconfig"</b>	
<b>/usr/lib/compat</b>	- FreeBSD-nin digər versiyaları ilə uyğunlaşmaq üçün kitabxana bu ünvandan götürülür.
<b>/usr/local/lib/compat/pkg</b>	- FreeBSD-nin digər versiyaları ilə uyğunlaşmaq üçün köhnə kitabxanalar bu ünvandadır.
<b>/lib</b>	- Susmaya görə ldconfig bu ünvana baxır.
<b>/usr/lib</b>	- Susmaya görə ldconfig bu ünvana baxır.
<b>setenv LD_LIBRARY_PATH /home/camal/lib:/tmp/testlibs</b>	- Test ələmək üçün öz şəxsi kitabxanalarımızı yükleyə bilərik.
<b>ldconfig_local_dirs</b>	- Bütün portlar və paketlər kitabxana üçün bu dəyişənin verdiyi cavabı oxuyur.
<b>ldd /bin/sh</b>	- Shell-in istifadə etdiyi kitabxanaları çap edir.

### GrowFS - GPART fayl sistem genişlənməsi

Deyək ki, sizin istifadə etdiyiniz fiziki diskdə bir neçə ədəd UFS slice-lar mövcuddur və bu fiziki diskdə hələ də boş, istifadəsiz yer mövcuddur. Tələb ondan ibarətdir ki, diskin istifadəsiz hissəsini bu slice-lardan birinə əlavə yer olaraq verək.

**Qeyd:** Bu proses umount edilmədən mümkün deyil.

Bizim **/dev/dal** diskimiz var və həcmi **20GB**-dir. Ancaq **/dev/dals1** slice-ında **10GB** istifadə edilir və qalan **10GB**-ı artırmaq (**resize**) lazımdır.

```
umount /dev/dals1
```

- Diskimizi umount edirik

(FreeBSD10.1 **/dev/dalp1**)

```
gpart resize -i 1 dal
```

- Əmri daxil edirik və deyirik ki, **dal** diskimizin ilk slice-i üçün mövcud həcmi tam olaraq artır.

Aşağıdakı cavab qayıtmış olacaq.

```
dals1 resized
```

```
growfs /dev/dals1
```

- Burada boş olan tam həcmin **/dev/dals1** diskinə verilməsini deyirik. Aşağıdakı cavab qayıtmış olacaq.

It's strongly recommended to make a backup before growing the file system.  
OK to grow filesystem on **/dev/dals1** from 10GB to 20GB? [Yes/No] No

```
gpart resize -i 1 -s 12GB dal
```

- Bu əmrde isə deyirik ki, **dal** diskinin ilk slice (**/dev/dals1-e**)-ni **12GB** edin. Yəni **2GB** artırın.  
Çünki **/dev/dals1** diskini özü elə **10GB** idi.

```
growfs /dev/dals1
```

- Və artırılmış disk formatlarıyıq

It's strongly recommended to make a backup before growing the file system.  
OK to grow filesystem on **/dev/dals1** from 10GB to 12GB? [Yes/No] Yes

super-block backups (for fsck -b #) at:

21798272, 23080512, 24362752

```
gpart show
```

- Bu əmr ilə mövcud disklərimizə baxırıq

```
=>      34  314572733  da0  GPT  (150G)
          34        128    1  freebsd-boot  (64k)
          162  306184064    2  freebsd-ufs  (146G)
  306184226    8388540    3  freebsd-swap  (4G)
  314572766            1      - free -  (512B)
```

```
=>      63  41942977  dal  MBR  (20G)
          63  29360079    1  freebsd  [active]  (14G)
  29360142  12582898            - free -  (6G)
```

```
=>      0  29360079  dals1  BSD  (14G)
          0  29360079            - free -  (14G)
```

```
=>      0  29360079  ufsid/540c350afa600d42  BSD   (14G)
      0  29360079                           - free - (14G)
```

**gpart delete -i 1 dal** - **dal** diskinin 1-ci slice-nı silirik.  
**gpart add -t freebsd -il -s 10GB dal** - **dal** diskinə 2-ci slice artırırıq.  
**dalsl added**

**gpart add -s 4G -t freebsd-swap -l swap -i2 dal**  
- Bu əmr ilə SWAP əlavə edirik. Deyirik:  
- **-s 4G** həcm ilə  
- **-t** tipi **freebsd-swap**  
- **-l** label (Bizim halda **swap** ad verdik)  
- **-i2** diskin 2-ci hissəsində

# **Kernel-in kompilyasiya edilməsi, SYSCTL idarə edilməsi, INETD**

## **Kernel kompilyasiya edilməsi**

Kernel: Avadanlıq ilə Program təminatı arasında olan interfeysə deyilir.

- 1. Dynamic Load olunan Modullar (DLMs):** '/boot/kernel' qovluğundadır.
- Kernel modulları yükleyir: '/boot/loader.conf' faylından

```
'kenv'  
'pciconf -lv'
```

- Kernel quruluşunu çap edir. '**bootfile=kernel**'
- Qoşulmuş avadanlıqları çap edir (2-ci başlıqda daha geniş yazılıb).

**Qeyd:** Əgər sistemin fiziki alətləri haqda detallı məlumat əldə etmək istəsək, onda "**dmidecode**" paketini yüklemək lazımdır.

```
cd /usr/ports/sysutils/dmidecode/  
make install clean  
rehash
```

- Port ünvanına daxil oluruq.
- Yükləyirik.
- Sistemdə olan binary faylların işlədiyi ünvanların bazasını yeniləyirik ki, əmrimiz dərhal işə düşün.

**dmidecode**

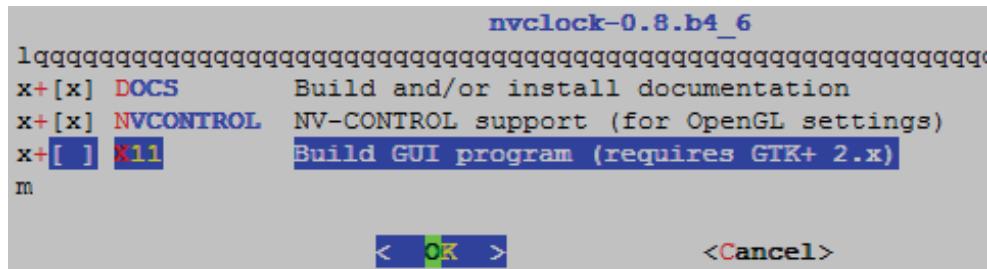
- Və BIOS-dan tutmuş serverin daxilində olan bütün alətlər detallı informasiya ilə çap olunacaq.

**dmidecode 2.7**

- Eynilə bütün informasiyanı çap edir.

**Qeyd:** Video Prosesorun tezliyini və digər xarakteristikasını manipulyasiya etmək üçün "nvctrl" adlı utilit var. Yalnız nvidia CARD-lar üçün keçərlidir.

```
cd /usr/ports/sysutils/nvclock - Port ünvanına daxil olurq.  
make config - Lazımi modulları seçirik.
```

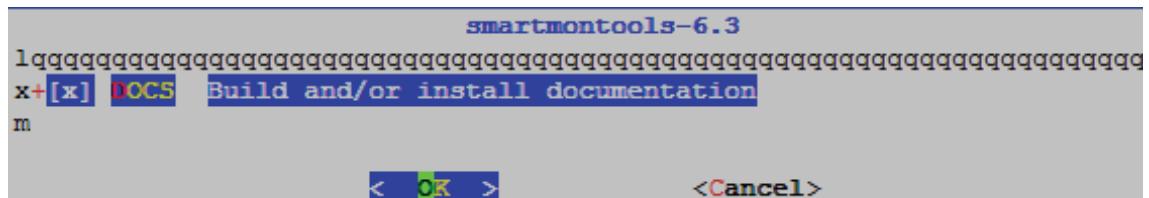


**make install clean** - Yükleyirik.  
**rehash**

- nvclock -s** - Hal-hazırkı tezliyi öyrənirik.
- nvclock -n 300** - Tezliyi 300 Mhz təyin edirik.

**Qeyd:** Sərt disk haqqında detallı məlumatə baxmaq üçün S.M.A.R.T utilitindən istifadə edilir. Yalnız Chipsetimiz bunu dəstəkləməlidir.

```
cd /usr/ports/sysutils/smartmontools/  
make config
```



- Yükleyirik.
- **da0** hakkında etrafı informasyonu cap edecek.

**Qeyd:** Əgər disklərimiz haqqında gündəlik hesabatlar istəsək, `/etc/periodic.conf` faylinə göstərilən sintaksisə `daily_status_smart_devices="/dev/ad0 /dev/da0"` disklərimizi sadalamaq vətər.

**Qeyd:** Sistem yenidən yüklenməsində işləməsi üçün `/usr/local/etc/smartd.conf.sample` faylinı `/usr/local/etc/smartd.conf` fayla nüsxələyib, lazımı quraşdırılmalar edilir və `/etc/rc.conf` (StartUP) faylinə `smartd_enable="YES"` sətri əlavə edilir.

**Qeyd:** USB alətlər haqda məlumat əldə etmək istəyirsinizsə, freebsd versiyalarından asılı olaraq "`usbdevs`" əmrindən, ya da "`usbconfig`" əmrindən istifadə edirik.

Kernel iki bacarığa əsaslanaraq kompilyasiya edilir.

1. '`options`' - software imkanlar üçün.
2. '`device`' - hardware imkanlar üçün.

**Qeyd:** Əgər yeni kernel (`/boot/kernel/kernel`) sistemi yükleyə bilmirsə, siz kernel-i `'/boot/kernel.old/kernel'`-dən götürə bilərsiniz.

Kernelin hansından yüklenməsini dəyişmək istəsək, sistemi reboot edirik. Sonra sistemə '`Escape to loader prompt`' 6-ci seçim rejimində daxil oluruq. Orada '`ls boot/kernel*`' edib, təpələn siyahıdan lazım olanı seçib, '`boot boot/kernel.old/kernel`' - edirik. Və `reboot`.

**kldstat** - Kernel-in yüklenmiş modullarını çap edir.  
**kldstat -v** - Kernel-in dinamik təhrik edilmiş modullarını çap edir (yeni hansı imkanlarla kompilyasiya edilmişdir, driver və modullar).

**kldstat -n acpi.ko** - Yüklenmiş modulun statusuna baxaq.  
**kldstat -v -n kernel** - Yüklenmiş modula detallı şəkildə baxırıq.  
(Submodulları burada görə bilərik.)

**kldstat -v -i 1** - 1-ci ID altında duran modulu verbose rejimdə submodullarla birləkde çap edir.

**kldload smbfs.ko** - Samba Fayl sistemin modulunu yükleyir.  
**kldload -v vesa.ko** - Vesa modulunu verbose modda yükleyir.

**Qeyd:** Əgər biz sistemdə hər hansı aparat elementini istifadə eləmək istəsək, kernel modullarını onun üçün yükleyə bilərik.

**cd /boot/kernel** - Modullar yüklenən qovluğa daxil oluruq. Bütün modullar "`/boot/kernel`"-də olur.  
**kldload modulename** - Yazdığımız modulu yükleyir.

```
kldunload modulename  
/boot/loader.conf
```

- Yüklediyimiz modulu çıkarır.  
- Yüklenmiş modullar bu faylla StartUP-a əlavə edilir.

```
cp -Rp /boot/kernel /boot/kernel.good
```

- Hər hal üçün işləyən kernel-imizi rezerv nüsxə edirik.

Kernel imkanları:

<b>CPU:</b>	Kernel hansı tipli prosessoru yükleyəcək?(x86, ia64)
<b>ident:</b>	Yüklenəcək kernel-in identifikasiya adı.
<b>makeoptions:</b>	Kompilyasiya olunan kernel-in Debug olması üçün istifadə olunur. (DEBUG=-g)
<b>options:</b>	Sistemə əlavə olunan imkanlardır. (Module)
<b>device:</b>	Kernel dəstəklənməsi üçün alətlərdir.

Kernel faylları:

<b>GENERIC.hints</b>	Yüklenmiş hardware modulları haqda informasiya yiğan fayldır.
<b>MAC</b>	Seçilmiş proseslərə verilmiş yetki. ( <b>Mandatory Access Control</b> )

**Qeyd:** Heç vaxt kernel qovluğunun özündə yerləşən fayllarda birbaşa dəyişiklik etməyin.  
(Mütəqə rezerv nüsxə edib sonra dəyişiklik edin.)

Aşağıdakı sətirlər kompilyasiya ediləcək kernel faylinin müəyyən bir hissəsidir:

```
options      SMP          # Symmetric MultiProcessor Kernel  
device       loop         # Network loopback  
device       wlan         # 802.1Q VLAN support  
device       random        # Entropy device  
device       ether         # Ethernet support  
device       ppp          # Kernel PPP  
device       tun          # Packet tunnel.  
device       gif          # IPv6 and IPv4 tunneling  
device       pty          # Pseudo-ttys (telnet etc)
```

```
mv /boot/kernel /boot/kernel.test  
mkdir /boot/kernel  
cp /boot/kernel.good/* /boot/kernel/  
nextkernel -k kernel.test
```

- Kernel-i arxiv edirik.  
- Yeni nüsxə üçün ünvan yaradırıq.  
- İşlək olandan bərpa edirik.  
- Növbəti kernel yüklenməsi **kernel.test** kernelindən yüklenəcək (FreeBSD8.4).

```
nextboot -k kernel.test
```

- Növbəti **kernel** yüklenməsi **kernel.test** kernelindən yüklenəcək (FreeBSD9.3,10.1).

Kernel susmaya görə istənilən prosessor platforması üçün GENERIC faylında kompilyasiya edilmiş olur. Bu faylin içinde sistemə susmaya görə tələb edilən program təminatları və avadanlıqların siyahısı təyin edilir. Öz tələblərimizə uyğun olan kernel kompilyasiya etmək üçün isə, aşağıdakı qaydadan istifadə etməliyik.

```
cd /sys/`uname -m`/conf
```

- Platformamıza uyğun olan kernel qovluğuna daxil oluruq.

```
cp GENERIC ownkernel
```

- Susmaya görə olan **GENERIC** kernel faylini özümüzə uyğun olan ada nüsxələyirik.

```
cd /usr/src/
```

**make buildkernel KERNCONF=ownkernel**

- **ownkernel** faylini özümüzə uyğun dəyişirik və yalnız mütləq tələb edilənləri saxlayırıq. Əsas odur ki, ident sətrinin qarşısında **ownkernel** adının təyin edilməsini unutmayın.

```
make installkernel KERNCONF=ownkernel
```

- Mənbə kodları yerləşən ünvana daxil oluruq.  
- Deyirik ki, kompilyasiya ediləcək struktur **ownkernel** faylından götürüləcək və kompilyasiya ediləcək.  
- Deyirik ki, **ownkernel** adı ilə kompilyasiya edilmiş kernel fayllarını yüklə.

## SYSCTL idarəedilməsi

SysCTL kernel-dən statusun alınması və ya təyin olunması üçün istifadə edilir. Yəni siz kernel tərəfindən idarə edilən istənilən resurs üçün limiti, yazılış standartını buradan təyin edə, ya da mövcud statusunu əldə edə bilərsiniz.

SysCTL MIB-ləri aşağıdakı hissələrə bölünür:

1.  **kern** (kernel-in Core funksiya və imkanları.)
2.  **vm** (Virtual Memory System)
3.  **vfs** (Filesystem)
4.  **net** (Networking)
5.  **debug** (Debugging)
6.  **hw** (Hardware)
7.  **user** (User interface information)
8.  **p1003\_1b** (POSIX behavior) "Portable Operating System Interface for Unix"

9. **compat** (Hal-hazırkı programla kernel uygunluğu)
10. **security** (Spesific-təhlükəsizlik kernel imkanları)
11. **dev** (Device driver information)

<b>sysctl net.inet.tcp.sendspace</b>	- Göndərilən paket həcmi yeri.
<b>sysctl net.inet.tcp.recvspace</b>	- Qayıdan paket həcmi.
<b>sysctl net.inet.udp.recvspace</b>	- UDP qayıdan paket həcmi. (Əgər hər istifadəçi <b>96 Kb</b> istifadə etsə, onda <b>tcp(send/recv)</b> paketini <b>1000</b> nəfərlik <b>9GB</b> RAM hesablamalıyiq.)
<b>sysctl kern.corefile="/root/%N.core"</b>	- Sistemə yüklənilən istənilən programın <b>coredump</b> faylı <b>"/root"</b> ünvanında " <b>%N programın adı</b> " ilə yazılışın.
<b>sysctl -d kern.corefile</b>	- ' <b>kern.corefile</b> ' dəyişənin ' <b>-d</b> '(açıqlanmasını) qısa formada çap edəcək.
<b>sysctl kern.maxproc</b>	- <b>kern.maxproc</b> controlleri haqda çap edir.
<b>sysctl kern.maxfiles=5000</b>	- <b>kern.maxfiles</b> dəyişəninə <b>5000</b> mənasını tikiir.
<b>sysctl -a</b>	- Sistemdə olan bütün controller-lər çap olunur.

Edilən hər bir dəyişikliyin sistem yenidənyüklənməsindən sonra işləməsini istəsək, "**/etc/sysctl.conf**" faylına yazımaq lazımdır:  
**kern.maxfiles=5000**

Verilən rəqəm sayı ya qoşulma sayı kimi, ya da resurs sayı kimi təyin oluna bilər.

Həm də dəyişən tipli ola bilər.

- 1** - **True** deməkdir.
- 0** - **False** deməkdir.

## INETD (Internet – Super Server)

Gündəlikdə istifadə edilən servislər siyahısı mövcuddur ki, istənilən təşkilatda onlar var. Məhz bu tələbə uyğun olaraq INETD-də standart istifadə edilən xidmətlərin müəyyən bir siyahısı var və onlar bir program vasitəsilə, yəni INETD ilə idarə edilir. Susmaya görə olan bu siyahıdakı xidmətləri, port və ya paketlərdən heç bir yüklemə olmadan rahat istifadə edə bilərik.

Məsələn: **SSH, FTP, TFTP**

Bunun üçün seçdiyimiz daemonun qarşısından comment-ini silməyimiz kifayət edir.

**Qeyd:** Unutmayaq ki, inetd tərəfindən idarə olunan daemon-ların yenidənyüklənmədən sonra işləməsini istəsək, **/etc/rc.conf** StartUP faylına **inetd\_enable="YES"** sətrini əlavə etmək lazımdır.

Inetd ilə sshd-ni aktiv etmək üçün **/etc/inetd.conf** faylında aşağıdakı sətrin qarşısından kommenti silmək lazımdır:

```
ssh      stream  tcp      nowait  root      /usr/sbin/sshd          sshd -i -4
```

Hər bir dəyişiklikdən sonra işləməsi üçün mütləq restart tələb olunur.

**/etc/rc.d/inetd restart**

Məsələn: **TFTP** Server yaradaq.

İmkanları:

1. Sürətli, UDP-bazalı fayl transferi.
2. TFDPD client-lər: Cisco devices, VOIP phones, switches, routers, firewalls.
3. Şəbəkədə saysız alətlərin yüklənməsi işini asanlaşdırır.
4. PXE installation/booting dəstəkləyir.
5. INETD-dən işləyir.
6. Son fayl tələb olunur: '**/tftpboot**'-da ev qovluğunda susmaya görə yazmağa icazəsi yoxdur. (Ancaq qovluğu özümüz yaratmalyıq və yazma lazım olsa, yetki verməliyik.)

TFTP Client:

1. '**tftp 192.168.1.122 69**' - 'client' 'host' 'port'
2. **status** - qoşulduqdan sonra statusu yoxlamaq olar.

**Qeyd:** Susmaya görə TFPD yazdığı faylı: '**nobody**' istifadəçi adından yazıır, o da '**anonymous**'-a bərabərdir.

**Qeyd:** Əgər anonymous yox, **tftp** istifadəçi adından informasiya gəlməsini istəsək, aşağıdakı göstəricilərlə '**tftp**' adlı istifadəçini sistemə əlavə edirik:

```
Username   : tftp
Password   : <random>
Full Name  : TFPD User
Uid        : 1003
Class      :
Groups    : tftp
```

```
Home      : /home/tftp  
Home Mode :  
Shell     : /usr/local/bin/bash
```

**/etc/inetd.conf** faylında tftp-ni təpib qarşısından commenti silirik və bir az dəyişiklik edib özümüzə uyğun olaraq quraşdırırıq:

```
tftp    dgram   udp     wait    root    /usr/libexec/tftpd      tftpd -l -s  /  
tftpboot -w -u tftp
```

Göstərilən sintaksisdə **/tftpboot** ünvanı **tftp** serverin qovluğudur.

- l - List deməkdir.
- s - Source deməkdir.
- w - Qoşulan hər bir host üçün yazma yetkisi verilir. (Bu, olmasa, yalnız məlumatı götürmək olar.)
- u - tftp istifadəçi adından işə salınır.

**Qeyd:** tftp qovluğunun ünvan fərqi yoxdur, ancaq onu yaratmaq lazımdır.  
**mkdir /tftpboot**

**Qeyd:** tftp qovluğuna lazımi yetkini veririk.

```
chgrp tftp /tftpboot  
chmod -R 777 /tftpboot
```

Test üçün GNS3(Şəbəkə avadanlıqları üçün spesifik program təminatıdır)-ün üzərində router quraşdırıb tftp üzərindən beynini dəyişdirə bilərsiniz.

Command Line-dan bir neçə misal çəkək:

```
/usr/sbin/inetd -wW -C 60 Default.
```

- TCP Warpper-i 'w' external və '-W' internal üçün aktiv edirik. Əgər IP ünvanından ən azı '-C' 60 dəfə çağırılıbsa.

```
/usr/sbin/inetd -wW -C 60 -R 128
```

- '-R' Bir dəqiqə ərzində maksimum istifadə edilə biləcək servis sayı **128** olacaq.

```
/usr/sbin/inetd -wW -a 192.168.0.1
```

- Inet-də susmaya görə bütün şəbəkə kartlarında qulaq asır. '-a' məcbur edir ki, **192.168.0.1** IP ünvanında dinləsin. (Əvvəlcə servisi söndürün, sonra yoxlayın.)

```
/usr/sbin/inetd -wW -o
```

- Report tipi linux kimi olsun. HACKER-i aldadır.

# **CheckSUM (İnformasiya bütövlüyü), Device File System(Devfs), DEVD quraşdırılması**

## CheckSUM

Bu imkan faylin və ya qovluğun bütövlüğünü yoxlamaq üçündür. Ən çox təhlükəsizlik məqsədləri ilə istifadə edilir. Həmçinin rezerv nüsxələrin götürülməsində də çox istifadə olunur. Aşağıdakı tipləri mövcuddur:

<b>md5</b>	- 128-bit
<b>sha1</b>	- 160-bit
<b>sha256</b>	- 256-bit

Məsələn: **sha1**, **md5** və **sha256** ilə **rc.conf** faylinin bütövlüğünü yoxlayaqq.

Aşağıdakı əmr BASH mühitində işləyəcək.

```
CFILE=rc.conf ; md5 $CFILE && sha1 $CFILE && sha256 $CFILE
```

```
MD5 (rc.conf) = efaa265ae575b254298ee36c18c4e8a2
```

```
SHA1 (rc.conf) = f2c10f72c4ae071c67fb1da907714c1181cfbd9f
```

```
SHA256 (rc.conf) = 97347e413a096925c499481569a7b115d8d67066ab467bddb41cc92cc2796cdf
```

```
md5 -s UNIX - Adı söz üçün md5 uzunluq əldə edirik.
```

```
MD5 ("UNIX") = 15395d0642f86c5992abbca5e56e2b29
```

## Device File System

### **devfs**

- Unix Device Type File System (əsas alətləri başqa adla link edir.)

**Qeyd:** Öncədən **/etc/sysctl.conf** faylinin sonuna **vfs.usermount=1** sətrini əlavə edin ki, adı istifadəçinin **mount** etmək hüququ olsun. Əks halda, **devfs** üzərində apardığınız testlər uğursuz nəticələnəcək.

### **/etc.defaults/devfs.rules**

- **Global devfs** qaydalar faylidir. Heç bir quraşdırma bu fayla əlavə edilmir və dəyişiklik olunmur. Bütün əlavələr və dəyişikliklər mütləq **/etc/devfs.rules** faylında edilməlidir. **/etc/devfs.rules** faylinin sonuna aşağıdakı sətri əlavə etməklə deyirik ki, **cavid** adlı istifadəçinin **dal** diskinə tam yetkisi olacaq. Artıq **cavid** adlı istifadəçi adı ilə sistemə daxil olub **dal** diskini format edə bilərsiniz.

```
[userrules=5]
add path 'dal*' mode 0660 user cavid
#add path 'dal*' mode 0660 group cavid - Eyni ilə qrup üçün edə bilərik.
```

Sonra isə **/etc/rc.conf** faylına **userrules** StartUP-da işə düşməsi üçün əlavə edirik:  
**devfs\_system\_rulesets="userrules"**

<b>/etc/devfs.conf</b>	- Əsas quraşdırma faylinin içinde
#action        realdevice      desiredvalue	
link            acd0            cdrom	- "acd0" alətini "cdrom" adına link edirik. Artıq sistemin yenidənyüklənməsindən sonra <b>/dev/cdrom</b> adlı virtual alət <b>/dev/cd0</b> virtual alətinə link edilmiş olacaq və biz <b>/dev/cdrom-u</b> birbaşa mount edə bilərik.

**Qeyd:** FreeBSD8.4 üçün **cdrom** alətin adı **acd0**, FreeBSD9.3,10.1 üçün isə **cd0**-dir.

perm            acd0            666	- "acd0" alətinə <b>666</b> yetkisi veririk.
own            da20            cavid:cavid	- "da20" alətini <b>cavid</b> istifadəçisi və qrupuna üzv edirik.

## DEVD Konfiqurasiyası

DEVD avadanlıq statusunu dəyişmək üçün daemon-dur. '**/etc/devd.conf**' və '**/usr/local/etc/devd/devd.conf**' faylından quraşdırımları oxuyur.

DEVD-in əsas 4 tipi var.

1. **Attach**. (Alət qoşulan kimi ona aid olan quraşdırmanı işə salır.)
2. **Detach**. (Aləti sistemdən ayıran kimi işə düşür.)
3. **Nomatch**. (Alət qoşulur və ona uyğun təyin olunmuş DEVICE Driver tapılmadığı halda işə düşür.)
4. **Notify**. (İstənilən alətin işə düşüb, sönməsi haqda məlumat verir, bütün xəbərdarlıqlar "**/var/log/messages**" faylinə yiğilir.)

Məsələn:

```
/etc/devd.conf faylinə aşağıdakı sətirləri əlavə etsək,  
notify 0 {  
    - notify adlı və bir prioritetli qayda yalnız o vaxt  
    işə düşəcək ki,  
    match "system"      "IFNET";      - sistemin IFNET şəbəkəsində,  
    match "type"        "ATTACH";     - və tipi attach olanda,  
    action "/etc/pccard_ether $subsystem start"; - göstərilən əmri işə salın ki, şəbəkəni işə salsın.  
};
```

Ya da sistemə Wireless Network Card qoşmuşuqsa,

```
attach 0 {  
    - sistem avtomatik olaraq alətə "wi" adı verir.  
    media-type "802.11";  
    - istənilən 802.11 tipli alət  
    action "/etc/pccard_ether $device-name start"; - əgər alət yuxarıdakı qaydalara uyğundursa,  
    - əmr işə düşəcək.  
};
```

Özümüz qayda yazaq.

```
/usr/local/etc/devd/devd.conf  
notify 10 {  
    match "system" "IFNET";  
    match "type" "LINK_UP";  
    media-type "ethernet";  
    # action "/etc/rc.d/dhclient start $subsystem";  
    action "/root/jobnetwork.sh"; - Özünüz yazdığınız skripti işə salın.  
};
```

Ya da elə edək ki, serverimizə USB External Storage qoşanda, o, automount olsun.

```
attach 10 {  
    match "device-name" "umass0";  
    action "/root/mount_usbstorage.sh";  
};
```

- 10 rəqəmli NOTIFY qayda  
- Düzgündür, əgər alet adı "umass0" olsa.  
(Bütün usb aletlər qoşulanda umass adı alır.)  
- Əgər öncəki sətirlər düzgün olsa, linkdəki skripti işə salın.

```
/root/mount_usbstorage.sh  
#!/bin/sh  
/bin/sleep 2  
# USB storage diskimizin təpiləşməsini avtomatlaşdırırıq və dəyişənə təyin edirik  
d=`dmesg | tail -n 13 | grep umass-sim | grep da | awk '{ print $1 }'`  
/usr/bin/true > /dev/$d  
/usr/local/bin/ntfs-3g /dev/"$d"s1 /media
```

- Skriptimizin məzmunu aşağıdakı kimi olacaq.  
- İki saniyə fasılə verin.  
- İlk Usb Storage disk \$d dəyişənindən olduğu doğrudur.  
- Və sonda disk NTFS formatında mount edirik.

# İstifadəçilərin sistemə əlavə edilməsinin avtomatlaşdırılması

Bizdən tələb oluna bilər ki, müəyyən standart quruluşa əsaslanan **5000** istifadəçini sistemə əlavə etmək lazımdır. Bu halda biz, sözsüz ki, hər bir istifadəçini əlimizlə sistemə bir-bir əlavə etməyəcəyik. Bu işi **/etc/adduser.conf config** faylinin sayesində avtomatlaşdıracaqıq.

## **adduser**

- Bu əmr istifadəçini əlavə etmək üçün utilitidir.  
(Ancaq bu utililit vasitəsilə bir anda minlərlə istifadəçi əlavə etmək olar)

**Qeyd:** Öncə istifadəçilər əlavə ediləcək "group"-lar yaradılmalıdır.

Yəni **3** group-u "**/etc/group**" faylinə əlavə edək.

**/etc/group** faylinə aşağıdakı **3** sətri əlavə edirik.

```
newuser1:*:3000:  
newuser2:*:3001:  
newuser3:*:3002:
```

Və ardınca "users.txt" faylı yaradıb ora istifadəçiləri əlavə edək.

```
newuser1:3000:3000::::Kamil Babayev:/home/newuser1:/bin/sh:12345  
newuser2:3001:3001::::Cavid Bayramov:/home/newuser2:/bin/csh:67890  
newuser3:3002:3002::::Faxri Iskandarov:/home/newuser3:/bin/tcsh:1234567890
```

**Qeyd:** "users.txt" faylda olan 10-cu sütun şifrələrdir, ancaq bu şifrələr '/etc/master.passwd' faylinə şifrələnib yazılaçaq. İstifadəçilər sistemə əlavə edildikdən sonra users.txt faylini mütləq silin ki, sistemə yetkisi olan digər şəxslərin əlinə keçməsin.

**adduser -f users.txt**

- Beləliklə, **users.txt** faylında olan istifadəçilər sistemə əlavə olunacaq.

Bütün istifadəçilər üçün təyin olunan **default** quraşdırmaları "/etc/adduser.conf" ünvanından da götürmək olar. Ancaq bu faylı özümüz redaktə etməli deyilik, əmrlə avtomatik yaratmalıyıq.

**adduser -C**

- -C əmri faylin avtomatik yaradılması və sonda çıxması üçün istifadə edilən spesifik opsiyadır.

Uid (Leave empty for default):

Login group []:

Enter additional groups []:

Login class [default]:

Shell (sh csh tcsh bash rbash nologin) [sh]: **bash**

Home directory [/home/]:

Home directory permissions (Leave empty for default):

Use password-based authentication? [yes]:

Use an empty password? [yes/no] [no]:

Use a random password? [yes/no] [no]:

Lock out the account after creation? [no]:

Pass Type : yes

Class :

Groups :

Home : /home/

Home Mode :

Shell : /usr/local/bin/bash

Locked : no

OK? [yes/no] : **yes**

Re-edit the default configuration? [yes/no]: **no**

Goodbye!

Beləliklə, yeni istifadəçi əlavə ediləndə, o, susmaya görə olan quraşdırmanı bu fayldan oxuyacaq.

```
cat /etc/adduser.conf
```

- Quraşdırma faylinin məzmunu aşağıdakı kimi olacaq:

```
# Configuration file for adduser(8).  
# NOTE: only *some* variables are saved.  
# Last Modified on Sun Dec 18 18:54:19 UTC 2011.
```

```
defaultHomePerm=
```

```
defaultLgroup=
```

```
defaultclass=
```

```
defaultgroups="mühəsibatlıq marketing IPS"
```

- Susmaya görə '/etc/login.conf'-dan götürülür.

```
upwexpire=120d
```

- Susmaya görə istifadəçi yaradılanda onları seçilmiş qruplara əlavə edə bilərik.

```
upwexpire=31-12-2015
```

- İstifadəçi şifrəsinin vaxtı 120 gündən sonra bitəcək.

```
passwdtype=no
```

- İstifadəçi şifrəsinin vaxtı 2015-ci il dekabrın 12-də bitəcək.

```
passwdtype=none
```

- Susmaya görə istifadəçilərə şifrə təyin etmirik, yəni qoşula bilməyəcəklər.

```
homeprefix=/home
```

- Olarsa, şifrə yoxdur, şifrəsiz qoşula biləcəklər.

```
defaultshell=/usr/local/bin/bash
```

- İstifadəçi nöqtə ilə başlayan bütün informasiyanı bu qovluqdan alır (Amma **bash** üçün gərək aşağıdakılari edək.)

```
udotdir=/usr/share/skel
```

- İlk istifadəçi identifikasiatoru 3003 rəqəmi ilə başlayacaq.

```
msgfile=/etc/adduser.msg
```

- Susmaya görə istifadəçiye təyin edilən siyasetdir, hansı ki, susmaya görə istifadəçi sistemə əlavə edildikdə istifadəçi şifrəsi '/etc/passwd' faylinə yazılır, ev qovluğu yaradılır, sonra '/usr/share/skel' qovluğunda olan skelet fayllar istifadəçinin ev qovluğununa nüsxələnir, ardınca istifadəçinin qrupu '/etc/group' faylinda yaradılır və istifadəçi öz qrupunun üzvü olur. Həmçinin

```
disableflag=
```

```
uidstart=3003
```

```
defaultclass
```

```
adduser -d /usr/people
```

istifadəçinin e-maili '**/var/mail**' qovluğunda yaradılır.

```
adduser -s /usr/local/bin/bash
```

- Sərt olaraq təyin edirik ki, bütün istifadəçilərin ev qovluğu '**/usr/people**'-da olsun.

```
adduser -k /usr/share/myskel
```

- Sərt olaraq təyin edirik ki, bütün istifadəçilər üçün susmaya görə olan shell '**bash**' olacaq.

```
adduser -N
```

- Susmaya görə istifadəçilər skelet fayllarını '**/usr/share/skel**'-dən yox, '**/usr/share/myskel**'-dən götürəcək.

```
adduser -D
```

- İstifadəçi üçün quraşdırmaları '**adduser.conf**'-dan götürməyin. Sistemin özündən susmaya görə götürün.

```
adduser -L dialer
```

- Susmaya görə yeni istifadəçi əlavə edildikdə, ona ev qovluğu yaratmayın.

- '**dialer**' adlı yeni login class istifadə edin.

**Qeyd:** login class susmaya görə bütün siyasetləri '**/etc/login.conf**' faylından götürür. Orada susmaya görə '**default**', '**news**' və bir neçə başqa class name-lər var. Ona görə də öncə '**dialer**' class yaratmaq lazımdır ki, sonra istifadə edə bilək. Susmaya görə normal istifadəçi siyasetləri '**default**' class-dan götürür.

**/etc/login.conf** faylında olan default class-in açılması:

```
default:\
```

```
:passwd_format=md5:\
```

- '**/etc/master.passwd**' faylında olan istifadəçi şifrələrinin şifrələnməsi '**md5**', '**des**' və ya '**blf**' ola bilər.

```
:copyright=/etc/COPYRIGHT:\
```

- '**/etc/COPYRIGHT**' faylı müəllif hüquqları haqqında olan fayldır, özümüz sistem yazsaq, müəllif hüquqlarını buraya yaza bilərik.

```
:welcome=/etc/motd:\n\n:setenv=MAIL=/var/mail/$,BLOCKSIZE=K,FTP_PASSIVE_MODE=YES:\n\n:path=/sbin /bin /usr/sbin /usr/bin /usr/games /usr/local/sbin /usr/local/bin\n/usr/X11R6/bin ~/bin:\n:nologin=/var/run/nologin:\n\n:cputime=unlimited:\n:datasize=unlimited:\n:stacksize=unlimited:\n\n:memorylocked=unlimited:\n\n:memoryuse=unlimited:\n\n:filesize=unlimited:\n\n:coredumpsize=unlimited:\n\n:openfiles=unlimited:\n\n:maxproc=unlimited:\n\n:sbsize=unlimited:\n\n:vmemoryuse=unlimited:\n\n:priority=0:\n
```

- '/etc/motd' bu fayl istifadəçi ilk dəfə daxil olanda ekrana çıxan salam mesajıdır. Bu fayla istədiyimiz mesajı yaza bilərik.

- Susmaya görə olan təyin edilmiş dəyişənlər, MAIL,BLOCKSIZE,FTP\_PASIVE\_MODE

- Sistem əmrlərinin PATH-ları.

- '/var/run/nologin' əgər bu fayl sistemdə yaradılıbsa, bütün istifadəçilərin sistemə giriş qadağandır. Administrator müvəqqəti bütün istifadəçilərin sistemə girişini qadağan etmək istəsə, bu faylı yarada bilər. Diqqət!! Mütləq sonra silməyi unutmayın, əks halda istifadəçilər daxil ola bilməyəcəklər. ☺

- CPU-dan istifadə etmək limiti vaxtla təyin edilir.

- Ümumiyyətlə, disk space-a limit təyin edirik, həcmə.

- Stacke təyin edilmiş maksimum həcm. Həcmə.  
(stack viki 'lifo' prinsiplə işləyir.)

- Bu siyaset üçün maksimum **lock** edilən yaddaş. Həcmə.

- Bu siyaset üçün maksimum istifadə edilə biləcək yaddaş. Həcmə.

- Bu siyaset üçün faylin həcminin maksimum qədəri. Həcmə.

- Kerneli dump etdikdə maksimum istifadə edilə biləcək həcm. Həcmə.

- Proses sayına görə maksimum açıla biləcək fayl. Rəqəmlə.

- İstifadə edilə biləcək proseslərin maksimum sayı. Rəqəmlə.

- Maksimum icazə verilmiş sockbuffer həcmi. Həcmə.

- Proses sayına görə maksimum istifadə edilə biləcək VM həcmi. Həcmə.

- Bu siyaset təyin edilmiş istifadəçilərdə 'nice' prioritet proseslərdə '0'(sıfır) olacaq.

**:ignoretime@:\**

- Nə Man-da, nə də google-da haqqında məlumat yoxdur.
- Susmaya görə istifadəçilər qovluq yaradanda **755**, fayl yaradanda isə **644** yetkisi alacaq.

**cap\_mkdb /etc/login.conf**

- **Qeyd:** **login.conf** faylında edilən hər dəyişiklikdən sonra onun bazasını yeniləmək lazımdır. Göstərilən '**cap\_mkdb**' əmri ilə.
- '**/etc/passwd**' faylini redaktə etmək üçün istifadə olunur. İstifadəsi ona görə mütləqdir ki, faylı '**vipw**' ilə açıldıqda, onun prosesini bloklayı ki, nüsxələmək mümkün olmasın.

**Qeyd:** **adduser.conf** haqda bütün detallı məlumatı '**man adduser.conf**' əmri ilə almaq olar.

# FreeBSD əməliyyat sisteminin, mənbə kodlarının və portların fərqli üsullarla yenilənməsi

## FreeBSD Update

Əməliyyat sistemində olan imkanlar həmişə təkmilləşir və təkmilləşmələr yeni versiyalarla istismara buraxılır. Ancaq yenilənmə təkcə yeni imkanlar üçün deyil, həmçinin köhnə buraxılışlarda olan yanlışlıqları və boşluqları aradan qaldırmaq üçün də istifadə olunur. Ən çox təhlükəsizlik məqsədlərinə əsaslanaraq sistemi gündəmdə saxlamaq lazımdır.

FreeBSD yenilənməsinin **3** yolu var.

1. **sysinstall** (FreeBSD 8.4)
  - a. HALF-GUI ilə UPGRADE seçirik. **sysinstall -> upgrade**
2. **binary** (FreeBSD8.4, 9.3 və 10.1)
  - a. Yenilənmə üçün hazır binar fayllar mövcuddur.
    - a1. Binar yenilənmələr iki hissədən ibarətdir. (Yenilənmələri endirmək və onları mənimşətmək) **/etc/freebsd-update.conf** - binar yenilənmənin quraşdırma faylıdır.
3. **source** (Mənbə)
  - a. Məsələn: 7.5 versiyasının mənbə kodları bizdə olsa, onu kompilyasiya edib, yenilənmə edə bilərik.
    - a1. Mənbə kodları CVSUP(FreeBSD8.4) və SVNUP(FreeBSD9.3,10.1) repository serverlərdən alınır.

## FreeBSD əməliyyat sisteminin binar üsulla yenilənməsi

Bu üsulla olan yenilənmə FreeBSD (8.4, 9.[123], 10.0, 10.1-RC[1234]) versiyaları üçün keçərlidir. Həmçinin kiçik versiyadan böyüya keçilən halda **/boot/kernel** və **/boot/Generic** əvəzlənə bilər. Həmçinin **/usr/src** qovluğu mövcuddursa, silinəcək və yeniisi ilə əvəz ediləcək.

**freebsd-update fetch**

- Mövcud yüklenmiş dünya kodlarına və təyin olunmuş opsiyalara əsaslanaraq, bütün binar yenilənmələri gətirəcək.

**freebsd-update install**

- Ən son gətirilmiş yenilənmələri yükleyəcək.

**Qeyd:** Əgər ən son yüklenmiş yenilənmələrdə problem yaranarsa, siz **freebsd-update rollback** əmri ilə ən son yüklenmiş yenilənmələri silə bilərsiniz.

**freebsd-update upgrade -r 10.1-RELEASE** - Bu əmr ilə biz deyirik ki, (-r versiyası) **FreeBSD 10.1** versiyasına yenilə.

**freebsd-update install**

- Öncə sistemin kernel-i yenilənir və kernel-ə aid olan komponentlər yüklenmir, çünki onların yüklenməsi mütləq əməliyyat sisteminin yenidənyüklənməsindən sonra olmalıdır.

**shutdown -r now**

- Yenidənyüklənmədən sonra **freebsd-update(8)** təkrarən işə salınmalıdır ki, istifadəçilərə aid olan komponentləri yüklesin.

**freebsd-update install**

- Bu hissədə FreeBSD 9.3 daha əvvəlki versiyadan ən sona yenilənmiş oldu. Sistemin kitabxanalarının ən son versiyaya yenilənməsi səbəbindən, yüklenmə müddətində **freebsd-update(8)** təklif edəcək ki, portlar və 3-cü tərəf program təminatlarını yeniləsin.

**freebsd-update install**

- Sistemin üçüncü tərəf programlarının yenilənməsindən sonra yenidən **freebsd-update(8)**-i işə salın ki, (yalnız **freebsd-update(8)** özü sizə buna ehtiyacın olduğu haqqında

məlumatı çap edərsə), o, köhnə, uzun müddət istifadə edilməyən sistem kitabxanalarını silə bilsin.

**shutdown -r now**

- Sonda **FreeBSD 10.1-RELEASE** içindən yenidənyüklənmə edək.

### FreeBSD əməliyyat sisteminin CSUP vasitəsilə yenilənməsi

**Qeyd:** Nəzərə alın ki, CVSUP artıq dayandırılma mərhələsindədir və SVN-ə keçid yaxın vaxtlarda mütləq tələb olunacaq.

**/usr/share/examples/cvsup**

- CVSUP üçün fayl nüsxələri bu ünvanda yerləşir.

**cvs-supfile**

- Bu supfile bütün FreeBSD Source repository-larını dəstəkləyir. (Ancaq Developerler üçündür.)

**doc-supfile**

- Əməliyyat sisteminin daxili sənədləşməsinin yenilənmə faylı.

**gnats-supfile**

- Bu zədəli FreeBSD bazalarının hesabatı üçün istifadə olunur.

**ports-supfile**

- Bu supfile sistem portlarını ən son versiyası ilə yeniləyir.

**stable-supfile**

- Bu supfile stabil mənbə kodunu yeniləmək üçün istifadə olunur.

**standart-supfile**

- Bu supfile istifadə etdiyimiz versiyaya aid olan son mənbə kodlarını yeniləyir.

**www-supfile**

- Bu supfile FreeBSD WEBSITE-in ən yeni (linklərini) versiyasını dərtir.

Bu supfile-lardan bizə uyğun olanı seçib, **/etc/** qovluğuna nüsxələyirik.

**cp /usr/share/examples/cvsup/cvs-supfile /etc/supfile**

**/etc/supfile**

- Faylda sətirləri aşağıda göstərilənə uyğun olaraq quraşdırırıq.

**\*default host=CHANGE\_THIS.FreeBSD.org**

- Öncə REPO server təyin edirik. Sintaksis:  
**cvsup<number>.<countrycode>.freebsd.org**  
Məsələn: **cvsup15.us.FreeBSD.org**  
servnumber.ölkəkodu.systemdomain  
Məsafə yaxınlığını tapmaq üçün hostu əvvəl ping-lə yoxlayın. Ya da portlardan  
**/usr/ports/sysutils/fastest\_cvsup**  
yükleyin ki, ən yaxın ünvanı avtomatik təpə biləsiniz (**fastest\_cvsup -c all** əmri avtomatik ən yaxın ünvanı axtarır təpəcək).

**\*default base=/var/db**

- Bu sətir **csup-a** deyir ki, yüklənilən faylların siyahısı və statusu bu ünvanda saxlanılacaq.

**\*default prefix=/home/ncvs**

- Bütün mənbə kodları bu ünvanda saxlanılacaq.

**\*default release=cvs tag=RELENG\_10.1**

- Burada "tag" freebsd versiyasını çap edir.

**\*default delete use-rel-suffix**

- Bu CVSUP-in baza qovluğunun paylaşma imkanını yaradır.

**\*default compress**

- CPU həcmini və keçidi yadda saxlayır.

**src-all tag=.**

- Bütün mənbə kodları "tag=" (öncə təyin etdiyimiz versiya)" endirilsin.

**ports-all tag=.**

- Bütün yeni portlar "tag=." endirilsin.

**projects-all**

- Bütün projektlər.

**cvsroot-all**

- Bütün CVSroot hint-lər.

Sonra yenilənmə üçün təyin etdiyimiz qovluğu yaradaq və yenilənmə əmrimizi işə salaq:

**mkdir -p /home/ncvs**

- Qovluğu yaradırıq.

**cvsup /etc/supfile**

- Və yenilənmə başlanır.

**CSUP** ən son versiyalı mənbə kodlarını endirdikdən sonra işimizi **MAKE** əmrinə ötürmək və sistemi ən son mənbə kodlarından yeniləmək lazımdır. Öncə ən son versiyalı mənbə kodlarını "**/home/ncvs**" ünvanından **/usr** qovluğuna (**mv /home/ncvs/src /usr/**) köçürməliyik. Sonra isə mənbə kodlarının ünvanına **/usr/src-a** daxil olub, kompilyasiya və yükləmə işlərini görmək lazımdır. Ancaq biz bu işi aşağıda başqa üsulla edəcəyik.

**Qeyd:** Əgər siz **/etc/supfile**-da mənbə kodlarının ünvanını öncədən **/usr/src** təyin etmiş olsanız, bütün işi sadəcə **csup /etc/supfile && make buildworld && make buildkernel** əmri ilə edə bilərsiniz.

#### Sistemimizin CSUP və MAKE ilə yenilənməsi

Biz sistemin mənbə kodlarından avtomatik yenilənməsi üçün bu üsuldan istifadə edə bilərik. Əvvəlcə hər iki fayl "**/etc/make.conf**" və "**/etc/src.conf**"-a aşağıdakı dəyişənləri əlavə edək:

```
SUP_UPDATE=yes  
SUP=/usr/bin/csup  
SUPHOST=cvsup9.us.freebsd.org  
SUPFILE=/usr/share/examples/cvsup/stable-supfile  
PORTSSUPFILE=/usr/share/examples/cvsup/ports-supfile  
DOCSUPFILE=/usr/share/examples/cvsup/doc-supfile
```

```
cd /usr/src  
make update && make buildworld && make buildkernel
```

- Sonra mənbə kodları ünvanına daxil oluruz  
- və kodların yenilənməsi, dünyanın yiğilması və kernel-in yiğilması əmrini daxil edirik.

**Qeyd:** Əgər sistemi mənbə kodlardan yeniləsək, bizim çoxlu imkanlarımız var ki, orada susmaya görə olan quruluşları dəyişək. Ona görə də yiğmağa başlamazdan önce bizə lazım olan dəyişənləri "**/etc/make.conf**" və "**/etc/src.conf**"-a əlavə etməliyik. Qurulma dəyişənlərinin siyahısını bu linkdən <https://www.freebsd.org/cgi/man.cgi?query=src.conf> əldə etmək olar. Məsələn: **WITHOUT\_BIND=YES** Yeni kompilyasiya edilmiş sistemdə susmaya görə **bind** olmayıcaq.

#### Sistemimizin SVN vasitəsilə yenilənməsi

CVSUP-in dayandırılması səbəbindən SVN-ə keçid məcburi tələbdir və bu başlığımızda SVN-i açıqlayacaq. Həmçinin SVNUP vasitəsilə də yenilənməni açıqlayacaq.

```
cd /usr/ports/devel/subversion  
make install
```

- SVN port ünvanına daxil oluruq.
- Yükləyirik.

```
pkg install subversion
```

- Eynilə hazır paket yükləyə bilərik.

SVN anbarı təyin etmək üçün URL-lərdən istifadə edir, hansı ki, **protocol://hostname/path** formasından götürülür. SVN fərqli protokollar dəstəkləyə bilər. Göstərilən URL ünvanında olan ilk komponent Repository-nin tipidir. Üç fərqli repository mövcuddur, **base** - FreeBSD mənbə kodları bazası, **ports** – FreeBSD portlar kolleksiyası, **doc** – sənədləşmə üçün. Misal üçün, **svn://svn0.us-east.FreeBSD.org/ports/head/** linkində deyir ki, protokol **SVN** ilə **svn0.us-east.FreeBSD.org** adlı repository-ə müraciət edirik.

```
svn checkout svn-mirror/repository/branch lwcdir
```

- Bu əmr ilə müəyyən repository-dən öz daxili ünvanlarımıza sinxronizasiya edəcəyik.

Burda:

- **svn-mirror**

- URL-dir, hansı ki,  
<https://www.freebsd.org/doc/handbook/svn.html#svn-mirrors>  
linkindən bu siyahını əldə edə bilərsiniz.

- **repository**

- Repository layihelərindən biri - **base**, **ports**, ya da **doc**.
- İstifadə edəcəyiniz repository-dən asılıdır. Adətən ən çox yenilənən **ports** və **doc** reposları **head** branch-də yerləşir. Ancaq **base** reposunun **-CURRENT** versiyası head-də -STABLE versiyası isə **stable/8** (8.x üçün), **stable/9** (9.x üçün) və **stable/10** (10.x üçün) yerləşir.

- **branch**

- İstifadə edəcəyiniz repository-dən asılıdır. Adətən ən çox yenilənən **ports** və **doc** reposları **head** branch-də yerləşir. Ancaq **base** reposunun **-CURRENT** versiyası head-də -STABLE versiyası isə **stable/8** (8.x üçün), **stable/9** (9.x üçün) və **stable/10** (10.x üçün) yerləşir.

- **lwcdir**

- Mənsəb qovluqdur, hansı ki, təyin edilmiş branch üçün kontent burda yerləşəcək. Adətən bu portlar üçün **/usr/ports**, baza üçün **/usr/src** və sənədlər üçün isə **/usr/doc** olur.

Aşağıdakı misal /usr/ports ünvanına US WESTERN reposundan **HTTPS** protokol istifadə edərək portları yenileyəcək. Əgər **/usr/ports** qovluğu mövcuddursa və ya svn tərəfindən yaradılmayıbsa, onu mütləq silin.

```
svn checkout https://svn0.us-west.FreeBSD.org/ports/head /usr/ports
```

İllkin yoxlanış qismən uzun çəkə bilər. Səbrli olmaq lazımdır. Illkin yoxlanışdan sonra daxili işlək qovluq aşağıdakı əmr ilə yenilənə bilər:

```
svn update lwcdir
```

Öncəki misalımızda yaratdığımız yenilənməni silmək üçün aşağıdakı əmrden istifadə edin (Yenilənmə yoxlanışdan qat-qat tez gedəcək, çünki yalnız dəyişiklik edilmiş qovluq və fayllar sinxronizasiya ediləcək):

```
svn update /usr/ports
```

Daxili işlek qovluqların yenilənməsinin başqa üsulu isə, həmin qovluqlarda (**/usr/ports**, **/usr/src**, ya da **/usr/doc**) yaranan **Makefile** sayesindədir. Bunu aşağıdakı kimi edəcəyik (Misal üçün, **/usr/src**):

```
cd /usr/src
make update SVN_UPDATE=yes
```

- Mənbə kodları ünvanına daxil oluruz.  
- Yeniləyirik.

Aşağıdakı əmrlərlə bütün yenilənilə biləcək ünvanlarımız FreeBSD rəsmi saytından edilir:

```
svn co svn://svn.FreeBSD.org/ports/head /usr/ports
svn co svn://svn.FreeBSD.org/doc/head /usr/doc
svn co svn://svn.freebsd.org/base/releng/10.1/ /usr/src
```

Mənbə kodları, sənədlər və portların gələcək yenilənməsi sadəcə onların öz qovluqlarına daxil olub aşağıdakı əmrin işə salınması ilə edilir:

```
svn update
```

Həmcinin SVNUP-dan da istifadə edə bilərsiniz.

SVNUP açıqlanması SVN-in daha da asanlaşdırılmış versiyası kimidir.

```
cd /usr/ports/net/svnup
make install clean
```

- Port ünvanına daxil oluruz.  
- Yükləyirik.

Mənbə kodlarının və portların avtomatlaşdırılmış yenilənməsi üçün **/usr/local/etc/svnup.conf** quraşdırma faylında aşağıdakı sətirlərin qarşısından kommenti silməniz kifayətdir:

```
host=svn.freebsd.org
host=svn0.us-west.freebsd.org
host=svn0.us-east.freebsd.org
```

Mənbə kodları və portların əldə edilməsi üçün aşağıdakı əmrləri daxil edirik:

```
svnup stable
svnup ports
```

Hal-hazırkı mənbə kodlarının əldə edilməsi üçün svnup current əmrini daxil etmək lazımdır. Mənbə kodlarını əldə etdikdən sonra **/usr/src/UPDATING** faylına nəzər yetiririk. Bu faylda hər bir versiyalar arası dəyişiklik edilən kimi fərqlər qeyd olunur.

SVNUP-la portları yeniləyirik:

```
svnup -h svn0.us-west.FreeBSD.org -b ports/head -l /usr/ports
```

SVNUP-la mənbə kodlarını yeniləyirik:

```
svnup -h svn0.us-west.FreeBSD.org -b base/stable/10.1 -l /usr/src
```

# BÖLÜM 6

## Şəbəkənin quraşdırılması, WireLess quraşdırılması, WEBMIN, ARP, FTP, DHCP, şəbəkə alətləri

- / Şəbəkə kartının quraşdırılması, şəbəkə alətləri və Routing (şəbəkənin yönləndirilməsi)
- / WireLess quraşdırılması
- / Əməliyyat sistemimizin WEB browser vasitəsilə idarə olunması və şəbəkənin keçirmə qabiliyyətinin yoxlanılması
- / ARP, FTP local servis və DHCP serverin quraşdırılması
- / Şəbəkə utilitləri

Ethernet və wireless şəbəkə kartının ilk quraşdırılması, şəbəkə alətləri və şəbəkədə istifadə edilə biləcək başlanğıc əmrlər bu başlıqda açıqlanır. Şəbəkədə olan problemlərin araşdırılması üçün daxili utilitlər, spesifik program təminatlarının yüklənməsi və istifadə qaydaları göstərilir. Şəbəkənin statik üsulla yönləndirilməsi, şəbəkədə keçirmə qabiliyyətinin yoxlanılması, ARP istifadəsi, FTP və DHCP daemonlarının işə salınması, spesifik program təminatı vasitəsilə serverimizin WEB browser ilə idarə edilməsi izah olunur.

# Şəbəkə kartının quraşdırılması və şəbəkə alətləri

FreeBSD şəbəkə kartlarının adları Linuxda olduğu kimi, eth\* adla adlandırılmıştır. Hər bir şəbəkə kartının driver-inə əsaslanan qısa adla təyin edilir. Bu driver adları kernel-də olan driver strukturunda hər bir brend üçün açıqlanır(device em # Intel PRO/1000 Gigabit Ethernet Family). Misal üçün, şəbəkə kartımızı bir neçə üsulla quraşdırma bilərik. Yarım qrafik üsul və ya konsolla. FreeBSD8.4-ü yarım qrafik rejimdə quraşdırmaq üçün:

**sysinstall -> configure -> networking -> interface -> em0**

FreeBSD9.3 və FreeBSD10.1 versiyalarında yarım qrafik rejimdə quraşdırmaq üçün (eynilə Default Gateway və DNS quraşdırmları da Network Management bölümündən edilir):

**bsdconfig -> Network Management -> Network Interfaces -> em0**

Konsolla aşağıdakı üsulla edirik:

**ifconfig em0 192.168.1.20 netmask 255.255.255.0**

- **em0** şəbəkə kartına IP ünvanı və MASK yazılır.

**route add default gw 192.168.1.1**

- Yazılmış IP ünvanı susmaya görə olan şəbəkə yolu təyin edirik.

**Qeyd:** Unutmayın ki, konsol ilə daxil edilmiş şəbəkə quraşdırması sistemin ilk yenidənyüklənməsində silinəcək. Yarım qrafik rejim isə özü avtomatik olaraq StartUP faylı **/etc/rc.conf**-a lazımi sətirləri əlavə edir.

StartUP faylı **/etc/rc.conf**-un sintaksisi aşağıdakı kimidir:

```
ifconfig_em0="inet 192.168.121.129 netmask 255.255.255.0"  
defau ltrouter="192.168.121.2"
```

Aşağıda bir neçə qosulma tiplərini açıqlayırıq:

**Ethernet CARD(em0,bge0,bce0,fxp0)**

- Fiziki Ethernet şəbəkə kartı.
- PLIP-lə digər kompüterlərə birbaşa qosulma.
- Serial qosulma aləti.
- null modem qosulması, ya da iki nöqtə qosulması.

**Paralel port IP(plip0)**

- dhclient əmri ilə serverimizə IP ünvan aldıqda, baş vermiş addımlar haqqında məlumatlar bu faylda saxlanılır.

**Serial Line Internet Protocol(cuad0)**

- Serverimzdə adların IP ünvanlarına çevrilə bilməsi üçün bu fayla ad serverləri lazımi sintaksislə əlavə edilir. Sintaksis aşağıdakı kimidir.

**Point-to-Point Protocol (ppp0)**

**/var/db/dhclient.leases\***

**/etc/resolv.conf**

- Əgər siz daxili şəbəkənizdə olan alt domain-lərin autocomplete edilib resolv olmasını istəyirsinizsə, bu sətir işinizi yarayacaq. Yəni siz **ping pub** əmrini daxil edən kimi sistem susmaya görə ilk olaraq **freebsd.org** domain altında pub adına axtarış verəcək.

**domain freebsd.org**

- Eynilə domain gördüyü işi görür, sadəcə burada bir neçə domain adı qeyd etmək olar.

**search freebsd.org openbsd.org**

**nameserver 188.72.128.20**

- DNS olmayan vaxtlarda IP-nin ada çevrilməsi üçün resolv faylı bu idi. Sintaksisi aşağıdakı kimidir.

**nameserver 188.72.128.10**

**/etc/hosts**

<b>127.0.0.1</b>	<b>localhost localhost.az</b>	
<b>188.72.128.12</b>	<b>elcin.az elcin</b>	
<b>188.72.128.12</b>	<b>elcin.az.</b>	
 <b>/etc/protocols</b>		- Dünya standartlarında təsdiqlənmiş bütün protokollar bu faylda yerləşir.
 <b>/etc/services</b>		- Dünya standartlarında təsdiqlənmiş bütün şəbəkə xidmətləri və onların portları bu faylda yerləşir.
 <b>ifconfig em0 down</b>		- Şəbəkə kartını söndürürük.
<b>ifconfig em0 up</b>		- Şəbəkə kartını işə salırıq.
<b>ifconfig -d</b>		- Bu əmrlə söndürülmüş şəbəkə kartlarının siyahısını görə bilərik.
<b>ifconfig -u</b>		- İslək şəbəkə kartlarımızın siyahısını görə bilərik.
 <b>ifconfig em0 192.168.1.10/32 add</b>		- Əmrlə <b>em0</b> şəbəkə kartımıza 2-ci (buna Alias deyilir) İP ünvanı təyin edirik.
 <b>ifconfig em0 inet 192.168.1.102/32 delete</b>		- Əmrlə <b>em0</b> şəbəkə kartımızda olan 2-ci (buna Alias deyilir) İP ünvanı silirik.
 <b>ifconfig -m em0</b>		- <b>em0</b> şəbəkə kartının driver-i haqqında məlumatları çap edir.
 <b>ifconfig em0 media 100baseTX mediaopt full-duplex</b>		- <b>em0</b> şəbəkə kartının <b>100</b> megabit sürətlə və <b>full-duplex</b> rejimində işləməsini təyin edirik.
 <b>ifconfig em0 media 10baseT/UTP</b>		- <b>em0</b> şəbəkə kartının <b>10</b> megabit-lə, <b>UTP</b> kabel tipi ilə işləməsini deyirik.
 <b>ifconfig em0 192.168.1.250/24 media 1000baseTX mediaopt full-duplex</b>		- <b>em0</b> şəbəkə kartına 1Gb şəbəkə və <b>full-duplex</b> rejimi ilə <b>IP</b> ünvan təyin edirik.

```
ifconfig em0 name eth0
```

- **em0** şəbəkə kartına eth0 adı təyin edirik.

**Qeyd:** Yuxarıda çəkdiyimiz misalların StartUP-da işləməsini istəsək, onları StartUP quraşdırma faylinə əlavə etmək lazımdır. Sadəcə aşağıdakı sətirlərdən tələbinizə uyğun olanı "**/etc/rc.conf**" faylinə əlavə etməniz kifayətdir.

```
ifconfig_em0="inet 10.0.0.208 netmask 255.255.255.0 media 100baseTX"  
ifconfig_re0="192.168.1.250 255.255.255.0 media 1000baseTX mediaopt full-duplex"  
ifconfig_em0="inet 192.168.121.129 netmask 255.255.255.0 name eth0"  
ifconfig_em0_name="eth0"
```

- **Qeyd:** Bu halda şəbəkə kartına IP təyin edəndə mütləq 'eth' şəbəkə kartına IP ünvanı vermək lazımdır.

```
wire-test
```

- Şəbəkə kartlarını, daxili IP ünvanları, TCP və UDP protokolları test etmək üçün utilit-dir.

```
systat -ifstat 1
```

- Şəbəkə kartlarının ümumi keçirtmə qabiliyyətini çap edir.

```
/etc/netstart
```

- Şəbəkə quraşdırmasını **/etc/rc.conf** faylinda dəyişiriksə, onda bu əmr sistemimizi yenidənyüklənmə etmədən işə salır.

```
/etc/rc.d/netif restart
```

- Şəbəkəyə aid bütün resursları **restart** edir.

```
/etc/rc.d/routing restart
```

- Bütün routing-ləri restart edir.

Şəbəkəyə aid olan bütün resursları aşağıda göstərilən bir əmrlə restart edirik:

```
/etc/rc.d/netif restart && /etc/rc.d/routing restart && /etc/netstart &
```

**Qeyd:** Bu əmr uzaqdan SSH vasitəsi ilə yerinə yetiriləndə sessiya da atacaq və yenidən qoşulacağıq. Əgər firewall quraşdırılıbsa, sizin sessiya qırılacaq və SSH vasitəsilə yenidən qoşula bilməyəcəksiniz. Çünkü susmaya görə ən sondakı **ACL**-də hər kəsin hər yerə gediş bağlıdır.

**Qeyd:** Əgər FreeBSD serverimiz birbaşa yox, hansısa HTTP və ya FTP proxy ilə internetə çıxsa, onda aşağıdakı sətirləri konsol-dan daxil edərək serverinizi internetə çıxara bilərsiniz.

```
HTTP_PROXY="http://192.168.0.1:3128"
```

- Bu halda 192.168.0.1 IP ünvanlı HTTP proxy serverin 3128-ci portu ilə internete çıxırıq.

```
FTP_PROXY=ftp://192.168.0.1:3128
```

- Bu halda 192.168.0.1 IP ünvanlı FTP proxy serverin 3128-ci portu ilə internete çıxırıq.

```
FETCH_ENV=FTP_PROXY=ftp://user:pass@192.168.0.1:3128
```

- fetch əmri ilə internete çıxışını FTP proxy ilə təyin edirik.

```
FETCH_ENV=HTTP_PROXY=http://user:pass@192.168.0.1:3128
```

- fetch əmri ilə internete çıxışını HTTP proxy ilə təyin edirik.

**Qeyd:** Kompilyasiya vaxtını azalda bilmərik, ancaq internetdən etdiyimiz endirimin sürətini artırıa bilerik. Biz ən yaxın olan ölkədən FreeBSD FTP Mirror listini internetdən tapa bilərik. Bunun üçün <http://google.com> axtarış motorunda **freebsd ftp mirrors** başlığı ilə axtarış etsəniz, həmin listi tapa bilərsiniz (<https://www.freebsd.org/doc/en/books/handbook/mirrors-ftp.html>). Taplıqdan sonra isə seçdiyiniz FTP serverləri **"/etc/make.conf"** faylına əlavə etmək lazımdır. Hal-hazırda Rusiya seçilmişdir, çünki bizə ən yaxın olan FreeBSD FTP serverləri Rusiyadadır.

```
/etc/make.conf                                     - Fayla aşağıdakı sətirləri əlavə edirik.
```

```
MASTER_SITE_OVERRIDE?=${MASTER_SITE_BACKUP}
```

```
MASTER_SITE_BACKUP?= \
```

```
ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ports/distfiles/${DIST_SUBDIR}/ \
ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ports/distfiles/${DIST_SUBDIR}/ \
ftp://ftp3.ru.FreeBSD.org/pub/FreeBSD/ports/distfiles/${DIST_SUBDIR}/ \
ftp://ftp4.ru.FreeBSD.org/pub/FreeBSD/ports/distfiles/${DIST_SUBDIR}/
```

### **Şəbəkə alətləri (İşimizi asanlaşdırırıq)**

Biz şəbəkə kartlarımıza daxil olan və çıxan paketləri daha da rahat görünən şəkildə etmək istəsək, aşağıdakı program təminatlarını yükləməliyik.

```
cd /usr/ports/net/nload
```

- Port ünvanına daxil oluruq.

```
make install clean
```

- Yükləyirik.

```
rehash
```

- Bütün binar fayllarınızın ünvanlarını yeniləyirik.

```
nload -i 10240 -o 10240 em0
```

- **nload** əmri **em0** şəbəkə kartında girişə və çıkışa maksimal həcmdə monitor edir.

```
nload -m
```

- Bütün şəbəkə kartlarımızda hal-hazırkı və ümumi daxil olub-çıxan paketlərin həcminə baxırıq.

```
cd /usr/ports/net-mgmt/iftop  
make install clean
```

- Port ünvanına daxil oluruq.  
- Yükləyirik.

```
iftop
```

- Susmaya görə olan ilk şəbəkə kartının girişini və çıkışını çap edəcək.

```
iftop -i em0
```

- **em0** şəbəkə kartının bütün RX və TX paketlərini MB ilə çap edir.

**Qeyd:** Əgər siz serverinizdə internet sürətini yoxlamaq istəsəniz, aşağıdakı paketdən istifadə edə bilərsiniz:

```
cd /usr/ports/net/py-speedtest-cli  
make install
```

- Port ünvanına daxil oluruq.  
- Yükləyirik.

```
speedtest
```

- Əmri daxil etdikdən sonra nəticəni aşağıdakı kimi görəcəyik.

Retrieving speedtest.net configuration...

Retrieving speedtest.net server list...

Testing from Delta Telecom Inc. (85.132.122.106)...

Selecting best server based on ping...

Hosted by AvirTel LLC (Baku) [14.67 km]: 28.054 ms

Testing download speed.....

Download: 31.08 Mbit/s

Testing upload speed.....

Upload: 24.20 Mbit/s

Və ya başqa bir utilit-lə də trafikə baxa bilərik. Adı **trafshow**-dur. Lakin burada şəbəkə kartına düşən tam yüklenməni yox, kimin hara hansı portla nə qədər paket yolladığını görə bilərsiniz.

```
cd /usr/ports/net/trafshow
```

- Portlardan yükləyirik.

<code>make config</code>	- IPv6-ni söndürürük, çünkü o, bizə lazım deyil.
<code>make install clean</code>	- Yükləyirik.
<code>trafshow</code>	- Əmri daxil etdikdən sonra şəbəkə kartlarımızın seçimi üçün menyu çıxacaq. Oradan bizə lazım olan şəbəkə kartını seçib Enter-i sıxaraq trafik-lərə baxırıq.

**Qeyd:** Əgər bizdə DNS server varsa və öz DNS-imizə daha çox müraciət edənləri görmək istəyiriksə, aşağıdakı paketi yükleyirik.

<code>cd /usr/ports/dns/dnstop</code>	- DNSTOP paketini portlardan,
<code>make install clean</code>	- yükleyirik.
<code>dnstop -Q em0</code>	- <code>em0</code> şəbəkə kartında <code>-Q</code> müraciətlərin sayına baxırıq.

Hostların işlek vəziyyətdə olub, ya da olmamalarının yoxlanışı üçün çox gözəl utilitimiz var. Hansı ki, onun sayəsində hostları fayldan da oxuya bilərik. Həmçinin şəbəkə aralığına ping edə bilerik. Onun adı '**fping**'-dir.

<code>cd /usr/ports/net/fping</code>	- Portlardan,
<code>make install clean</code>	- yükleyirik.
<code>fping &lt; /root/ping_file</code>	- <code>ping_file</code> faylında olan hostların hamısını ' <b>fping</b> ' əmrinə yönəldirik, fping isə öz növbəsində STDIN-dən aldığı hər IP ünvanı ping edir.
<code>cat /root/ping_file</code>	- Faylin tərkibinə isə istədiyimiz IP ünvanları əlavə edirik.
<code>rutracker.org</code>	
<code>mail.ru</code>	
<code>gmail.com</code>	
<code>facebook.com</code>	
<code>nobody.az</code>	- Mövcud olmayan domain. Cavabı: " <b>nobody.az address not found</b> " olacaq.
<code>192.168.1.100</code>	- Mövcud olmayan host. Cavabı: " <b>192.168.1.100 is unreachable</b> " olacaq.
<code>fping -g 192.168.1.0 192.168.1.255</code>	- Burada <code>192.168.1.0</code> şəbəkəsini skan edirik.

```
fping -g 192.168.1.0/24
```

- Öncəki sətir ilə eyni işi görür.

**Qeyd:** Əlavə olaraq qeyd edə bilərik ki, MySQL bazası istifadə etsəniz və orada müraciət sayını monitoring etmək istəsəniz, portlarda '**mytop**' adlı gözəl bir paket var. Siz onu **/usr/ports/databases/mytop** ünvanından yükləyə bilərsiniz. Ancaq onun çatışmayan bir cəhəti var ki, hər bir baza istifadəçisi üçün ayrı quraşdırma yaratmalısınız.

Sistem resurslarının top ilə göstərilən nəticəsinin alternativi kimi daha başa düşülən formada nəticə verən utilit **htop** mövcuddur. Onu aşağıdakı qaydada yükləyib quraşdırı bilərik.

```
cd /usr/ports/sysutils/htop  
make install clean
```

- Port ünvanına daxil oluruq.  
- Yükləyirik.

**Qeyd:** Yüklənmə zamanı program Linux uyğunluqlu olduğuna görə, işləməsi üçün "**linproc**" fayl sistemin mount edilməsini tələb edəcək. Onu mütləq etməliyik. Əks halda **htop** işləməyəcək.

```
mkdir -p /usr/compat/linux/proc  
ln -s /usr/compat /compat
```

- Fayl sistem üçün qovluq yaradırıq.  
- Tələb edilən linki yaradırıq.

```
/etc/fstab  
linproc /compat/linux/proc linprocfs rw 0 0
```

- Faylin sonuna aşağıdakı sətri əlavə edirik.

```
mount linproc
```

- Linproc fayl sistemi mount edirik.

```
make install clean
```

- Yükləməmizə davam edirik.

```
htop
```

- Əmri daxil edirik və nəticəyə baxırıq.

RAM və SWAP-in ayrı əmrlə ümumi statusuna və istifadə edilmişlə qalan həcmələrini görmək istəsek, **freecolor** paketindən istifadə edə bilərik.

```
cd /usr/ports/sysutils/freecolor  
make install
```

- Port ünvanına daxil oluruq.  
- Yükləyirik.

```
freecolor -t -m -o          - Total çıkış RAM-la göstermekle çıkışa çap edin.
      total      used      free     shared    buffers    cached
Mem:       3928       528     3400         0         0         0
Swap:      4095        0     4095
Total:    8024 = (      528 [used] +     7496 [free])
```

## Routing (şəbəkənin yönləndirilməsi)

**route add**

- Bu əmr vasitəsilə şəbəkənin sistemimizə yönləndirilmə işi görülür.

**route add default 188.72.128.1**

- Konsol-dan birbaşa default gateway əlavə olunur, ancaq sistem yenidənyüklənməsində silinəcək.

**route del default**

- Konsoldan susmaya görə olan şəbəkə yolunu silirik.

**route add -net 192.168.2.0/24 192.168.1.2**

- **192.168.2.0** şəbəkəsini görmək üçün **192.168.1.2** gateway-nə statik route əlavə edirik.

**route del -net 192.168.2.0/24 192.168.121.2**

- Öncə yazdığımız şəbəkəyə aid olan route-u silirik.

**route add -host 192.168.1.2 10.10.10.1**

- **192.168.1.2** IP ünvanını görmək üçün gateway-nə statik route əlavə edirik.

**route del -host 192.168.1.2 192.168.121.2**

- **192.168.1.2** IP ünvanına aid olan route-u silirik.

**route get azdatacom.az**

- '**azdatacom.az**' saytına hansı şəbəkə kartı ilə və hansı IP gateway-dən keçdiyimizi çap edir.

**Qeyd:** Ancaq əlavə olunan routing-lər sistemin ilk yenidənyüklənməsinədək işləyəcək. Sonra işləməsi üçün **/etc/rc.conf** faylına yazırıq.

**/etc/rc.conf**

- StartUP faylımiza aşağıdakı sintaksisi əlavə edək ki, sistemimizin yenidənyüklənməsindən sonra lazımi şəbəkələri görə bilək.

**static\_routes="net1 net2"**

- iki route qrup yaradırıq:

**route\_net1="-net 192.168.0.0/24 192.168.0.1"**

- 1-ci qrup routing yazırıq.

**route\_net2="-net 192.168.1.0/24 192.168.1.1"**

- 2-ci qrup routing yazırıq.

# WireLess quraşdırılması

Bu başlığımızda Wireless quraşdırılmasına aid olan bütün görülecek işler açıqlanır.

**cd /usr/ports/sysutils/pciutils**

- **lspci** emrindən istifadə edə bilməyimiz üçün bu

**make install**

portu yüklemək lazımdır.

- Yükləyirik.

**lspci | grep -i wireless**

- Hansı Wireless şəbəkə kartından istifadə etdiyimizi axtarıraq.

**Qeyd:** Əgər sistemimiz tərəfindən tanınmayan brend-dən istifadə etsək, onda "**ndis miniport driver wrapper**"-dən istifadə etməliyik. '**man ndis**' əmri ilə ətraflı oxuya bilərsiniz.

Dəsteklənən Wireless kartlar.

**an** 'an' başlığı ilə dəsteklənən brend-lər.  
Aironet Communications 4500/4800  
Cisco Aironet 340 and 350 series  
Xircom Wireless Ethernet Adapter

- Əgər alətimiz an-dırsa, onda **loader.conf**-a aşağıdakı sətri əlavə edirik.

**if\_an\_load="YES"**

**ath** 'ath' başlığı ilə dəsteklənən brend-lər.  
Atheros 802.11 Wireless Adapter  
Supports all Atheros Cardbus/PCI (except AR5005VL)

**ee /boot/loader.conf**

**if\_ath\_load="YES"**

- Öğər aletimiz **ath**-dirse, onda **loader.conf**-a aşağıdakı sətri əlavə edirik.

<b>Driver adı</b>	<b>Dəstəklənən Wireless səbəkə kartı</b>	<b>/boot/loader.conf-a əlavə edilməlidir</b>
awi	AMD PCnetMobile IEEE802.11 BayStack 650/660 Icom SL-200 Melco WLI-PCM NEL SSMagic Netwave AirSurfer Plus/Pro Nokia C020 WLAN Farallon SkyLINE	<b>if_awi_load="YES"</b>
cnw	Netwave AirSurfer Wireless LAN Xircon CreditCard Netwave	<b>if_cnw_load="YES"</b>
ipw	Intel PRO/Wireless 2100 MiniPCI	<b>if_ipw_load="YES"</b>
iwi	Intel PRO/Wireless 2200BG/ 2225BG MiniPCI/ 2915ABG 802.11 Adapters	<b>if_iwi_load="YES"</b>
ral	Ralink Technology RT2500	<b>if_ral_load="YES"</b>
ural	Ralink Technology RT2500USB	<b>if_ural_load="YES"</b>
wi	Dozens of cardsthat include the following wireless chipsets: Lucent Hermes, Intersil PRISM-II, Intersil PRISM-2.5 and Symbol Spectrun24	<b>if_wi_load="YES"</b>

**Qeyd:** Üçüncü sütunda olan modulları o halda istifadə edirik ki, driver-lər kernel-də kompilyasiya edilməyib. Həmçinin WLAN driver-in özü də yüklənməlidir.

```
ee /boot/loader.conf  
wlan_load="YES"
```

- StartUP-da əlavə edirik.
- WLAN driver-i aktiv edirik.

## WiFi şəbəkə kartının quraşdırılması

```
ifconfig an0 inet 10.0.1.1 netmask 0xffffffff00 - Aironet şəbəkə kartı əlavə edirik və IP ünvan  
veririk.
```

SSID 'net01' və adı **net01** olan atheros şəbəkə kartı əlavə edirik.

```
ifconfig ath0 inet 10.0.1.1 netmask 0xffffffff00 ssid "net01" named net01  
PCnetMobile card (64-bit WEP) əlavə edib, açarı yazaraq aktiv edirik.  
ifconfig awi0 inet 10.0.1.1 netmask 0xffffffff00 ssid "net01" wepmode on wepkey 0x45320185622
```

Eyni işi Intel PRO 2200BG (128-bit WEP) üçün edirik.

```
ifconfig iwi0 inet 10.0.1.1  
netmask 0xffffffff00 ssid "net01" \  
wepmode on wepkey 0x01020304050607080910111213 weptxkey 1
```

<b>ifconfig cnw0 scan</b>	- Scan edirik ki, stansiya tapaq.
<b>ifconfig cnw0 down</b>	- İnterfeysimizi söndürürük.
<b>ifconfig cnw0 up</b>	- İnterfeysimizi işə salırıq.
<b>wicontrol -i wi0 -o</b>	- Wireless kartın statistikasını çap edirik.
<b>wicontrol -i wi0 -C</b>	- Sıqnal gücü haqqında olan məlumatı çap edirik.
<b>wicontrol -i wi0 -X</b>	- Sıqnal gücü haqqında olan məlumatı silirik.
<b>raycontrol -i ray0 -o</b>	- Hal-hazırkı statistikani driver-dən alırıq.
<b>raycontrol -i ray0 -t 3</b>	- Ötürmə sürətini medium təyin edin.
<b>raycontrol -i ray0 -n net02</b>	- Şəbəkə adını <b>net02</b> təyin edin.
<b>ancontrol -i an0 -A</b>	- Əsas Acces Point haqda məlumatı çap edin.
<b>ancontrol -i an0 -S</b>	- Bütün SSID-ləri siyahilayın.

# **Əməliyyat sisteminin WEB browser vasitəsi ilə idarə olunması və Şəbəkənin keçirmə qabiliyyətinin yoxlanılması**

## **FreeBSD WEBMIN**

Əgər tələbat yaransa ki, serverimizin idarə edilməsi əmrlərlə çox çətindir və bizə qismən rahat üsul lazımdır, o halda webmin adlı program təminatından istifadə edə bilərsiniz. Ancaq nəzərə alın ki, bu, inzibati tənbələşdirir və bu web paneldə sistem imkanlarının 90%-i mövcuddur. Lakin dünya IP ünvanında olmayan hansısa bir serverin WEB ilə işləməsi tələbi yaranarsa, bu üsul heç də piş deyil. Aşağıdakı ardıcılıqla WEBmin serverini yükleyib quraşdırırıq.

```
cd /usr/ports/sysutils/webmin  
make install clean
```

- Port ünvanına daxil oluruq.
- Yükleyirik. (Yükləmənin sonunda çap edilən tələblərə diqqətlə fikir verin!!!)

```
cd /usr/local/lib/webmin/  
/usr/local/lib/webmin/setup.sh
```

- Bu ünvana daxil oluruq.
- Skripti işə salırıq ki, webmini quraşdırıq.

```
echo 'webmin_enable="YES"' >> /etc/rc.conf
```

- Startup faylinə əlavə edirik ki, sistemin yenidənyüklənməsindən sonra avtomatik işə düşsün.

```
/usr/local/etc/rc.d/webmin start
```

- webmin daemon-u işə salırıq.

Nəhayət, browser vasitəsi ilə skriptimizin quraşdırılması müddətində təyin etdiyimiz **HTTP/HTTPs** protokol və təyin etdiyimiz port vasitəsilə serverimizə qoşuluruq (**http,s://server\_IP:port**). Aşağıdakı şəklə uyğun nəticə olmalıdır:

**Login to Webmin**

You must enter a username and password to login to the Webmin server on 192.168.121.129.

Username:

Password:

Remember login permanently?

Həmçinin təyin etdiyimiz istifadəçi adı və şifrə ilə qoşulub aşağıda görünən şəkildəki nəticəni əldə etməliyik:

>Login: admin

- Webmin
  - Backup Configuration Files
  - Change Language and Theme
  - Webmin Actions Log
  - Webmin Configuration
  - Webmin Servers Index
  - Webmin Users
- System
- Servers
- Others
- Networking
- Hardware
- Cluster
- Un-used Modules

Search:

[View Module's Logs](#) [System Information](#) [Refresh Modules](#) [Logout](#)

### webmin

**System Information**

- System hostname freebsd10.1
- Operating system FreeBSD 10.1
- Webmin version 1.740
- Time on system Sun Mar 22 21:47:14 2015
- Kernel and CPU FreeBSD 10.1-RELEASE-p6 on amd64
- System uptime 4 hours, 14 minutes
- Running processes 98
- CPU load averages 0.31 (1 min) 0.53 (5 mins) 0.56 (15 mins)
- Real memory 501.25 MB used, 1.98 GB total
- Local disk space 12.88 GB used, 20.79 GB total
- Package updates 4 package updates are available

## Şəbəkənin keçirtmə qabiliyyətinin yoxlanılması

Bize MPLS və ya hansısa bir imkanla 100 megabit, ya da 1 gigabit şəbəkə sürəti verilmiş olarsa, ancaq reallıqda bu sürəti əldə etməsek, köməyimizə iperf(client/server program təminatıdır) adlı utilit gələcək. Ancaq şəbəkəmizin son nöqtəsində də başqa bir FreeBSD, ya da Linux maşın olmalıdır ki, İPerf-ə qulaq assın. Məhz bizim serverdən həmin serverə və ya uzaq serverdən geriyə bizim əməliyyat sistemimizə trafik generasiya edib ötürülməklə şəbəkəmizin keçirtmə qabiliyyətini təyin edə bilərik. Aşağıda iperf paketin yüklənməsi və istifadə qaydası açıqlanır.

```

cd /usr/ports/benchmarks/iperf
make install clean
iperf -s

iperf -c server_IP_Address
iperf -u -s -il -p 65005

iperf -c server_IP_address -u -p 65005
iperf -u -c server_IP_address -l100 -b20k -t180

iperf -n 10m -p 5001 -c 192.168.1.37

iperf -c localhost -p 65005 -b 100M

```

- Port ünvanına daxil oluruq.  
 - Yükləyirik.  
 - Serverdə bunu işə salırıq(server bütün IP ünvanlarında susmaya görə 5001-ci TCP portla qulaq asacaq) -s server  
 - client ilə həmin serverə qoşuluruq. -c client  
 - Serverimiz 1 saniyə intervalı ilə UDP 65005-ci port ilə qulaq asacaq.  
 - Serverimizdə təyin etdiyimiz port və protokola client-dən qoşulub test edirik.  
 - Serverimizə 180 saniyə müddətində saniyədə 20kbit sürətlə 100 bayt uzunluğunda trafik yollayıraq. (Adətən VoIP test etmək üçün istifadə edilir.)  
 - 192.168.1.37 IP ünvanlı serverin 5001-ci portuna 10megabitlik trafik axını yollayıraq.  
 - Serverimizə ötürülən trafikin həcmini -b opsiyası təyin edə bilərik.

**Qeyd:** Həmçinin trafikin generasiya edilməsinin daha irəliləmiş utilit-i iperf3-ə diqqət yetirsəniz, sizə xeyirli olar. iperf3-də paralel bir neçə sessiya üzərindən trafik generasiya etmək imkanı mövcuddur. Ümumiyyətlə, imkan baxımından iperf-dən xeyli fərqlənir. "/usr/ports/benchmarks/iperf3" port ünvanından yükləyə bilərsiniz.

İPerf3 server üçün aşağıdakı əmri daxil etməniz kifayətdir:

```
iperf3 -p 5000 -fM -i 5 -s `ifconfig em0 | grep "inet " | awk '{ print $2 }'` -4
```

İPerf3 Client üçün isə aşağıdakı əmri daxil etməniz lazımdır:

```
iperf3 -p 5000 -f M -i 5 -c Server_IP_Address -4
```

```
iperf3 -h
```

- **IPERF3** haqqında bütün məlumatları bu əmrin çıxışından əldə edə bilərsiniz.

# ARP, FTP local servis və DHCP serverin quraşdırılması

Başlığımızda Address Resolution Protocol, File Transfer Protocol və Dynamic Host Configuration Protocol istifadəsi haqqında danışılır. ARP-in imkanları və real praktikada istifadəsi, FTP serverin asanlıqla işə salınması və portlardan DHCP serverin yüklənilər quraşdırılması müzakirə edilir.

## ARP və onun real praktikada istifadəsi

Təsəvvür edin ki, FreeBSD əməliyyat sistemi müəssisəniz üçün Router olaraq işləyir və istifadəçilər üçün serverin daxili IP ünvanı Default Gateway-dır. Siz server üzərindən daxili şəbəkədə işləyən istifadəçiləri NAT vasitəsilə ilə dünyaya çıxarırsınız. Hər bir istifadəçinin öz vəzifəsinə uyğun olaraq, internetə çıxışı təyin edilmişdir. Şəxsi təcrübəmdən bir misal olaraq açıqlaya bilərəm ki, ağıllı istifadəçi daha yüksək hüquq olan IP ünvanı təpib özündə istifadə edəcək. Məhz bu halda siz şəbəkəni həm 2-ci, həm də 3-cü səviyyədə kontrol etməlisiniz. Bu alt başlıqda bunu detallı şəkildə açıqlayıb qurulmasını göstərəcəyik.

FreeBSD əməliyyat sistemində şəbəkə kartlarımızın MAC ünvanlarını aşağıdakı qayda ilə dəyişə bilərsiniz (Ancaq mövzumuzun məqsədi bu deyil):

**ifconfig em0 ether 11:22:44:55:33:55** - **em0** şəbəkə MAC ünvanını dəyişirik.

**ifconfig em0 link 00:00:a1:b2:c3:a4** - Öncəki əmr ilə eyni işi görür.

Test üçün Windows maşınınımızdan Wireshark vasitəsilə və ping-lə yoxlayırıq. Nəticədə təyin etdiyimiz MAC ünvanı görməliyik.

```
/etc/start_if.em0
```

- Dəyişmiş MAC ünvanın StartUP-da işləməsini istəyirikse, faylin içində aşağıdakı sətri əlavə etməyimiz kifayətdir. Görünən MAC ünvanı dəyişdirilmiş ünvan olmalıdır.

```
ifconfig em0 ether 00:0c:29:bb:f6:14
```

İndi isə keçək serverimizin quraşdırılmasına. Təsəvvür edək ki, serverimizdə **NAT (Network Address Translation)** və **Firewall** artıq quraşdırılmışdır (Gələcək mövzularımızda NAT və FireWall haqqında geniş müzakirələr aparılacaq). İstifadəçilərimiz üçün default gateway olan serverimizin daxili IP ünvanı **10.0.0.1/24**-dur və istifadəçilər bu aralıqdan IP ünvanı istifadə edə bilərlər.

İşimizə başlamazdan önce ARP əmrinin bir neçə imkanlarını açıqlayırıq:

```
arp -a
```

- arp-in öz cədvəlində olan hər bir MAC ünvanları çap edir.

```
arp 10.0.0.10
```

- Yalnız **10.0.0.10** IP ünvanının MAC ünvanını çap edəcək.

```
arp -d 10.0.0.10
```

- **10.0.0.10** IP ünvanının MAC ünvanını silir.

```
arp -d -a
```

- Serverin MAC cədvəlində olan bütün MAC ünvanları silir.

```
arp -s 10.0.0.10 c4:17:fe:7c:74:a8
```

- Təyin edilmiş IP ünvana təyin edilmiş MAC ünvanı tikiłır.

```
arp -f /usr/local/etc/ethers
```

- ARP əmri **/usr/local/etc/ethers** faylında təyin edilmiş IP və MAC düzülüşünü oxuyub sərt olaraq öz cədvəlinə yazar.

Faylin sintaksisi.

```
cat /usr/local/etc/ethers
# Host      MAC-address
10.0.0.1    00:05:5d:ce:d6:3f
10.0.0.2    00:05:5d:29:ec:f4
10.0.0.255  ff:ff:ff:ff:ff:ff
```

**Qeyd:** Unutmayıñ, sonda şəbəkə üçün broadcast MAC ünvanını mütləq yazın. ARP bu mesaj ilə şəbəkəni yoxlayır. Həmçinin şəbəkə kartına 'arp' və ya 'staticarp' təyin etmədən yoxlamayın. Yəni 'ifconfig em1 staticarp' və 'ifconfig em1 arp' etdikdən sonra testlərinizi edin.

**Qeyd:** Əgər avtomatik bütün MAC ünvanlarını fayla yazmaq istəsək, görünən əmrələ bunu edə bilirik. Bu əmr çap olunan ARP ünvanlardan 2-ci və 4-cü sütunu tabulyasiya ilə ayıräraq "ethers" faylinə yazar.

```
arp -an | awk -v OFS="\t" '{print(substr($2, 2, length($2)-2), $4)}' > /usr/local/etc/ethers
```

<code>arp -d -a</code>	- Bütün MAC-ları silirik.
<code>arp -f /usr/local/etc/ethers</code>	- arp öz cədvəlini göstərilən fayldan oxuyub sərt olaraq tikir.

Bu, o deməkdir ki, istifadəçi öz IP ünvanını dəyişsə də, Internet resursları üçün artıq yetki ala bilməyəcək ☺.

Ancaq gördüyüümüz işin avtomatlaşdırılması üçün StartUP-da işləməsini istəsək, aşağıdakı skripti yaradıb lazımi yetki vermək gərəkdir.

<code>/usr/local/etc/rc.d/statarp.sh</code>	- StartUP skript-i yaradırıq.
<code>chmod 777 /usr/local/etc/rc.d/statarp.sh</code>	- Skript-i yerinə yetirilən edirik.

Bu skript-lə **start**, **stop**, **restart** etmək və status-a baxmaq olar.

Ancaq bu skript-dən istifadə etmək məsləhət deyil, çünki bu skript hər dəfə cədvəli silib yenidən yazar.

<code>/usr/local/etc/rc.d/statarp.sh</code>	- Skriptin kontenti aşağıdakı kimidir.
<code>#!/bin/sh</code>	
<code># Static ARP-table loader</code>	
<code>case \$1 in</code>	
<code>    start)</code>	
<code>        arp -d -a &gt; /dev/null</code>	
<code>        arp -f /usr/local/etc/ethers &gt; /dev/null</code>	
<code>        echo 'Static ARP-table is loaded'</code>	
<code>        ;;</code>	
<code>    stop)</code>	

```

        arp -d -a > /dev/null
        echo 'Static ARP-table is unloaded'
        ;;
    restart)
        arp -d -a > /dev/null
        arp -f /usr/local/etc/ethers > /dev/null
        echo 'Static ARP-table is reloaded'
        ;;
    status)
        arp -an
        ;;
*)
    echo "Usage: `basename $0` {start|stop|restart|status}" >&2
    ;;
esac
exit 0

```

Ən yaxşısı seçdiyimiz faylı bir dəfə bütün istifadəçilər işlək vəziyyətdə olanda yaratmaq lazımdır. Sonra start skriptinə ünvanı göstərib işə salmaq lazımdır.

**touch /etc/rc.d/arpstart** - Sistem işə düşəndə bu fayl avtomatik işə düşəcək.

**chmod 777 /etc/rc.d/arpstart** - Skripti yerinə yetirilən edirik.

**/etc/rc.d/arpstart** - Fayla aşağıdakı sətri əlavə etməyimiz yetər ki, sistem yenidənyüklənməsində içindəki əmr avtomatik işə salınsın.

**arp -f /usr/local/etc/ethers**

**Diqqət!!!** Ancaq biz hansı şəbəkə kartının staticARP işini görəcəyimizi mütləq seçmeliyik. Aşağıdakı əmrlərlə onun işləməsini test edə bilərsiniz.

**ifconfig em1 -ARP** - Bu şəbəkə kartında ARP söndürülür.

**ifconfig em1 ARP** - Bu şəbəkə kartında ARP işə salınır.

**ifconfig em1 staticARP** - Əgər ARP aktivləşdirilibsə, host ancaq bu ünvanların müraciətləri üçün cavab verəcək və heç vaxt heç bir müraciət yollamayacaq.

## FTP local servis

File transfer protocol – Əməliyyat sistemimizin daxili imkanı var ki, heç bir paketi yüklemədən müəyyən xidmətlərdən istifadə edə bilək. Bu imkanların içində daxil olan xidmətlərdən biri də FTP serverdir. Bu alt başlıqda FTP serverin qurulması haqqında danışırıq.

FTP serveri aktiv etmək üçün 2 üsul var:

- |   |   |
|---|---|
| 1. <b>ee /etc/rc.conf</b>               | - StartUP faylına sətirləri əlavə edirik.   |
| <b>ftpd_enable="YES"</b>                |   |
| <b>ftpd_program="/usr/libexec/ftpd"</b> | - ftpd daemon yerləşən PATH-ı təyin edirik. |
| <br><b>/etc/rc.d/ftpd start</b>         | - FTP daemon-u işə salırıq.                 |

CLI(Command Line Interface)-dan işə sala bilərik.

**/usr/libexec/ftpd -D** - Susmaya görə 'D' daemon rejimində işə salınır.

**/usr/libexec/ftpd -D -A** - FTP server yalnız 'A' anonymous(anonim) girişə icazə var. Ancaq sistemdə 'ftp' istifadəçi adını 'ftp' şifrəsi ilə yaratmağı unutmayın.

Hər dəfə daemon üzərində müxtəlif opsiyalarla dəyişiklik etdiqdən sonra servisi proseslərdən dayandırıb yenidən işə salmağı unutmayın. Əks halda, köhnə opsiyalar işlək vəziyyətdə qalacaq.

**/usr/libexec/ftpd -D -A -m** - anonymous istifadəciyə fayla yazma və yetki təyin etmə hüququ verilir.

**/usr/libexec/ftpd -D -a 192.168.0.11 -m** - FTP server **192.168.0.11** IP ünvanında 21-ci portda dinləyəcək.

Qulaq aslığı portu dəyişmək istəsək, **/etc/services** faylında aşağıdakı sətrə uyğun olaraq dəyişiklik etməliyik:

**ftp 1505/tcp #File Transfer [Control]**

2. 2-ci üsul işə Super Daemon üzərindən edilir.

**/etc/inetd.conf** - Quraşdırma faylına daxil oluruq və aşağıdakı sətrin qarşısından kommenti silirik ki, aktivləşsin.

```
ftp      stream  tcp      nowait  root      /usr/libexec/ftpd      ftpd -l
```

**Qeyd:** Adı halda FTP istifadəçiləri öz 'ev qovluğu' ünvanından kənara çıxa bilirlər. Bunun üçün **ftp** istifadəçisini yaratdıqda onun shell-ini **/bin/false** etməliyik. Ancaq öncədən "**/etc/shells**" ünvanına **/bin/false** sətri əlavə etməliyik. Bu **ftp** istifadəçiye shell-dən istifadəyə qadağa qoyur.

#### **/etc/ftpusers**

- Bu faylin siyahısında olan istifadəçilərə **ftp** ilə giriş qadağandır. Hər sətirdə bir istifadəçi yazılmalıdır.

#### **/etc/ftpchroot**

- Bu faylin siyahısında olan istifadəçilər isə öz ev qovluğundan başqa ünvana daxil ola bilmirlər. Hər sətirdə bir istifadəçi yazılmalıdır.

#### **/etc/ftpwelcome**

- İstifadəçi daxil olan anda faylin içindəki informasiya çap olunur.

#### **/etc/ftpmotd**

- İstifadəçi daxil olduqdan sonra faylin içindəki informasiya çap olunur.

#### **/etc/ftphosts**

- Əgər serverinizin bir neçə PUBLIC IP ünvanı varsa, FTPD virtualhost məntiqini dəstəkləyir, hansı ki, sizə fərqli Internet IP ünvanında işləyən bir neçə anonim FTP ünvanı təyin etməyə imkan yaradır. Məhz bu faylda həmin virtualhost-lar quraşdırılır. Hər bir virtualhost bir sətirdə olur və aşağıdakı struktura malik olmalıdır:

**hostname** - Virtual host üçün IP, ya da ad.

**user** - Sistem passwd faylinda olan istifadəçi.

**statfile** - Fayldır, hansı ki, bütün fayl ötürülməsi buraya **/var/log/ftpd** jurnallanır.

**welcome** - İstifadəçi daxil olduqda salam mesaj faylı. Susmaya görə **/etc/ftpwelcome** faylidir.

**motd** - İstifadəçi daxil olduqdan sonra ekrana çıxan mesaj. Susmaya görə **/etc/ftpmotd**-dir.

**Qeyd:** Susmaya görə ftp server 'anonymous'-la işləmir və 'anonymous'-un işləməsi üçün 'ftp' adlı istifadəçi tələb edir. Aşağıdakı göstəricilərlə 'ftp' istifadəçi əlavə edirik.

Username: **ftp**

Full name: **FTPD User**

Shell (sh csh tcsh bash rbash nologin false) [sh]: **false**

Use a random password? (yes/no) [no]: **yes**

Və daxil olub yoxlayırıq '**ftp ftp@localhost -> Enter -> Enter**'

**/etc/rc.conf** StartUP quraşdırma faylinə aşağıdakı sətri əlavə edirik ki, sistem yenidənyüklənməsindən sonra işləsin:

**ftpd\_flags="-A"** - Yalnız anonim ftp girişə icazə var.

Eynilə **/etc/inetd.conf** vasitəsilə etmək üçün aşağıdakı sətri əlavə edirik:

**ftp stream tcp nowait root /usr/libexec/ftpd ftpd -l -A**

## DHCP SERVER quraşdırılması

DHCP (Dynamic Host Configuration Protocol) - İstənilən daxili şəbəkədə istifadəçi sayı 20-ni aşarsa, IP idarə etməsi statik olduğu halda, işimiz çox çətinləşir. İşimizin asanlaşdırılması üçün isə DHCP server qurmaq lazımdır. Biz bu başlığımızda isc-dhcp serverin qurulması işini görəcəyik.

**cd /usr/ports/net/isc-dhcp42-server/** - DHCP serverin port ünvanına daxil oluruq.  
**make install clean** - Yükləyirik.

**cd /usr/local/etc** - Quraşdırma faylı yerləşən qovluğa daxil oluruq.

**cp dhcpcd.conf.sample dhcpcd.conf** - Orijinal nüsxə faylından quraşdırma nüsxəsini çıxarıırıq.

**ee /usr/local/etc/dhcpcd.conf** - Əsas quraşdırma faylında aşağıdakı kimi edirik.  
**option domain-name "internal.freebsd";** - Müştərilərə istədiyimiz domain adı təyin edə bilərik. Bizim halda internal.freebsd-dir.

**option domain-name-servers 8.8.8.8;** - Müştərilərə ötürəcəyimiz DNS client IP ünvanları.

**default-lease-time 3600;** - Susmaya görə istifadəçi kirayə müddətini **3600** saniyə istifadə edir.

```

max-lease-time 86400; - Maksimal 84600 saniyə eyni ünvan üçün istifadə olunur.

ddns-update-style none; - Dinamik DNS-i söndürürük.

subnet 10.0.0.0 netmask 255.255.255.0 { - Müşterilərə ötürəcəyimiz IP subnet.

range 10.0.0.129 10.0.0.254; - Müşterilərə ötürəcəyimiz IP aralıq.

option routers 10.0.0.1; - Müşterilər üçün gateway

}

host Behruz { - Bəhruz adlı qeyd etdiyimiz

hardware ethernet 00:0c:29:22:49:f7; - kompüterin MAC ünvanını

fixed-address 10.0.0.27; - IP ünvanına tikirik

}

ee /etc/rc.conf - StartUP faylına sistem yenidənyüklənməsində işləməsi üçün əlavə edirik.

dhcpd_enable="YES" - DHCP serverimiz müraciətləri eml şəbəkə

dhcpd_ifaces="em1" kartında qəbul edir.

/usr/local/etc/rc.d/isc-dhcpd start - DHCP serverimizi işə salırıq.

/var/db/dhcpd/dhcpd.leases - Kirayə verilən IP ünvanlar haqda tam məlumat bu fayla yığılır.

```

DHCPD daemon haqqında jurnalların ayrılmış fayla filter edilməsi üçün ayrıca fayl yaradaq və tələb edilən quraşdırılmayı edək.

```

touch /var/log/dhcp.log - Jurnal faylı yaradırıq.

chmod 644 /var/log/dhcp.log - Yazma yetkisi veririk.

```

**Qeyd:** Syslog və Newsyslog xidmətləri haqqında gələcək başlıqlarımızda geniş müzakirələr edilir.

**/etc/syslog.conf** faylına aşağıdakı sətirləri əlavə edirik (quraşdırırmızda deyirik ki, dhcpd başlığı ilə olan istənilən jurnal sətrini **/var/log/dhcp.log** faylına topla):

```

!dhcpd                               /var/log/dhcp.log

*.*

```

```

/etc/rc.d/syslogd restart - Sonda Syslogd daemonu restart edirik.

```

# Şəbəkə utilitləri

Bu başlıqda biz şəbəkəmizdə olan problemlərin fərqli utilit-lər vasitəsilə təyinatını və həlli üsullarını açıqlayırıq. Həm sistemin daxili imkanlarından, həm də fərqli paketləri yükləyib onların imkanlarından istifadə edirik.

## Netstat

Şəbəkə statusunu göstərmək üçün alətdir. Şəbəkəyə aid olan simvolik strukturu fərqli formatlarda çap edə bilir. Çıxış formatından asılı olaraq, geniş opsiya imkanları aşağıda açıqlanır.

**netstat -w 2 -I em0**

- Əmrli **em0** şəbəkə kartının statistikasını 2 saniyədən bir çap et.

**netstat -w 5 -d**

- **-w 5** saniyə müddətinə sistemə daxil olub-çıxan "packet" və "byte"-ları, **-d**, həmçinin sistem tərəfindən göndərilmeyən paketləri də çap et.

**netstat -p tcp**

- Yalnız aktiv TCP qoşulmalarını çap et.

**netstat -rs**

- Routing statistikasını çap et.

**netstat -s -p tcp**

- TCP qoşulmalar üçün statistikanı detallı şəkildə çap et.

<b>netstat -m</b>	- Kernel tərəfindən istifadə edilən Network Memory Bufer-də olan informasiyanı çap et.
<b>netstat -s</b>	- '-s' TCP,UDP,ICMP paketlərin statistikasını çap et.
<b>netstat -i -p tcp</b>	- '-i' bütün şəbəkə kartlarında '-p' tcp protokoluna aid statistikanı çap et.
<b>netstat -i</b>	- Şəbəkə kartına görə çap et.
<b>netstat -i -f inet</b>	- IPv4 haqda tam məlumat çap et (-n IP ünvanını ada çevirmədən).
<b>netstat -AanW -p tcp</b>	- 'tcp' ilə ESTABLISHED və LISTEN olan bütün statusları çap et.
<b>netstat -I em0 -b</b>	- 'em0' şəbəkə kartında Input/Output paketləri baytlarla çap et.
<b>netstat -inbh</b>	- '-i' şəbəkə kartlarına görə göstər, -n port və IP ünvanları resolve etmədən göstər, -b statistikanı baytlarla göstər, -h oxuna biləcək rahat formatda (yəni KB, MB və GB) olsun.
<b>netstat -a -f inet</b>	- Bütün INET socket-lərini çap et.
<b>netstat -a -f inet6</b>	- Bütün INET6 socket-lərini çap et.
<b>netstat -a -f inet -p tcp</b>	- Bütün INET TCP socket-lərini çap et.
<b>netstat -a -f inet -p udp</b>	- Bütün INET UDP socket-lərini çap et.
<b>netstat -na -f inet</b>	- IPv4-ün qulaq aslığı bütün portları çap et.
<b>netstat -na</b>	- Şəbəkənin statusunda hər bir məlumatı ada çevirmədən çap et.
<b>netstat -rn</b>	- Routing cədvəli ada çevirmədən çap et.
<b>U</b>	- Yazılmış Routing hansı şəbəkə kartından keçirə, o, UP-dir.
<b>G</b>	- Bu default gateway-dir.
<b>H</b>	- Statik marşrut konkret hansısa hosta yazılır. (localhost da ona aiddir.)

## Traceroute və TCPtraceroute

<b>traceroute</b>	- Şəbəkə hostuna gedən paketlərin marşrutunu çap edir.
<b>tcptraceroute</b>	- TCP paketlərdən istifadə edərək şəbəkə hostuna gedən paketlərin marşrutunu çap edir.

Traceroute utilit sayesində şəbəkə yolumuzun harada iləşdiyini təyin edə bilərik. Ancaq nəzərə alın ki, utilit susmaya görə **UDP** ilə işləyir və **UDP**-yə bağlı olan nöqtələri görə bilməyəcəksiniz.

**traceroute mail.ru** - "mail.ru" saytına gedən yolu tapırıq.

**traceroute -n -w 3 -q 1 www.cyberciti.biz** - **www.cyberciti.biz** saytınınadək gedən yolu tam araşdırırıq.

**-n:** Müraciətlərin sürətli getməsi üçün DNS-ə baxış keçirilməsini dayandırırıq.

**-w saniyələrlə:** Göndərilən sınaq paketlərin qayıdış cavabı üçün saniyələrlə təyin edilən gözləmə vaxtı (susmaya görə 5 saniyədir).

**-q RƏQƏM:** Hər hop (şəbəkədə keçid nöqtəsi) üçün sınaq paketin sayı (susmaya görə 3-dür).

**Qeyd:** traceroute əmri ilə biz millisaniyələrə görə şəbəkə sürətinin harada az olduğunu təyin edə bilərik. Susmaya görə "**UDP**" ilə işləyir, həm də Firewall-larda traceroute-u bağlaya bilirlər.

**traceroute -I mail.ru**

- '**I**' ICMP, 'mail.ru' ünvanını ping vasitəsilə traceroute edin. Gördüyünüz kimi, əgər firewallarda **UDP** bağlırsa, **ICMP** ilə də yoxlaya bilərik.

**traceroute -DI mail.ru**

- **mail.ru** saytınınadək olan yolu araşdırıraq.  
**-D:** Göndərdiyimiz ICMP sınaq paketin cavabı qaydan kimi göndərilən və qaydan ICMP paketlər arasında olan fərqləri çap edin. Sütunların yerləşdiyi ünvanları göstərən açar göndərilən paketdə, ardınca hex-də olan orijinal paket və ardınca da nəzərimizdə tutulan 16-liqda olan paket çap edilir. Heç bir dəyişikliyi məruz

qalmayan və haqqında danışılan paketlər  
altdan xətlə (\_) qeyd edilir. Nəzərə alın ki, qeyd  
edilən paketin IP checksum-u və TTL-in uyğunluğu  
gözlənilmir. Susmaya görə bu opsiya ilə hər hop  
üçün yalnız bir **prob** ötürülür.

-**I**: UDP datagramlarının əvəzinə ICMP ECHO-dan  
istifadə edin (Sinonim olaraq "-P icmp").

**traceroute -P TCP mail.ru**

- '-P' spesifik protokolla traceroute edirik  
(**UDP**,**TCP**,**ICMP**,**GRE**), **TCP** ilə 'mail.ru' ünvanına  
traceroute edirik.

**Qeyd:** Susmaya görə traceroute 80-ci porta müraciət edir. Ancaq biz onu '-p' opsiyası ilə  
dəyişə bilərik.

**traceroute -P TCP -p 25 mail.ru**

- '-P' TCP protokolla '-p' 25-ci portla **mail.ru**  
ünvanına '**traceroute**' edirik.

**traceroute -n mail.ru**

- 'mail.ru' ünvanına '**traceroute**' etdikdə  
'-n' ada çevrilməni söndür.

Həmçinin portlardan tcptraceroute paketini yükleyib yoxlaya bilərisiniz:

**cd /usr/ports/net/tcpt traceroute**  
**make install**

- Port ünvanında daxil oluruq.  
- Yükləyirik.

**tcptraceroute -d mail.ru**

- **-d debug** rejimdə **mail.ru** ünvanına olan  
yolumuzu analiz edirik (**man tcptraceroute**).

**tcptraceroute -f 5 pages.ebay.com**

- Çıxan ilk paket üçün **TTL** təyin edirik  
(susmaya görə 1-dir).

## PİNG gündəlik istifadədə

Internet tələb edilən serverimizin şəbəkə quraşdırılmalarını etdikdən sonra internetin olmadığını  
hansısa üsulla təyin etmək lazımdır. Problemin araşdırılması üçün ilk köməyimizə çatan alət ping-dir.  
Problemin təyin edilməsi üçün ilk olaraq serverimizin IP ünvanına, susmaya görə olan yolumuzun IP  
ünvanına və DNS IP ünvanlarına ping paketlər yollayırıq. Aşağıda gündəlikdə biziə lazım olan ping  
imkanları və onun opsiyalarını açıqlayırıq.

<code>ping 0</code>	- Susmaya görə olan gateway-imizə ping atırıq.
<code>ping -a mail.ru</code>	- <b>mail.ru</b> hostunu audit edirik, əgər hosta çatmaq mümkündürse və cavab qayıdırsa, kompüterimizdən xəbərdarlıq səsi eşidəcəyik.
<code>ping -s 500 mail.ru</code>	- Hər bir paketində -s 500 bayt olmaqla mail.ru ünvanına ping yolla.
<code>ping -D -s 1300 -i 0.2 mix.az</code>	- Hosta ping atarkən -D fragment etmə bitini təyin edirik.
<code>ping -S 192.168.121.129 mail.ru</code>	- <b>mail.ru</b> hostuna ping atarkən serverimizdən çıxan paketin -S mənbəyinə təyin ediləcək IP ünvanın <b>192.168.121.129</b> olmasını deyirik.
<code>ping -c 3 10.0.0.1</code>	- <b>10.0.0.1</b> IP-sinə <b>3</b> ədəd ping paket yolla.
<code>ping -c 100 -f mail.ru</code>	- -c <b>100</b> ədəd paketi -f cavabının qayıtmağını gözləmədən, <b>mail.ru</b> ünvanına ping vasitəsilə yolla.
<code>ping -i 0.0005 -c 3 10.0.0.1</code>	- <b>0.0005</b> mikrosaniyə intervalı ilə <b>3</b> ədəd ICMP paketi göstərilən hosta yolla.
<code>ping -c 5 -q mail.ru</code>	- Yalnız ping əmrinin -q statistik məlumatlarını çıxışda çap et.
<code>ping -W 0.001 mail.ru</code>	- <b>mail.ru</b> hostuna gedən paketlərin qayıtmasında gözləmə vaxtını -W millisaniyələrlə təyin et.
<code>ping -R 192.168.1.63</code>	- ECHO_REQUEST necə göndərilməsi və ECHO_REPLY-in necə qayıtmasını qeyd edin və route-u çap et.

## DNS-in yoxlanılması

DNS (Domain Name System) istifadəsi bize istənilən informasiya texnologiyaları mühitində gərəklidir, çünki İP yadda saxlamaq əvəzinə, ad daha rahatdır. Ancaq adlar cavab vermədiyi hallarda, problemin harada olmasını təyin etmək vacibdir. Aşağıdakı misallarda DNS-in fərqli yazırlara görə təyin edilməsini açıqlayırıq.

**Qeyd:** Unutmayın ki, FreeBSD son versiyalarında susmaya görə BIND saxlamır və dig əmri BIND paketin komponenti olduğuna görə sistemdə mövcud olmur. Siz onu `/usr/ports/dns/bind-tools/` port ünvanından, ya da `pkg install bind-tools` əmri ilə paketlərdən yükleyə bilərsiniz.

<code>dig www.mail.ru @8.8.8.8</code>	- 'www.mail.ru' adını '8.8.8.8' IP ünvanlı server-də axtar.
<code>dig mail.ru mx</code>	- 'mail.ru' DNS adının MX yazısını axtar.
<code>dig mail.ru ns</code>	- 'mail.ru' DNS adının NS yazılarını axtar.
<code>dig +trace mail.ru</code>	- 'mail.ru' DNS A yazısını rekursiv olaraq bütün DNS serverlərdə axtar.
<code>dig +short mail.ru</code>	- 'mail.ru' DNS A yazısının yalnız qısa şəkildə IP ünvanlarını çap et.
<code>dig -x 8.8.8.8</code>	- '8.8.8.8' IP ünvanının PTR yazısını çap et.
<code>host 8.8.8.8</code>	- '8.8.8.8' IP ünvanına əsasən adı çap et.
<code>host -l sayt.az</code>	- 'sayt.az' saytına aid olan bütün NS, PTR, A yazılarını AXFR (Zone transfer) istifadə edərək çap et.
<code>hostname</code>	- Serverimizin tam adını çap et.
<code>hostname -s</code>	- Serverin 's'(short) qısa adını çap et.
<code>hostname server1</code>	- Serverimizin adını 'server1' təyin et.

```
host -a az
```

- 'az' domain-in qeydiyyatda olduğu bütün ünvanları çap et (-a – Tipi ANY olan ətraflı məlumat verəcək müraciət formasıdır).

```
whois -I az
```

- 'az' domain-i haqda informasiyani IANA bazasında axtarış edir.

## ARPİNG – təyin edilmiş IP və ya arp pingləri ötürür

**Qeyd:** Əgər şəbəkədə istifadəçi öz Desktop-unda Firewall-u açıq saxlayıbsa, onda həmin istifadəçinin IP-sinə ping getməyəcək və biz dəqiq bilməyəcəyik ki, bu IP istifadə olunur, ya yox. Bunun üçün 'arping' adlı paket var. 'arping' IP adresə MAC-la ping edir (**man arping**). Yəni işimizi rahatlaşdırır.

```
pkg_add -r arping
```

- FreeBSD8.4 üçün 'arping' paketini yükleyirik.

```
pkg install arping
```

- FreeBSD9.3,10.1 üçün 'arping' paketini yükleyirik.

```
arping 192.168.192.159
```

- '192.168.192.159' IP ünvanına MAC-la ping edirik.

```
arping -I em1 192.168.192.159
```

- '-I' şəbəkə kartı em1-lə, 192.168.192.159 IP ünvanına arping edirik.

```
arping -c 1 192.168.192.159
```

- '-c' say (count), yalnız 1 dəfə 192.168.192.159 IP ünvanına arping edirik.



# BÖLÜM 7

## Dump-Restore, OpenSSH, RCS, TempFS-MemFS, Snapshot-lar, FSCK, Quota

- / FreeBSD Dump (Rezerv nüsxə) və Restore (Bərpa edilmə)
- / OpenSSH
- / Revision Control System, TempFS və MemFS
- / Snapshots, File System Checking və Disk Quota
- / Syslogd, Syslog-NG və Newsyslog

Başlığımız işlək vəziyyətdə olan bir FreeBSD serverin digər işlək vəziyyətdə olan boş serverə nüsxələnməsini açıqlayır. Uzaqdan şifrələnmiş kanalla qoşulma üçün istifadə edilən SSH adlı daemonun spesifik xüsusiyyətlərini, həm UNIX, həm də Windows məşinlər üçün sertifikatla girişini və SSH diskin istifadə edilməsini izah edirik. Fərqli istifadəçilər üçün fayllarda olan dəyişikliklərin idarəedilməsi qaydaları, fayl sistemin RAM-də saxlanılması metodikası, bərpa oluna biləcək nüsxələrin götürülməsi, fayl sistemin yoxlanılması və diskə məhdudiyət də bu başlıqda göstərilir. Fərqli tip avadanlıqlardan gələn jurnalların bir serverdə yiğilib analiz edilməsi üçün Syslog və Syslog-NG daemonlarının qurulması və daxili daemonla jurnalların planlaşmasını edəcəyik.

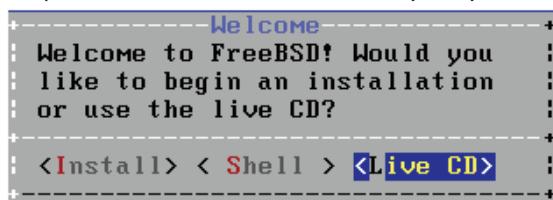
# FreeBSD Dump (Rezerv nüsxə) və Restore (Bərpa edilmə)

Dump əmri blok tipli alətlərlə və inode ağac strukturu ilə işləyir. 10 səviyyəli arxiv yaratmaq imkanına malikdir. Tam arxiv və incremental (əlavələr olan) rezerv nüsxə çıxarmaq imkanına sahibdir. İşlek vəziyyətdə olan sistemin rezerv nüsxəsini çıxara bilir (Ancaq öncədən snapshot çıxarılır). Data-nı sıxa və təyin edilmiş uzunluğşa görə hissələrə bölə bilir.

**Qeyd:** Başlıqda şəbəkədən istifadə edildiyinə görə nəzərə alın ki, şəbəkə quraşdırılması 6-ci başlıqda geniş açıqlanır.

DUMP edilən server IP: **192.168.1.10**  
Restore edilən Server IP: **192.168.1.50**

Bərpa ediləcək server-i liveCD-dən yükleyirik



**root** istifadəçi adı ilə daxil oluruz və aşağıdakı ardıcılıqla işimizi görürük.

```
umount -f /tmp  
mdmfs -M -s256m md1 /tmp
```

- MD driver-dən istifadə edərək RAM daxilində fayl sistemini quraşdırın və mount edin.
- **M** malloc (Memory Allocation) istifadə edin SWAP əvəzinə, yəni RAM-da saxlayın.
- **s** həcmi **256** Megabayt olan **md1** diskinə mount edin.

Bərpa ediləcək server-də SSH-i qaldırıq. İlk olaraq, '**/etc**' qovluğuna yazma yetkimiz olmalıdır.

```
mkdir /tmp/etc
```

- '**/tmp**' qovluğun altında '**etc**' qovluğu yaradaq ki, ora mount edə bilək.

```
mount_unionfs /tmp/etc /etc
```

- **UNION** (birgə) fayl sistem yaradaq, yəni iki mövcud olan qovluğu bir-birinə mount edirik.

```
ee /etc/ssh/sshd_config
```

- SSH qlobal quraşdırma faylında ROOT adlı istifadəçi ilə girişə izin veririk.

```
PermitRootLogin yes
```

- **root** istifadəçi adı ilə giriş imkanını yaradırıq.

```
/etc/rc.d/sshd onestart  
passwd
```

- SSHD daemon-u bir dəfə üçün işə salırıq.

- root istifadəçisinə şifrə təyin edirik ki, DUMP edilən serverdən bura qoşula bilək.

```
ifconfig em0 192.168.1.50 netmask 255.255.255.0
```

- Bərpa ediləcək serverə IP ünvani quraşdırırıq.

```
root@:~ # gpart create -s gpt da0  
  
da0 created
```

- GPT disk bölgü "-s" sxemini **da0** diski üçün təyin edirik.

```
root@:~ # gpart add -t freebsd-boot -l gpboot -b 40 -s 512K da0  
  
da0p1 added
```

- **512K** olan **boot** hissəsini **da0** diskinə **gpboot** adı ilə təyin edirik.

```
root@:~ # gpart bootcode -b /boot/pmbz -p /boot/gptboot -i 1 da0
          - GPT boot kodunu boot hissəsinə yükleyirik.
```

**bootcode written to da0**

```
root@:~ # gpart add -t freebsd-ufs -l gprootfs -b 1M -s 19G -i 2 da0
          - / fayl sistem üçün 19GB həcmində olan 2-ci
          indeksli disk yaradırıq. Disk üzerinde sektorların
          düzgün bölündürülməsi üçün o, 1M sərhədində
          başlamalıdır.
```

**da0p2 added**

```
root@:~ # gpart add -t freebsd-swap -l gpswap -s 512M da0
          - gpswap adlı və 512MB həcmli swap disk hissəsi
          yaradırıq.
```

**da0p3 added**

Hansısa səhv etdiyiniz hallarda aşağıdakı əmrlərlə disk hissəsini və ya bütöv diskin özünü silə bilərsiniz:

```
gpart delete -i 1 da0
          - Diskin 1-ci hissəsini silir.
da0p1 deleted
```

```
gpart destroy da0
          - Bütöv diskini silir.
da0 destroyed
```

```
gpart show
          - Sonda diskimizin strukturu aşağıdakı kimi olacaq:
=>      34  41942973  da0  GPT  (20G)
        34          6      - free -  (3.0K)
        40        1024    1  freebsd-boot  (512K)
      1064         984      - free -  (492K)
     2048  39845888    2  freebsd-ufs  (19G)
  39847936   1048576    3  freebsd-swap  (512M)
 40896512   1046495      - free -  (511M)
```

```
root@:/ # newfs /dev/da0p2
          - Bərpa ediləcək diskimizə UFS fayl sistem yazırıq.
```

```
root@:/ # mount /dev/da0p2 /mnt/
          - Bərpa ediləcək diskimizi /mnt/ ünvanına
          mount edirik.
```

```
root@/: # rm -rf /mnt/.snap
```

- Faylı silirik ki, dump edəndə mövcudluğu haqda xəbərdarlıq çıxmasın.

Bu addımdan sonra sadəcə DUMP-un bitməsini gözləyirik. Dump bitən kimi isə, serverimizə yenidənyüklənmə edib, İP ünvanı dəyişirik və işlərimizə davam edirik.

#### DUMP çıxarılan server

Əgər siz serveri işlək halda DUMP etsəniz, o, sizə imkan verməyəcək, çünki serverdə fayl sistemi **JURNALLAŞMA** metodu ilə format edilib. **JURNALLAŞMA** fayl sistemi zədələndiyi halda özünə yiğdiyi informasiya sayəsində(yəni META verilənlər) '**fsck**' əmri ilə zədələnmiş faylları sizə geri qaytara bilər. Ancaq buna baxmayaraq, siz yenə də həmin jurnallaşmanı söndürüb işinizə davam edə bilərsiniz. Yalnız DUMP bitdikdən sonra həmin serverdən jurnallaşmanı işə salmağı unutmayın.

Beləliklə, DUMP əmrini daxil etdikdə çıxan problemimiz aşağıdakı kimi olacaq.

```
root@node1:/root # dump -0 -u -L -f - / | ssh -l root 192.168.1.50 "cd /mnt ; restore -rf -"
```

```
mksnap_ffs: Cannot create snapshot //snap/dump_snapshot: /: Snapshots are  
not yet supported when running with journaled soft updates: Operation not  
supported
```

```
dump: Cannot create //snap/dump_snapshot: No such file or directory
```

**Password:**

```
cannot open /dev/tty: Device not configured
```

```
Tape is not a dump tape
```

İndi isə problemi həll edək. DUMP ediləcək serverimizdə Single USER rejiminə keçid edirik.

```
shutdown now
```

- SINGLE USER rejiminə keçid edirik.

```
cd /
```

- Kök qovluğa keçid edirik.

```
ls -lo .sujournal
```

- JURNALLARI özündə saxlayan faylimizin  
üstündə olan flag-ları gözdən keçiririk.

```
-r----- 1 rootwheel schg,sunlnk,nodump,opaque 33554432 Mar 10 00:05 .sujournal
```

Biz bu faylı ya qovluqdan silməliyik, ya da yerini dəyişməliyik. Ona görə də öncə Jurnallaşmanı söndürməliyik, Jurnalları silməliyik. Sonra flag-ları silməliyik.

<b>tunefs -J disable /dev/da0p2</b>	- <b>/dev/da0p2</b> diskində ikinci partition-un GEOM jurnallaşmasını söndürürük.
<b>tunefs -n disable /dev/da0p2</b>	- <b>/dev/da0p2</b> diskində ikinci partition-un program yenilənməsini söndürürük.
<b>tunefs -j disable /dev/da0p2</b>	- <b>/dev/da0p2</b> diskində ikinci partition-un program yenilənmə jurnallaşmasını söndürürük.

Sonra coxistifadəçili rejimə (yəni adı yüklənmə) keçid alıb, flag-ları dəyişirik.

<b>chflags noschg,nosunlnk,dump /.sujournal</b>	- <b>.sujournal</b> faylıının flag-larında sistemə hissiyyatlılığı, silinə bilməyəcəyini söndürürük və dump edilə biləcək edirik.
<b>mv /.sujournal /root</b>	- <b>.sujournal</b> faylıının ünvanını dəyişirik.

İndi isə DUMP ediləcək serverdən DUMP əmrini işə salırıq və həmin axının nəticəsini kanalla SSH ilə **192.168.1.50** IP ünvanlı bərpa ediləcək serverin '**restore**' əmrinə yollayıraq.

```
root@node1:/# dump -0 -u -L -f - / | ssh -l root 192.168.1.50 "cd /mnt ; restore -rf -"
```

- 0** (sıfır) - DUMP səviyyələri mövcuddur və **0-9** aralığında olur. Sıfır **FULL BACKUP** (tam) deməkdir.
- u** - Xidməti mesajları **/etc/dumpdates** faylına yazacaqıq.
- L** - Sistem işlək vəziyyətdə olduğuna görə **SNAPSHOT**-dan istifadə edin.
- f** - Çıxışı **STDOUT**-a yollayın.

Bərpa olunan tərəfdə:

- r** - Fayl sistemini yenidən qurun.
- f** - Çıxışı **STDOUT**-a yönəldirin.

Öncə daxil edilən əmrin çıxışında uğurlu nəticəni aşağıdakı kimi alacaqsınız:

```
Password for root@: DUMP: Date of this level 0 dump: Sun Mar 22 15:38:56 2015
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping snapshot of /dev/da0p2 (/) to standard output
DUMP: mapping [Pass I] [regular files]
```

```

DUMP: mapping (Pass II) [directories]
DUMP: estimated 7756335 tape blocks.

cannot open /dev/tty: Device not configured
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
expected next file 10, got 9
DUMP: 12.47% done, finished in 0:35 at Sun Mar 22 16:19:02 2015
DUMP: 30.49% done, finished in 0:22 at Sun Mar 22 16:11:43 2015
DUMP: 43.31% done, finished in 0:19 at Sun Mar 22 16:13:34 2015
DUMP: 61.54% done, finished in 0:12 at Sun Mar 22 16:11:25 2015
DUMP: 77.19% done, finished in 0:07 at Sun Mar 22 16:11:19 2015
DUMP: 88.96% done, finished in 0:03 at Sun Mar 22 16:12:39 2015
DUMP: DUMP: 7774840 tape blocks
DUMP: finished in 2062 seconds, throughput 3770 KBytes/sec
DUMP: DUMP IS DONE

```

Nəticədə, **/etc/dumpdates** faylin tərkibi aşağıdakı formada olacaq:

<b>/dev/da0p2</b>	<b>1 Mon Jun 15 16:10:08 2015</b>
<b>/dev/da0p2</b>	<b>0 Sun Mar 22 18:02:53 2015</b>

#### DUMP – Bərpaya yararlı əmrlər

**dump -auL /usr**

- "/usr" slice-nı "/dev/sd0" tape-nə dump edəcək. (-**aauto-size**, -**L** hal-hazırda işləyən fayl sistemi **dump** edir və işə başlamazdan once onun **snapshot-u .snap** qovluğuna çıxarılmalıdır, -**u** spesifik vaxt möhürü ilə dump ediləcək fayl haqqında məlumatı **/etc/dumpdates** faylinə yazır).

**dump -0auL -f - /dev/da0s1d | bzip2 | ssh backup@192.168.1.50 -p 22 dd of=/backup/vds-admin.dump**

| - çıxışı **bzip2** alqoritminə ötürür ki, sıxılmış vəziyyətdə olsun və öz çıxışını SSH-a ötürəndə daha az həcm tutsun.

-**f** rezerv nüsxəni fayla yazın. (Bizim halda fayl SSH vasitəsilə dd-yə ötürülür və o da öz növbəsində **/backup/vds-admin.dump** faylinə yazır)

**Qeyd:** İnkremental (yalnız dəyişiklik olan hissə) arxivin çıxarılması üçün opsiyalarda dump-un səviyyəsini 1-dən 9-dək aralıqda təyin etməlisiniz.

Bu halda dump son edilən rezerv nüsxə haqqında verilənləri /etc/dumpdates faylından istifadə edəcək.

<code>dump -f /dev/nsa0 -auL /</code>	- '/' kök slice-in rezerv nüsxəsini götürür.
<code>dump -f /dev/nsa0 -auL /var</code>	- '/var' slice-in rezerv nüsxəsini götürür.
<code>dump -f /dev/nsa0 -auL /tmp</code>	- Və '/tmp' slice-in rezerv nüsxəsini götürür.
 <code>restore -t</code>	- /dev/sd0 tape-dən bərpa siyahısını çap edəcək.
 <code>restore -t home/mwlucas/.cshrc</code>	- Arxivin seçdiyimiz ünvanında ".cshrc" faylini axtarış edirik.

**Qeyd:** Əgər biz "/usr" slice-ni rezerv nüsxə etmişiksə, dump ilk slice-i görür, ona görə də "/home"-dan başlamaq lazımdır.

Bərpa ediləcək serverdə liveCD vasitəsi ilə daxil olub, diskimizə fayl sistem yazmalıyıq ki, sonra hansısa vasitə ilə Live serverə yüklediyimiz rezerv nüsxəni bərpa edə bilək. Bunu aşağıdakı addımlarla icra edirik:

<code>newfs /dev/adls1g</code>	- Yeni disk yaradırıq.
<code>mount /dev/adls1g /mnt</code>	- Onu slice-a mount edirik "/mnt".
<code>cd /mnt</code>	- /mnt slice-na daxil olub,
<code>restore -r</code>	- Bərpa işinə başlayırıq.
 <code>restore -i</code>	- Bərpa edilmənin interaktiv rejimidir (menyunu isə idarə etmək olur).
 <code>restore&gt;cd home/mwlucas</code>	- Ünvana daxil olub,
<code>restore&gt;add ssh.tar</code>	- lazımı faylları
<code>restore&gt;add .cshrc</code>	- əlavə edirik,
<code>restore&gt;extract</code>	- açırıq.

# OpenSSH

OpenSSH version1 **2** tip açardan istifadə edir **3DES/Blowfish**

OpenSSH version2 **5** tip açardan istifadə edir **3DES/Blowfish/AES/Arcfour/CAST128**

**/etc/ssh/sshd\_config**

**/usr/bin/ssh-keygen**

**/usr/bin/ssh**

**/root/.ssh**

- SSHD server üçün qlobal quraşdırma faylıdır

- Unikal **public/private** açar cütlüyünü generasiya edir. (**RSA1,RSA,DSA**)

- **ssh client** utilit bu ünvanda yerləşir.

- root adlı istifadəçi üçün **ssh** client-lərin bütün **Public/Private** açarları və qoşulmuş ünvanların FingerPrint-ləri bu ünvanda yiğilir. Ümumiyyətlə, hər bir istifadəçi üçün ssh client istifadə etdiyi ilk andan etibarən ev qovluğunda **.ssh** adlı qovluq yaranacaq və istifadəçi tərəfindən generasiya edilmiş bütün açarlar bu qovluqda yiğılacaq.

**'.ssh' qovluğunu açıqlayaq.**

**known\_hosts**

**identity**

**id\_rsa**

- Avtorizasiyadan keçmiş kompüterlərin siyahısını təşkil edir.

- SSH version 1-dən gəlmış Private Açar.

- SSH version 2-dən gəlmış **RSA** tipli Private Açar.

<b>id_dsa</b>	- SSH version 2-dən gəlmiş <b>DSA</b> tipli Private Açıar.
<b>identity.pub, id_dsa.pub, id_rsa.pub</b>	- Öncə yazdığınız SSHv1 IDENTITY və SSHv2 RSA, DSA tipli Private Açıarlar üçün PUBLIC açarlardır. - Faylda Remote istifadəçilərin Public Açıarları olur.
<b>authorized_keys</b>	
<u>SSH Daemon quraşdırma faylımızın vacib məqamlarını açıqlayaq:</u>	
<b>/etc/ssh/sshd_config</b>	- Qlobal quraşdırma faylı bu ünvanda yerləşir.
<b>Permit RootLogin no</b>	- Bu sətr SSH serverimizə birbaşa root istifadəçi adı ilə qoşulmağa icazə verir ( <b>yes</b> ), qadağan qoyur ( <b>no</b> ).
<b>AllowGroups sshusers</b>	- Serverimizə SSH vasitəsi ilə yalnız serverimizdə olan " <b>sshusers</b> " adlı qrupun üzvləri qoşula bilərlər.
<b>DenyGroups sshusers</b>	- Serverimizdə olan " <b>sshusers</b> " adlı qrupun üzvləri serverimizə SSH vasitəsilə qoşulma edə bilməzlər.
<b>AllowUsers root unix@192.168.5.10</b>	- Sistemə yalnız <b>root</b> adlı istifadəçi istənilən IP ünvandan və <b>unix</b> adlı istifadəçi ancaq <b>192.168.5.10</b> IP ünvandan daxil ola bilər.
<b>DenyUsers blocked blocked@3.3.3.3</b>	- Bu sətrin qarşısında olan " <b>blocked</b> " adlı istifadəçi adı, ya da <b>3.3.3.3</b> IP ünvandan qoşulmağa çalışacaq, " <b>blocked</b> " adlı istifadəçi üçün giriş qadağandır, girişə icazə yoxdur.
<b>Port 22</b>	- Susmaya görə SSH server <b>22</b> -ci porta qulaq asır. (Dəyişməniz daha düzgün olar.)

Command Line Interface-dən fərqli quraşdirmalarla aşağıdakı əmrlər ilə işə sala bilərik.

**/usr/sbin/sshd -f /root/my\_cfg** - SSH serverimiz işə düşdükdə qlobal quraşdırma

	faylı olaraq ' <b>/root/my_cfg</b> ' faylinin oxunmasını təyin edirik.
<b>/usr/sbin/sshd -g 60</b>	- SSH serverimizə istifadəçilərin daxil ola bilmə vaxtını <b>60-120</b> saniyə aralığında təyin edirik.
<b>/usr/sbin/sshd -p 53943</b>	- SSH serverin qulaq asacağı portun rəqəmini <b>53943</b> edirik.
<b>/etc/rc.d/sshd keygen</b>	- SSH serverimizi işə salmazdan önce açar cütlüyü generasiya edirik.
<b>ssh -p 2222 root@10.0.0.2</b>	- <b>10.0.0.2</b> IP ünvanlı serverə <b>2222</b> -ci portla SSH-la qoşuluruq.

**Qeyd:** Siz SSH serverin portunu istifadəçilər serverinə qoşulmuş vəziyyətdə dəyişmək istəsəniz, narahat olmayın, çünkü SSH daemon-u restart etsəniz də, köhnə qoşulmuş istifadəçilərin tranzaksiyaları bitməyənədək toxunmayıcaq.

<b>ssh -1 -p 2222 root@localhost</b>	- SSH version <b>1</b> ilə <b>2222</b> -ci portla root istifadəçi adı ilə localhosta qoşuluruq.
<b>ssh -2 -p 2222 root@localhost</b>	- SSH version <b>2</b> ilə <b>2222</b> -ci portla root istifadəçi adı ilə localhosta qoşuluruq.

### SSH Client

<b>SSH</b> qoşulma uzaqdan qoşulan kimi tty avtomatik pts terminalını yaradır.	
<b>tty</b>	- Əmrin nəticəsini ilk seans olaraq bize <b>/dev/pts/0</b> çap edəcək.
<b>ssh root@10.0.0.2</b>	- 10.0.0.2 IP ünvanına root istifadəçi adı ilə SSH-la qoşuluruq.
<b>ssh -l root 10.0.0.2</b>	- 10.0.0.2 IP ünvanına root istifadəçi adı ilə SSH-la qoşuluruq.

```
ssh -b 10.0.0.3 root@10.0.0.7
```

- Serverimizdə seçdiyimiz IP ünvanı 10.0.0.3-dən 10.0.0.7 IP ünvanına root istifadəçi adı ilə qoşulur.

```
ssh root@10.0.0.2 uname -a
```

- root istifadəçi adı ilə 10.0.0.2 IP ünvanına "uname -a" əmrini yollayırıq.

```
ssh -l root 10.0.0.3 "uname -a"
```

- Öncəki sətirlə eyni işi görür.

SSH 3 debug (Problemin araşdırılması) rejimində çalışır.

```
ssh -v root@10.0.0.5
```

- Debug level1

```
ssh -vv root@10.0.0.5
```

- Debug level1,level2

```
ssh -vvv root@10.0.0.5
```

- Debug level1,level2,level3

### Secure Copy (Təhlükəsiz nüsxələmə)

```
jot 10000000 > test.exe
```

- Test üçün öz serverimizdə bir fayl yaradırıq.

```
scp -p testfile cavid@192.168.0.105:
```

- 'testfile' adlı faylı **cavid** adlı istifadəçi ilə **192.168.0.105** IP ünvanlı serverə '-p' eyni vaxt möhürü ilə nüsxələyirik.

```
scp -P 10002 testfile cavid@192.168.0.105:
```

- 'testfile' adlı faylı **cavid** istifadəçi adı ilə **192.168.0.105** IP ünvanlı serverin '-P' seçilmiş **10002**-ci portuna qoşulub yükləyirik.

```
scp -r testfolder cavid@192.168.0.104:globalfolder
```

- '-r' testfolder adlı qovluğu yerləşdiyimiz serverdən cavid istifadəçi adı ilə **192.168.0.105** IP ünvanlı serverin globalfolder adlı qovluğununa nüsxələyirik.

```
scp test.exe root@10.0.0.2:
```

- scp əmri root istifadəçi adı ilə **10.0.0.2** IP ünvanlı SSH serverə yerləşdiyimiz serverdə olan **test.exe** faylini nüsxəleyəcək.

```
scp /usr/test.exe root@10.0.0.2:/var
```

- scp /usr ünvanından test.exe faylinı **root** istifadəçi adı ilə **10.0.0.2** IP ünvanlı serverin **/var** qovluğuna nüsxələyəcək (Nəzərə alın ki, qoşulduğumuz istifadəçi adının **/var** qovluğuna yetkisi olmalıdır).

```
scp test.exe root@10.0.0.2:new
```

- test.exe faylinı root istifadəçi adı ilə 10.0.0.2 IP ünvanlı serverin **new** adlı faylina nüsxələyəcək.

```
scp root@10.0.0.5:/usr/home/test.exe .
```

- root istifadəçi adı ilə 10.0.0.5 IP ünvanlı serverin **/usr/home** ünvanında olan **test.txt** faylini hal-hazırda olduğumuz serverimizə nüsxələyirik.

```
scp root@10.0.0.5:test.exe /usr/home
```

- 10.0.0.5 IP ünvanlı serverin root adlı istifadəçinin ev qovluğundan **test.exe** faylini olduğumuz serverin **/usr/home** ünvanına nüsxələyir.

```
scp -o BindAddress=10.0.0.5 root@10.0.0.2:test.exe .
```

- Yerləşdiyimiz serverin daxili 10.0.0.5 IP ünvanı istifadə edilməklə, root istifadəçi adı ilə 10.0.0.2 IP ünvanlı serverdən **text.exe** faylini yerləşdiyimiz serverə nüsxələyirik.

```
scp -v
```

- Nüsxələnmə vaxtı problem çıxarsa, debug rejimdən istifadə edirik.

```
scp -l 56 -o BindAddress=10.0.0.5 root@10.0.0.2:test.exe .
```

- scp **56 kbit** sürət ilə yerləşdiyi server IP ünvanı 10.0.0.5-lə, root istifadəçi adı ilə **test.exe** faylini 10.0.0.2 IP ünvanlı serverdən öz yerləşdiyi məkana nüsxələyəcək.

### SecureFTP – SFTP (Təhlükəsiz fayl köçürülmə protokolu)

SFTP eynilə FTP server kimi işləyir və FTP-də istifadə olunan əmrlər da istifadə edilir. Sadəcə burada istifadəçi ilə server arasında olan yol burada şifrələnir.

```
sftp 10.0.0.2
```

- sftp bu halda **10.0.0.2** IP ünvanlı serverə öz daxili sisteminə qoşulduğu istifadəçi adı ilə qoşulacaq (unutmayın, su əmri ilə root səviyyəsinə qalxsanız da, son istifadəçi root sayılır.)

```
sftp -o BindAddress=10.0.0.3 root@10.0.0.5
```

- SFTP ilə serverimizdə seçdiyimiz IP ünvanı **10.0.0.3** ilə root istifadəçi adı ilə **10.0.0.5** serverinə qoşuluruq.

```
help
```

- SFTP serverimizə qoşulduğdan sonra bu əmr ilə daxildə işləyən bütün əmrləri siyahılaya bilərik. ftp serverdə işləyən sintaksislərin demək olar ki, hamısı burada da işləyir.

```
progress
```

- Əmri işə saldıqdan sonra serverdə bütün görülən işlər progressiv rejimdə çalışacaq.

```
lcd /usr
```

- Mənbə serverimizdən götürdüyümüz daxili qovluğu **/usr** təyin edirik. Bu qovluq faylların transfer edilməsində istifadə olunacaq. Yəni qoşulduğumuz mənsəb serverdən məlumatların köçürülməsi ünvanını mənbə serveri **/usr** qovluğuna təyin edirik.

```
lpwd
```

```
put test.exe
```

- Bu əmr **lcd** əmri ilə təyin edilmiş qovluğu çap edir.  
- test.exe faylini yerləşdiyimiz mövcud qovluğumuzdan qoşulduğumuz serverə nüsxələyir.  
(Əgər qoşulduğumuz serverdə yetki varsa.)

```
get test.exe
```

- test.exe faylini qoşulduğumuz serverdən yerləşdiyimiz mövcud qovluğa nüsxələyir.

```
sftp -b script_name root@10.0.0.5
```

- sftp **root** istifadəçi ilə **10.0.0.5** IP ünvanlı serverə qoşduqda yerinə yetiriləcək işləri  
- **b** (yerinə yetiriləcək skripti oxuyur)  
**script\_name** faylından oxuyacaq.

**Qeyd:** Unutmayın ki, bu rejim yalnız **SSH token authentication** olduğu halda işləyir.  
Yəni interaktiv rejimdə işləməyəcək.

```
script_name  
lcd /var  
cd /var/tmp  
mput test.txt  
exit
```

- Skriptimizin məzmunu isə aşağıdakı kimi olacaq:

#### SSH-KEYGEN (SSH açarlarının generasiya edilməsi)

```
ssh-keygen -t rsa|rsa|dsa
```

- ssh-keygen uyğun olaraq **rsa1**, **rsa** və **dsa** açar cütlüyünü generasiya edə bilir.

```
ssh-keygen -t dsa
```

- ssh-keygen **dsa** tipli açar cütlüyünü generasiya edir.

```
ssh-keygen -v -t dsa -C Comment
```

- Əmr **dsa** tipli açıq və gizli açarı yaradanda **-v** (detallı açıqlama) verbose edir və "**-C**" opsiyası ilə şərh yazmaq olur.

```
ssh-keygen -t dsa -C unix@unix.az
```

- "**unix@unix.az**" şərhi ilə **dsa** tipli açar cütlüyü yaradırıq (yalnız unutmayaq ki, sistem istifadəçisi üçün açar cütlüyü yaratıldıqda gərək həmin istifadəçi adı ilə sistemə **1** dəfə də olsa, daxil olaq ki, **known\_hosts** faylı açıq açar yazılsın.)

```
ssh-keygen -t rsa -C unix1.unix.internal
```

- Öncəki sətirlə eyni işi görür. Sadəcə açar tipi burada RSA-dir (susmaya görə açarlar **/usr/home/`whoami`/.ssh** ünvanında olur).

```
ssh-keygen -l -f /root/.ssh/id_rsa
```

- Bu əmr **/root/.ssh/id\_rsa** gizli açarının istifadə etdiyi public açarını çağırır.

```
2048 fb:4f:be:71:f4:c3:be:12:ce:bb:a2:17:9e:10:18:25 root@freebsd10.1 (RSA)
```

```
ssh-keygen -l -f /root/.ssh/id_dsa
```

- Bu əmr **/root/.ssh/id\_dsa** gizli açarının public açarını çağırır.

```
1024 1d:a5:09:c6:cb:48:1e:41:88:3b:7a:da:ac:40:a3:9f root@freebsd10.1 (DSA)
```

```
cat /root/.ssh/id_rsa.pub
```

- Faylin içində baxsaq, "**-C**" opsiyası ilə təyin etdiyimiz şərhi görə bilərik.

**Qeyd:** Susmaya görə istenilən tip açarlar `~/.ssh/` qovluğunda yığılır.

**Qeyd:** DSA susmaya görə 1024 bit ilə şifrələnir.

**Qeyd:** RSA susmaya görə 2048 bit ilə şifrələnir.

### PKI Logins – Açar vasitəsilə qeydiyyatdan keçmə

Əgər serverlarınızın şifrələrlə yox, açarlarla qeydiyyatdan keçməsini istəyirsizsə, o serverlərin hər birində mütləq dsa və ya rsa tipli açar cütlüyü yaratmaq lazımdır.

```
ssh-keygen -t dsa -C root@localhost      - yaranacaq (id_rsa.pub və id_rsa)  
ssh-keygen -t rsa -C root@localhost      - yaranacaq (id_dsa.pub və id_dsa)
```

Öz serverimizdən **10.0.0.5** IP ünvanlı serverə açar vasitəsilə daxil olmağa çalışırıq. Öncə hər iki serverdə RSA tipli açar generasiya edək. Sonra isə hər bir serverin özünə aid olan PUBLIC açarı digər serverin `~/.ssh/` qovluğuna **serevrname.id\_rsa.pub** adı ilə nüsxələyək. Sonra isə qarşı serverdən gələn PUBLIC açarı hər serverin `~/.ssh/` qovluğunda **authorized\_keys** faylinə nüsxələmək lazımdır. Nəticədə, hər iki server bir-birinə şifrəsiz qoşulmalıdır.

Serverlərimizin IP ünvanları, deyək ki, aşağıdakı kimidir:

<b>10.0.0.5</b>	- Mənbə serverin IP ünvanı
<b>10.0.0.2</b>	- Mənsəb serverin IP ünvanı

```
ssh-keygen -t rsa                                - Əmrələ hər iki serverdə RSA tipli açar  
                                                 generasiya edirik.
```

```
scp /root/.ssh/id_rsa.pub root@10.0.0.2:.ssh/10.0.0.5.root.rsa.pub  
                                             - Mənbə serverimizdə root istifadəçi üçün  
                                             yaratdığımız id_rsa.pub açarını root istifadəçi  
                                             adı ilə 10.0.0.2 serverinə /root/.ssh/  
                                             10.0.0.5.root.rsa.pub ünvanına nüsxələyirik.
```

```
cat /root/.ssh/10.0.0.5.root.rsa.pub > /root/.ssh/authorized_keys  
                                             - Daha sonra mənsəb serverimizdə 10.0.0.5 IP  
                                             ünvanlı serverimizə aid olan PUBLIC açarını  
                                             "authorized_keys" faylinə nüsxələyirik.
```

İndi isə mənsəb serverimizdən generasiya edilmiş RSA PUBLIC açarı mənbə serverimizə nüsxələyək və sonra həmin açarı **authorized\_keys** faylına köçürək.

```
scp /root/.ssh/id_rsa.pub root@10.0.0.5:.ssh/10.0.0.2.root.rsa.pub
```

- Mənsəb serverimizdə root istifadəçi üçün yaratdığımız **id\_rsa.pub** açarını root istifadəçi adı ilə **10.0.0.5** serverinə **/root/.ssh/10.0.0.2.root.rsa.pub** ünvanına nüsxələyirik.

```
cat /root/.ssh/10.0.0.2.root.rsa.pub > /root/.ssh/authorized_keys
```

- Ardınca mənbə serverimizdə **10.0.0.2** IP ünvanlı serverimizə aid olan PUBLIC açarı "**authorized\_keys**" faylına nüsxələyirik.

Sonda isə root istifadəçi adı ilə həm mənsəb, həm də mənbə serverlərimizə daxil olmamızı test edək. Şifrə tələb olunmayıacaq.

**Qeyd:** Əgər açarlar yaranan zaman passphrase yazmışıqsa, qoşulma zamanı istənilən şifrə həmin passphrase-dir.

Eyni strukturu bütün serverlər üçün yaratsaq, onlar açarlar ilə qeydiyyatdan keçəcəklər. Quruluş DSA tipli açarlar üçün də eynidir.

**/root/.ssh/authorized\_keys**

- Bu fayl **/etc/passwd** faylinin ekvivalentidir. (Yəni serverimizə qoşula biləcək istifadəçilərin siyahısını özündə təşkil edir. Bu fayla həmin istifadəçilərin public RSA, ya da DSA açarlarını yazmaları kifayət edir.)

## Putty Windows

Başlığımızda UNIX/Linux əməliyyat sistemlərində SSH client vasitəsilə edilən işlərin Windows/MacOS Desktop-larda Putty client vasitəsilə istifadə edilməsini açıqlayırıq.

### Putty(CMD)

Windows maşınımızda "<http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.60-installer.exe>"-putty installer-i endiririk və yükləyirik.

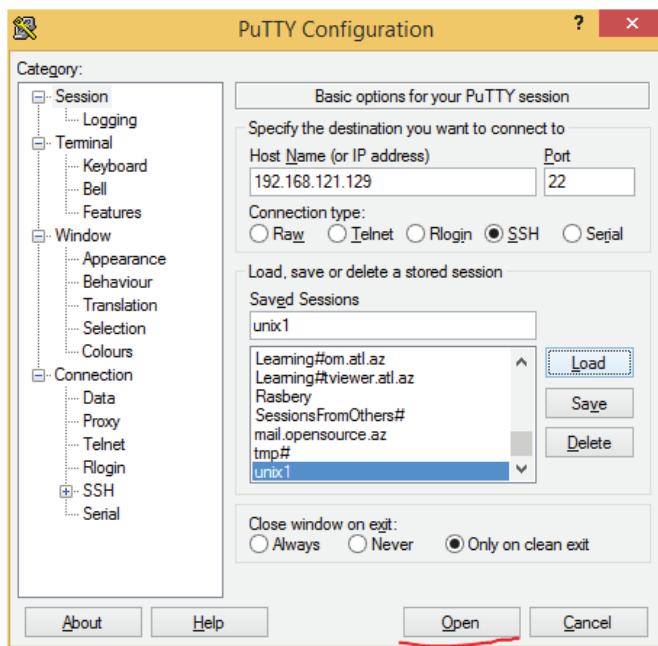
Putty installer yüklənən ünvana **cmd**(Command) ilə daxil oluruq və aşağıdakı ünvanı daxil edirik.

**cd C:\Program Files\PuTTY**

- Putty-nin ev qovluğuna daxil oluruq. **x64** platformalarda "C:\Program Files (x86)\PuTTY>" olacaq.

**putty -load unix1**

- Öncədən yadda saxlanılan **unix1**-adlı serverə daxil oluruq. Bu əmr GUI vasitəsilə aşağıdakı şəkildəki işi görür.



**putty -P 2222 -load unix1**

- Bu əmrlə **unix1**-adlı serverin **2222**-ci portuna qoşuluruq.

**putty -2 -ssh 10.0.0.2**

- SSH version **2** ilə **10.0.0.2** IP ünvanlı serverə qoşuluruq.

**putty.exe root@10.0.0.2 -pw freebsd**

- Bu əmrlə biz **10.0.0.2** IP ünvanlı serverimizə root istifadəçi adı ilə və **-pw "freebsd"** şifrəsi ilə birbaşa qoşuluruq.

**Qeyd:** Eyni ilə siz CMD vasitəsilə **psftp.exe** və **pscp.exe** əmrlərindən istifadə edə bilərsiniz.

**pscp -2 1.mp4 root@10.0.0.5:**

- pscp (version 2 ilə) **1.mp4** faylini 10.0.0.5 İP ünvanlı serverə root istifadəçi adı ilə nüsxələyecək.

**pscp -v root@10.0.0.5:test.exe .**

- pscp -v (detallı informasiya rejimi) ilə root istifadəçi adıyla 10.0.0.5 serverindən test.exe faylini mənbə qovluğuna nüsxələyir.

**pscp -v  
psftp root@10.0.0.5**

- Detallı informasiya rejimi.  
- psftp root istifadəçi adı ilə 10.0.0.5 İP ünvanlı serverə qoşulur.

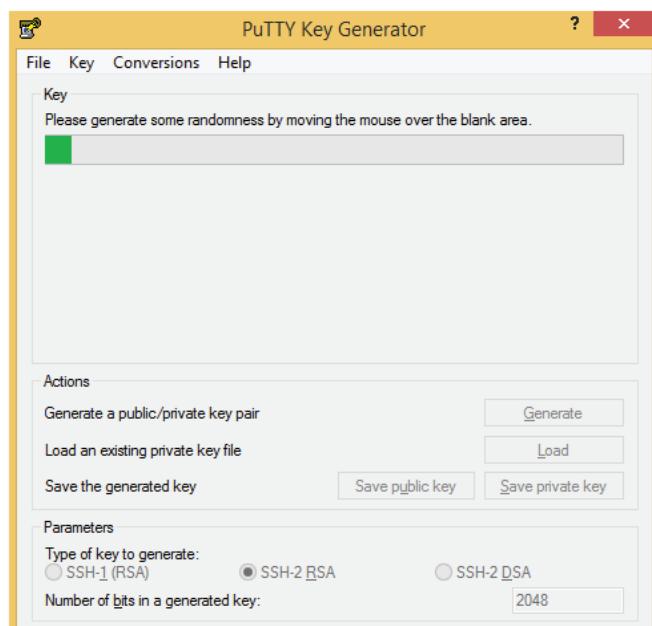
**pstfp -v -i id\_rsa.ppk -P 22 -l root 10.0.0.5**

- pstfp görüyü işləri jurnallayaraq **id\_rsa.ppk** faylini 22-ci port ilə **10.0.0.5** İP ünvanlı serverə nüsxələyir.

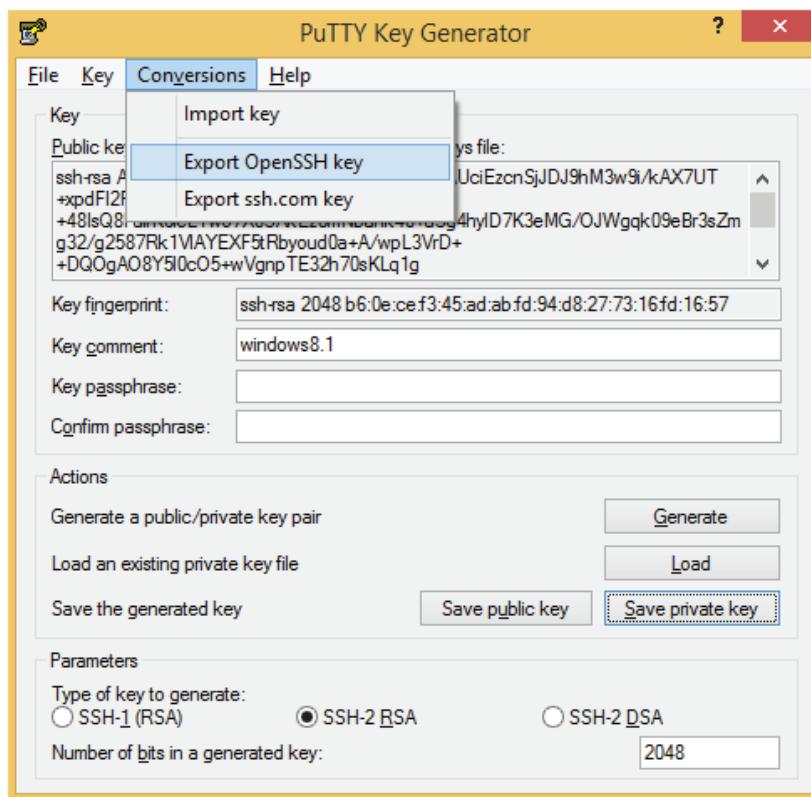
### Putty PKI login

C:\Program Files\PuTTY

- Bu ünvanda puttygen.exe fayl ilə yeni açar generasiya edirik və o müddətdə mousumuzu boş kvadratın içində oynadırıq (Mousun hərəkətinə əsasən açar generasiya edilir). Şəkildə göstərilir.

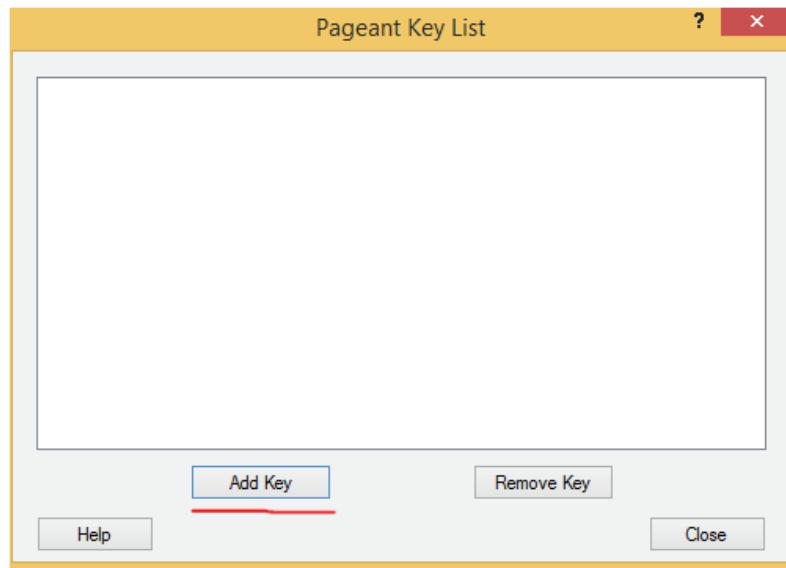


Sonra Public açara lazım bildiyimiz şerhi yazıp **Save public key** düymesini sıxaraq, yadda saxladığımız ünvanda **ir\_rsa.pub** adı ile yadda saxlayırıq. Ardınca da Private açarı lazım bildiyimiz ünvanda **Save private key** düymesini sıxaraq, **id\_rsa** adla yadda saxlayırıq. Sonra Conversations-a daxil olub, "**Export OpenSSH key**" edib **id\_rsa\_openssh** adla yadda saxlayırıq.



"**Putty Key Generator**"-un Generasiya nəticəsində alınan orta pəncərədə olan simvolların hamısını nüsxələyirik. Və bu simvolları qoşulacağımız serverin (**/root/.ssh/authorized\_keys**)-faylinin sonuna nüsxələyirik.

Ardınca **C:\Program Files\PutTY** ünvanında olan **Pageant**-i işə salırıq (Qeyd: O, avtomatik aşağıya düşür, orada axtarın). Sonra "**Add key**" və yaratdığımız "**private key**"-i browse edib seçirik.



Sonda isə "Putty key Generator"-un yaratdığı public açarı hansı serverdə yadda saxlamışıqsa, ona qoşuluruq. İstifadəçi adını daxil edib, avtomatik serverə qoşuluruq.

### SSH File Sharing

SSH vasitəsilə fayl paylaşmaq istəsək, bizi fusefs-sshfs adlı paket lazım olacaq. Bu başlıqda biz SSH vasitəsilə faylları NFS formatında paylaşacaqıq.

```
pkg_add -r fusefs-sshfs  
pkg install fusefs-sshfs
```

- FreeBSD8.4 üçün **fusefs-sshfs** paketini yükleyirik.  
- FreeBSD9.3,10.1 üçün paketi yükleyirik.

Disk paylaşacağımız serverdə aşağıdakı addımları edirik:

```
kldload fuse
```

- **fuse.ko** modulunu yükleyirik.

```
echo "kldload /usr/local/modules/fuse.ko" > /etc/rc.d/fusestart
```

- Sistem yenidənyüklənməsindən sonra işleməsi üçün **fuse.ko** modulunu yükleyirik. StartUP qovluğununda fusestart faylinə əlavə edirik.

```
chmod +x /etc/rc.d/fusestart
```

- Faylı yerinə yetirilən edirik.

```
sudo sysctl vfs.usermount=1
```

- İstifadəçilərə mount etməyə izin veririk.

```
echo "vfs.usermount=1" >> /etc/sysctl.conf
```

- StartUP-a əlavə edirik.

**Qeyd:** Əgər bu paylaşılmış ünvanın sistem yenidənyüklənməsindən sonra istifadəcidə işləməsini istəsəniz, onda SSH serverlər mount istifadə edəcək istifadəçi adı ilə bir-birile açarla qeydiyyatdan keçməlidir.

Sonda istifadəçi maşınınımızdan **192.168.121.130** IP ünvanlı serverimizin **/mnt/docs** qovluğunu öz daxili **/mnt/docs** qovluğunumuza root istifadəçi adı və qrup adından mount edirik.

```
sshfs -o uid=0 -o gid=0 -o allow_other root@192.168.121.130:/mnt/docs /mnt/docs/
```

# Revision Control System, TempFS və MemFS

Bu mövzu faylların versiyalarla qeydə alınması və bərpası, Temp fayl system (məlumatın RAM-da saxlanılması) və virtual fayl system MemFS haqqındadır.

## Revision Control System

Hərdən çox vacib serverlərdə seçilmiş faylda olan dəyişiklikləri və kim tərəfindən edildiyini təyin etmək önəmli olur. Onda RCS istifadə olunur.

**mkdir /etc/RCS**

- Yaratdığımız qovluqda istədiyimiz faylların control versiyaları olacaq.

**ci -l /etc/rc.conf**

- Burada description yazib ->**ENTER** sıxırıq -> və sonra nöqtə yazib **ENTER** sıxırıq. (avtomatik olaraq ",v" işarəsi ilə "/etc/RCS" ünvanında saxlanılır).

-r--r--r-- 1 root wheel 445B Mar 24 18:51 rc.conf,v

Saxlanılan faylda dəyişiklik etmək üçün aşağıdakı işləri görürük.

**co -l /etc/rc.conf**

- Bu əmr "/etc/RCS/rc.conf" faylini "/etc/rc.conf" faylına atacaq (soruşur ki, silib üstünə yazım, ya yox y/n). Sonra faylimızda dəyişiklik edib yadda saxlayırıq.

- ci -u /etc/rc.conf**
- "/etc/rc.conf" faylında etdiyimiz dəyişikliklər barədə "/etc/RCS/rc.conf" arxivimizdə təsdiqləyirik.
- rcsdiff /etc/rc.conf**
- Bu əmr "/etc/RCS/rc.conf" faylı ilə "/etc/rc.conf"-i arasında fərqi yoxlayacaq. (Və hansı dəyişikliklər olduğunu çap edəcək.)
- rcs -l /etc/rc.conf**
- Etdiyimiz dəyişiklikləri təsdiqləyirik. (Sonda mütləq bu edilir: "ci -u /etc/rc.conf", Sonda nöqtə simvolunu daxil edirik və **ENTER** sıxırıq)
- Faylı bərpa edək. Orijinal **/etc/rc.conf** faylında nə isə silək və yadda saxlayaqq.
- co /etc/rc.conf**
- Ən son dəyişiklik edilən vəziyyətdən qaytarış ("/etc/RCS/rc.conf" faylından "/etc/rc.conf" faylinə bərpa ediləcək).
- co -rl.1 /etc/rc.conf**
- "/etc/rc.conf" faylini seçdiyimiz versiyaya geri qaytarır.
- co -p -rl.1 /etc/rc.conf**
- '/etc/rc.conf' faylinin öncəki vəziyyətini çap edir (-p previous).
- rlog /etc/rc.conf**
- '/etc/rc.conf' faylinin üzərində edilmiş dəyişikliklərin tam siyahısını çap edir.

### Unix Memory fayl System

FreeBSD əməliyyat sistemində "**tmpfs**" var.

Açıqlaması o deməkdir ki, biz lazımı informasiyanın daha sürətli işləməsi üçün onu sərt olaraq həmişə RAM-da işlədə bilərik (RAM - Random Access Memory). Yəni bizim "**tmpfs**" ünvanında saxladığımız informasiya yenidənyüklənmədən sonra yox olacaq. (Əgər **tmpfs**-dəki məlumat biziə lazımdırsa, mütləq sistemin yenidənyüklənməsindən əvvəl onu sərt diskə nüsxələməliyik).

TMPFS-i aşağıdakı şəkildə aktiv edirik.

**/etc/rc.conf** StartUP quraşdırılma faylımiza aşağıdakı sətirləri əlavə edirik:

**tmpmfs="YES"**

- Sistem avtomatik olaraq temp ünvanını "/tmp"-də təyin edir.

```
tmpsize="200m"  
tmpmfs_flags="-S"
```

İndi işə komanda sətrindən yaradaq:  
**mdmfs -s 8m md /home/cavid/test**

- **cavid** adlı istifadəçinin ev qovluğunda "**test**" qovluğu üçün 8MB-lıq **tmpfs** yaradıldı (**-s size "md"** adında ilk boş olanı istifadə edəcək).  
Qeyd: Unutmayın ki, **/home/cavid/test** qovluğu mövcud olmasa, əmr işləməyəcək.

```
mdmfs -F diskimage.file md /mnt
```

- **imagefile-i "/mnt"** ünvanına mount edirik.  
(**-F** image fayla gedən yol üçün istifadə olunur).

```
mdmfs -F diskimage.file md9 /mnt
```

- **imagefile-i "md9"** alət adı ilə sərt olaraq **"/mnt"** ünvanına makedevicemountfs edirik.

TMPFS üçün özümüzdə disk yaradıb silə bilərik:

```
mdconfig -d -u 41
```

- **md41** adlı disk yaradın.(-**u** rəqəm verin, **-d** disk yaradandan sonra sistemdən ayırın).

```
mdconfig -a -t vnode -f /home/cavid/freebsd_7.3-stable.iso md0
```

- (**-a** attach edin, **-t** tipi "vnode" olan, yəni "**iso**" **-f** faylı **md0** diskinə).

```
mdconfig -l
```

- (**-l** quraşdırılmış alətlərin siyahısını çap edirik.)

```
mdconfig -l -u 1
```

- (**-u** yalnız **md1** alətini **-l** çap edin).

İndi yaratdığımız tmp-i StartUP-a təyin edirik:

**/etc/fstab** – disk StartUP faylına aşağıdakı sətri əlavə edirik:

```
md      /home/cavid/test      mfs      rw,-s8m      0      0
```

## Memory File System

**Qeyd:** Əgər biz fayl sistemimizi köçürürlən etmək istəsək, (başqa sözlə, fiziki diskə bağlı olmayan FS), onda memory File System yaratmaq olar.

**dd if=/dev/zero of=mydisk count=2048000**

- 1GB həcmində **mydisk** adlı fayl yaradaq.

**mdconfig -a -t vnode -f mydisk -u 2** - "mydisk"-i "-a" mənimşədin, "-t" tipi virtual node olan "-u" md ələtinin 2-cisinə.

**bslabel -w md2 auto**

- "**bslabel**"-lə **md2**-yə ilk hissəni "**auto**" ilə əlavə edin.

**newfs md2a**

- "**newfs**"-lə "**md2a**" hissəsinə UFS2 fayl sistem yazırıq.

**mkdir /mnt/mydisk**

- mount ediləcək "**/mnt/mydisk**" ünvan üçün qovluq yaradırıq.

**mount /dev/md2a /mnt/mydisk**

- "**md2a**" virtual ələtini **/mnt/mydisk** ünvanına mount edirik.

**Qeyd:** Artıq virtual fayl sistem mount olunduqdan sonra biz onu istədiyimiz fayl sistemə mount edə bilərik. Fayl sistemə mənimşətməni bitirdikdən sonra isə, onu tərk edib umount edə bilərik.

**cd /mnt/mydisk**

- Mount etdiyimiz ünvana daxil oluruq.

**mkdir test**

- Fayl sistemdə test adlı qovluq yaradırıq.

**cp /etc/hosts .**

- "**hosts**" faylini olduğumuz qovluğa nüsxələyirik.

**umount /mnt/mydisk**

- "**/mnt/mydisk**" qovluğunu umount edirik.

**Qeyd:** Virtual fayl sistem umount olduqdan sonra biz onu başqa sistemə də köçürə bilərik və ya diskə yazıb başqa fayl sistemdə istifadə edə bilərik. Əgər istəməsək, sadəcə sile bilərik.

# Snapshots, File System Checking və Disk Quota

Bu başlığımızda fayl sistemin snapshot-lar vasitəsilə rezerv edilməsi, fayl sistemin yoxlanışı və fayl sistemlərə məhdudiyyətin təyin edilməsi haqqında danışılır.

## Snapshots

Snapshot-lar fayl sistemdə olan faylların strukturunu öz yaddaşında bir kiçik faylda saxlamaq imkanına malikdir. Misal üçün, desək ki, sizin əməliyyat sisteminin "/" fayl sisteminin işlek vəziyyətdə tutduğu həcmi 8.4G-dir. Təsəvvür edin ki, snapshot-un həcmi 64MB olacaq. Siz 64MB həcmində olan ünvandan istilənilən lazımlı olan faylı bərpa etmək imkanına sahib olacaqsınız.

**Qeyd:** Snapshot çıxarılan vaxt snapshot 'slice'-la eyni ünvanda çıxarılmalıdır.

Snapshot çıxarılan ünvandan bir faylı yaddaşımızda saxlayaqla, snapshot çıxarılandan sonra onu silib, sonra da snapshotdan bərpa edək.

İşimiz:

1. '/usr'-in snapshot-unu götürürük.

**Qeyd:** Fayl sistemin üzərində sujournal işlek vəziyyətdə olarsa, siz snapshot-u götürə bilməyəcəksiniz. Sujournal-ın söndürülməsi proseduru 7-ci başlıqda "**FreeBSD Dump** (Rezerv nüsxə) və **Restore** (Bərpa edilmə)" mövzusunda açıqlanır.

- a. 'mount -u -o snapshot /usr/.snap/snap1 /usr && echo \$?'
  - b. 'mksnap\_ffs /home /home/.snap/snap2' - Snapshot çıxarmağın alternativ üsulu.
2. '/usr'-in Snapshot-unu geri qaytaraq.
- a. 'mdconfig -a -t vnode -f /usr/.snap/snap1 -u 6' - '/dev'-də, '/dev/md6' (-u 6 o deməkdir) adlı alet yaradırıq.
  - a1. 'mdconfig -a -t vnode -f /usr/.snap/snap2 -u 7' - '/dev'-də, '/dev/md7'(-u 7 deməkdir) adlı alet yaradırıq.

**Qeyd:** Snapshot üçün virtual disk yaratdıqda bu xəbərdarlığın "mdconfig: WARNING: opening backing store: /.snap/snap1 readonly" çap edilməsi normaldır.

- b. 'mkdir /snapshots; mount -o ro /dev/md6 /snapshots' - Snapshot verilənləri bu ünvandan istifadə edək.
- b1. 'mkdir /snapshots2; mount -o ro /dev/md7 /snapshots2' - Snapshot verilənləri bu ünvandan istifadə edək.
- c. Və nəticədə, '/snapshots2' ünvanından bizə gərəkli olan informasiyanı lazım olan ünvana nüsxələyək.

### File System Checking (Fayl sistemin yoxlanılması)

Fayl sistemimizin yoxlanılması üsullarını bilmək çox önemlidir, çünkü server otağının işiq və UPS-lə bağlı problemi ola bilər. Server işlek vəziyyətdə hansısa real tranzaksiya getdiyi müddətdə işıqlar sönərsə, fayl sistem zədələnə bilər. Bu hallarda bizim köməyimizə **fsck** çatır. Başlığımızda **fsck** və onun istifadəsində bizə yararlı olan bir neçə əmri açıqlayıraq.

- |                                |   |
|--------------------------------|---|
| <b>fsck_ufs /dev/da0s1e</b>    | - "/dev/da0s1e" aletini ufs yoxlanış et.  |
| <b>fsck_ufs -B /dev/da0s1f</b> | - '-B' arxa fonda fayl sistemi yoxla.   |
| <b>fsck_ufs -B -f da0s1e</b>   | - "fsck_ufs" özü avtomatik 'da0s1e' ünvanını təpib "-B" arxa fonda "-f" məcburi rejimində yoxlanış edəcək, hətta təmiz olsa belə. |

```
fsck_ufs -d /dev/da0s1a
```

- "-d" debugging rejimdən başqa, bütün əmrləri çap edir.

```
fsck_ufs -y /dev/das01a
```

- Əgər hansısa faylı bərpa edə bilərsə, bütün suallara "yes" cavabı verin.

```
fsck_msdosfs /dev/dals2
```

- "msdosfs" fayl sistemi yoxlanış et.

```
df -hi
```

- Diskləri "-h" insan tərəfindən oxunula bilən və "-i" istifadə olunan inode-ların siyahısını çap et.

```
df -hl
```

- "-h" insan tərəfindən oxunula bilən formatda və "-l" daxili fayl sistemimizi çap et. Yəni əgər biz nfs və ya SMBFS-lə paylaşımı mount etmişiksə, onlar çap edilməyəcək.

```
df -ht ufs
```

- "-h" insan tərəfindən oxunula bilən formatda və fayl sistem "-t" tipi 'ufs' olanı çap et.

```
df -h /home/
```

- "-h" insan tərəfindən oxunula bilən formatda ancaq "**home**" slice-ni çap et.

```
du -sh /home/file
```

- "-h" insan tərəfindən oxunula bilən formatda "-s" seçilmiş fayl üçün "**/home/file**" faylinin həcmini çap et.

```
du -sch /home /var
```

- '-h' insan tərəfindən oxunula bilən formatda "**/home**" və "**/var**" slice-nin ayrı həcmi və "-c" ikisi birlikdə ümumi həcmini çap et.

```
find /var -maxdepth 1 -type d -exec du -sh {} \;
```

- Axtar "**/var**" qovluğunda, "-maxdepth" dərinliyində 1-ci səviyyə qovluqların hamisində "-type" tipi "d" qovluq olanları və nəticələrinin həcmini çap et. Yəni "**/var**" qovluğunda olan yalnız bütün 1-ci qovluqların həcmini çap edəcək.

### Disk Quotas(Fayl sistemlərə quota-nın mənimsədilməsi)

Siz öz fayl sisteminizdə çoxlu sayıda istifadəçilərə saysız xidmət imkanları verə bilərsiniz. Ancaq heç bir istifadəçi müəssisəyə aid olan resursların məhdudiyyəti haqqında düşünməyəcək və resurs məhdudiyyəti olmayan ünvanda informasiyanın təkrar nüsxələrinin qalması haqqında da düşünməyəcək. Bu halların yaşanmaması üçün də quota təyin etmək imkanı mövcuddur.

İmkanları:

1. Hər bir istifadəçi və qrupa diskin limitlənməsi
2. İki limit metodikası var
  - a. **Block** (disk) istifadəsi
  - b. **Inode** istifadəsi
3. Hər bir limitləmə metodu iki limit tipindən ibarətdir.
  - a. **Soft limit** (susmaya görə 1 həftəlik kirayə müddəti), kirayə müddəti user/group-a həmin müddət bitənədək limit həddini aşmağa izin verir, yalnız HARD limite çatanadək).
  - b. **HARD limit** (həddi aşmaq mümkün deyil).
4. Yalnız '**/etc/fstab**' faylında lazım bilinən fayl sistemə '**userquota**', ya da '**groupquota**' təyin olunduğu halda işləyir. Misal üçün, aşağıdakı sətirdə **/usr** fayl sistemimizə userquota təyin etmişik.

```
/dev/da0s1f          /usr          ufs      rw,userquota  2      2
```

5. Quota özünə aid olan hər bir informasiyanı root-da saxlayır.

Tələbləri:

1. Kernel Compile olmalıdır (Göstərilən opsiya ilə): **options QUOTA**
2. '**/etc/rc.conf**' – StartUP-a əlavə edirik.
  - a. '**enable\_quotas="YES"**'
  - b. '**check\_quotas="NO"**'  
- boot-vaxtı, '**quotacheck**' təyin olunmuş fayl sistemlərində yoxlanış eləmə.

İşimiz:

1. Quota-nı aktiv edək: '**/home**'
  - a. '**/etc/fstab**'  
- '**/home**' slice-i üçün faylda lazımi quraşdırma növbəti sətirdəki kimi olacaq. Kernel-imizi quota ilə kompilyasiya edirik.  

```
/dev/da0s1f          /usr          ufs      rw,userquota  2      2
```
  - b. '**/etc/rc.conf**' faylinə quota üçün lazımı sətirləri əlavə edirik.
  - c. '**quota -v**'  
- Bütün fayl sistem daxilində axtarış edir və yoxlayır ki, onların hansında quota aktiv olunub.

- d. `'export EDITOR=ee'`
  - e. `'edquota -u cavid'`
- Fayl redaktoru "ee" təyin edirik.  
- `$USER=cavid` üçün quota faylı bize lazım olan kimi dəyişək.

**Quotas for user cavid:**

```
/usr: kbytes in use: 18, limits (soft = 50000, hard = 75000)
      inodes in use: 9, limits (soft = 0, hard = 0)
```

Burada 'cavid' adlı istifadəçiye 50MB soft limit və 75MB hard limit təyin olunub.

**Qeyd:** Susmaya görə yaradılan quotafayl-da hər bir opsiya sıfır olur, bu da limitin olmamağı anlamına gəlir.

2. Kəsintinin qarşısını alaq.

- a. `'dd if=/dev/zero of=/home/cavid/quota.1 count=20 bs=1024k'`
  - 20 MG-liq fayl generasiya edirik.
- b. `'quota -v cavid'`
  - `Cavid` istifadəçisinə təyin olunmuş quotaları çap edirik.

Disk quotas for user cavid (uid 1002):

Filesystem	usage	quota	limit	grace	files	quota	limit	grace
/	0	50	75		0	0	0	

- c. `'quotacheck -a'`
  - `/etc/fstab`-i oxuyaraq bütün fayl sistemdə olan quotaları oxuyur.
- d. `'quotacheck -v /home'`
  - Yalnız `'/home'` slice-na təyin olunmuş quotanı yoxlayırıq.

**Qeyd:** Ümumiyyətlə, '/' fayl sisteminə quota təyin eləmək olmaz. Yaxşı olar ki, sistem yüklenəndə `'/home'` ayrıca slice kimi yaradılsın.

- e. `'checkquota -v /usr'`
  - `/usr` slice-ında quotaları yoxlayır.
- f. `'repquota /usr'`
  - `/usr` slice-ında bütün istifadəçilərə aid olan quotaları çap edir.

3. Sonda quotaları işə salaq.

- a. `'quotaon -a -v'` - Bu hissə `/etc/fstab`-da olan bütün fayl sistemləri axtarır, hansında (`userquota`, `groupquota`) varsa, onu işə salır.  
/: group quotas turned on with data file //quota.group  
/: user quotas turned on with data file //quota.user

a.1 Sonda '**cavid**' istifadəçi adından sistemə daxil olurq və ev qovluğunda bir fayl yaradırıq.  
**'dd if=/dev/zero of=quota.buster.1 count=100 bs=10M'**

**Qeyd:** 'root' istifadəçi hər bir halda istənilən istifadəçi üçün təyin edilmiş quotanı aşır.

**Qeyd:** Əgər istifadəçi ona təyin edilmiş quotanı aşarsa və yenidən fayl sistemdən istifadə etməyə çalışarsa, bu, "**/: write failed, user disk limit reached**" səhv ekranında çap ediləcək.

a.2 '**quotaoff /usr**'

- quota **/usr** fayl sistemində olan **I/O** əsaslandırılmış qaydaları söndürür. (Yəni bu fayl sistemə artıq quota təyin edilmir).

4. '**quotacheck -a**' - Statistika yenilənməsini CRON-dan edin.

# Syslogd, Syslog-NG və Newsyslog

Bu başlığımızda Syslog serverimizin qurulması, uzaq maşınlardan jurnalların qəbul edilməsi və jurnalların planlaşdırılması siyasetini açıqlayacağımız.

## SysLog Server

İmkanları:

1. Daemon informasiyalarını jurnallaşdırır (daxili) və (uzaq)
2. Dəstəkləyir: Unix Domain Sockets (**/dev/log**) Internet Sockets (**UDP:514**)
3. Daxili və Uzaq jurnalları özündə cəmləşdirə bilir (**@hostname**)
4. Syslog hər iki halda işləyir: **client/server**
5. Susmaya görə uzaq jurnal quraşdırması açıq olur. Təhlükəsizlik üçün bağlanması məsləhətdir və yalnız ehtiyac olduğu halda açılması lazımdır. İşə salmaq və ya dayandırmaq üçün StartUP faylı '/etc/rc.conf'-a əlavə etmək lazımdır.

İşimiz:

1. '**/etc/syslog.conf**' - əsas quraşdırma faylı
  - a. Sol tərəf - imkanların səviyyəsi
  - b. Sağ tərəf - mənsəb (**files|remote hosts|pipes**)

**Qeyd:** Syslog başqa səviyyədə təyin olunmayıbsa, o, daha yüksək səviyyəni tutur.

Syslog səviyyələri:

- a. **debug** (0)
- b. **info** (1)
- c. **notice** (2)
- d. **warning** (3)
- e. **error** (4)
- f. **critical** (5)
- g. **alert** (6)
- h. **emergency** (7)

- Problemlerin həll olunması və təpiləməsi üçün çox hallarda bu səviyyə istifadə olunur.

Syslog-un ümumi imkanları:

- a. **MAIL, AUTH, LOCAL0-7, USER, AUTHPRIV, NEWS**

**auth** - İstifadəçi qeydiyyatı olan public informasiya, həmçinin su əmri ilə daxil olanlar.

**authpriv** - root istifadəcisi tərəfindən yetki olunan şəxsi informasiya.

**console** - Normal halda sistem consoluna çap olunan mesajlar.

**cron** - Cron mesajlar.

**daemon** - Yalnız sistem daemonları haqda olan mesajlar.

**ftp** - FTP və TFTP serverlərdən gələn mesajlar.

**kern** - Kerneldən gələn mesajlar.

**lpr** - Line printing sistemdən gələn mesajlar.

**mail** - Mail sistem mesajlar.

**mark** - Bu mesaj hər 20 dəqiqədən bir generasiya olunacaq.

**news** - NEWS daemondan gələn mesajlar.

**ntp** - Network Time Protocol-dan gələn mesajlar.

**user** - İstifadəçi haqqında olan bütün mesajları tutur, əgər istifadəçi tərəfindən istifadə olunan mesajlar syslog siyasetinə uyğun deyilsə.

**uucp** - Unix-to-Unix Copy Protocol mesajları.

**local0...local7**

- Sistem inzibatçısı tərəfindən istifadə oluna biləcək aralıq.

2. "dd" adlı bir server quraşdırıq, hansı ki, uzaq ünvanlardan jurnalları qəbul edəcək.

a.'ee /etc/syslog.conf' (dd server) – faylin əvvəlinə aşağıdakı sətirləri əlavə edək.

Quraşdırırmamızda deyirik ki, **asterisk.az** hostundan gələn istenilən səviyyə jurnalları

**/var/log/asterisk.log** faylına yazın. "+" simvolu host üçün quraşdırma sonu deməkdir.

**Qeyd:** Ad istifadə etdikdə mütləq FQDN istifadə edin.

+asterisk.az

\*.\*

+\*

/var/log/asterisk.log

- Uzaq host-dan gələn log süzgəcinin sonu deməkdir.

b. 'touch /var/log/asterisk.log' - faylini yaradaq.

c. '/etc/rc.conf' - faylimizi elə quraşdırıq ki, syslog serverimizə 'asterisk.az' hostundan gələn jurnalları qəbul eləsin.

syslogd\_enable="YES"

syslogd\_flags="-d -a asterisk.az -a ciscorouter -v -v"

**Qeyd:** -d opsiyası syslog-u debug rejimdə işə salır. Bu, sizin problemləri araşdırıb həll edə bilməniz üçün lazımdır. Server tam işlek vəziyyətə gətirildikdən sonra /etc/rc.conf faylında opsiyanı silməyi yaddan çıxarmayın.

**Qeyd:** Həmçinin aşağıda göstərilən sətirlərlə IP ünvana, tam şəbəkəyə izin verilməsini və Syslog serverinizin hansı IP ünvanında dinləyəcəyini təyin edə bilərsiniz.

syslogd\_flags="-a 192.168.1.9"

- Göstərilən IP ünvandan syslog serverə журнал yollamaq yetkisi var.

syslogd\_flags="-a 192.168.1.0/24"

- Göstərilən şəbəkədən syslog serverimizə журнал yollamağa izin verilir.

syslogd\_flags="-b 192.168.1.1"

- Syslog server -b yalnız 192.168.1.1 IP ünvanında qulaq asır.

d. Ancaq syslog serverimizə '/etc/hosts' faylinda adın IP ünvanına çevrilməsi üçün lazımi sətirləri əlavə etməliyik, əks halda, syslog serverimiz təyin etdiyimiz adı IP ünvana çevirə bilməyəcək və işimiz alınmayıacaq.

192.168.1.110 asterisk.az

192.168.1.122 ciscorouter

e. '/etc/rc.d/syslogd restart' - Syslog serverimizi yenidən işə salırıq.

f. 'netstat -a -p udp' - Syslog serverimzdə portun qulaq asmağını yoxlayaqq (514).

3. Uzaq maşında quraşdırıq ki, (dd)syslog servere jurnal yollasın.

a. '/etc/hosts' – fayla syslog serverin adını əlavə edirik ki, dd adını IP ünvana çevirə bilsin.

**192.168.1.111 dd.az dd**

b. '/etc/syslog.conf' – Faylın sonuna aşağıdakı sətri əlavə edək ki, istənilən jurnalı dd.az serverine yollasın. syslogd servisi yenidən işə salmağı unutmayın.

'\*.\*

@dd.az'

**Qeyd:** Əgər jurnal fayllar bir neçə dəqiqədən sonra da boşdursa, onda 'debug' rejimi işə salıb servisimizin çıxışını analiz etməliyik. '/etc/rc.conf'-da 'syslogd\_flags="-d"' yazmaq lazımdır.

**Qeyd:** Bütün port aralıqlarından syslog mesajının gəlməsini istəyiriksə, onda qəbul etdiyimiz hostun sonuna ':\*' əlavə edirik. '/etc/rc.conf' faylında 'syslogd\_flags' sətrində.

**Qeyd:** Jurnal yollayıb test edə bilərik.

**logger -p warn -t CARD -f /tmp/my.txt** - 'logger' əmri ilə sistemə '-p' info priority olan, '-t' başlıq kimi "CARD" sözünü mənimşəyərək, '-f' '/tmp/my.txt' faylinin hər sətrini jurnalaya yolla deyirik. Susmaya görə '/var/log/messages' faylına yiğilir.

/etc/rc.conf - faylimizə əlavə edirik ki, StartUP-da işləsin:

**syslogd\_enable="YES"**

**syslogd\_program="/usr/sbin/syslogd"**

**syslogd\_flags="-s"**

- İşə salırıq.
- Daemonun ünvani.
- Susmaya görə uzaq serverlərdən gələn jurnalları qəbul eləmə, əgər '-ss' təyin edilibsə, onda syslog server kimi daxildə belə port açmayıñ, yalnız daxili jurnallaşma işini görün.

Jurnallaşmadan çıxarmaq üçün lazımi səviyyələr qeyd edildikdən sonra sonuna none, ya da !=\* yazmaq lazımdır.

**\*.\*;authpriv.none /var/log/all.log**

**\*.\*;authpriv.!!=\* /var/log/all.log**

**+@**

- Simvoldan sonra quraşdırılan bütün jurnallaşma ancaq syslog serverimizin özünə aiddir.

Bəzi misalları açıqlayaq:

**+my.host**

- Qayda my.host-dan gələn bütün mesajlara aiddir.

**!logger**

- Qayda logger-dən gələn bütün mesajlara aiddir.

**!+su**

- İstənilən host üçün su ilə yerinə yetirilmiş bütün mesajlara aiddir.

**!\***

- Qayda bütün mesajlara aid edilir.

Command Line İnterfeys-dən bir neçə misal çəkək:

**/usr/sbin/syslogd -s**

- 's' təhlükəsiz rejim uzaq məşinlardan gələn jurnalları qəbul etməyəcək.

**/usr/sbin/syslogd -4**

- Yalnız IPv4 IP ünvanlar üçün jurnalları qəbul edəcək.

**/usr/sbin/syslogd -a 192.168.0.5/24**

- Syslogd-ə izin verək ki, datagramları **192.168.0.5** IP-ünvanından ala bilsin.

**/usr/sbin/syslogd -b 192.168.0.1/24**

- Syslogd serverə deyirik ki, **192.168.0.1** IP ünvanında qulaq as. (Ancaq əvvəlcə susmaya görə olan servisi dayandırın).

**/usr/sbin/syslogd -d**

- Syslogd-i '-d' debugging eləmək üçün istifadə olunur.

**/usr/sbin/syslogd -m 10**

- Göndərilən “**mark**” mesajlarının arası olan vaxtı 10 dəqiqə təyin edirik. Susmaya görə 20 dəqiqədən bir olur.

## SYSLOG-NG

Siz həmçinin Syslog-NG syslog serveri portlardan yükləyib quraşdırı bilərsiniz.

**pkg\_add -r syslog-ng**

- FreeBSD8.4 server üçün

**pkg install syslog-ng**

- FreeBSD9.3,10.1 server üçün

Syslog-NG quraşdırma nüsxəmizi edirik.

**cp /usr/local/etc/syslog-ng/syslog-ng.conf.sample /usr/local/etc/syslog-ng/syslog-ng.conf**

**/etc/rc.conf** faylına aşağıdakı sətri əlavə edirik ki, sistem yenidənyüklənməsində işə düşsün.  
**syslog\_ng\_enable="YES"** - Startup-a əlavə edək.

**/etc/rc.conf** faylında daxili Syslog serveri söndürürük ki, Syslog-NG ilə konfliktə girməsin.  
**syslogd\_enable="NO"** - Sistem syslogd-ni söndürək.

**kill `cat /var/run/syslog.pid`** - FreeBSD Standart syslogd-ni dayandırırıq.

**/usr/local/etc/rc.d/syslog-ng start** - Syslog-NG-ni işə salırıq.

#### Log siyaseti - 'newsyslog'

İmkanları:

1. Jurnalları **3** özəlliyyə görə paylaşdırır: '**/etc/newsyslogd.conf**' quraşdırma faylında, ya da CLI vasitəsilə:

- a. Həcm
- b. Son arxivdən başlayan vaxt
- c. Hal-hazırkı vaxt qeydiyyat qrafikinə uyğundur, təyin olunmuş vaxtdan 1 saat sonra
2. Avtomatik olaraq hər jurnal faylini rəqəmləmə bacarığı(rotate başlıdıqda)
3. Auto-signals HUP: Sistem jurnalında təyin olunan fayllar üçün
4. 'syslogd'-dən kənardə olan daemona siqnal yollamaq işini asanlaşdırır
5. CRON-la integrasiya olunmuşdur - '**/etc/crontab**'

İşimiz:

1. '**ee /etc/newsyslog.conf**' - faylini analiz edək.

- a. İstifadə olunan flag-lar:
  - 'J' - Bzip2-le Compress edin,
  - 'C' - fayl yaradır, əgər mövcud deyilsə, CLI-da və ya Cron-da istifadə eləmək olur,
  - 'N' - Siqnal üçün proses yoxdur,
  - 'B' - Jurnal faylda binar informasiyadır.

2. Ümumi istifadəni analiz edək:

- a. '**newsyslog -n -v**' - test yoxlanış işini görür.
- b. '**newsyslog -n -F**' - Rotate-i işə salır.
- c. '**newsyslog -v -F -t DEFAULT**' - Rotate-i vaxt möhürü ilə işə salır (-t vaxt möhürü)

3. Özümüzün istifadəsində olan müəyyən bir jurnal faylı quraşdırıq.'/etc/newsyslog.conf' faylinin sonuna aşağıdakı sətri əlavə edək.

```
a. /var/log/camal.log      cavid:cavid      600  365      100  @T00      J'  
      @T00 - Thursday 00:00
```

**Qeyd:** Bu qayda jurnal faylinin **365** nüsxəsini yaradacaq, hansı ki, **bzip2** vasitəsilə sıxılacaq və sysloga siqnal yollayacaq.

**Qeyd:** '**/pid\_file**' - Bu rotate-ə təyin etdiyimiz hər hansı daemon jurnalının '**/var/run**'-da yerləşən pid faylini yoxlayır. Prosesdən çıxış olduqda rotate-i yenidən yoxlayır.



# BÖLÜM 8

## Wget, Curl, LFTP, Rsync, Unison, NFS, vaxt və tarix əməliyyatları

- / Uzaq serverlərə data ötürülməsi və götürülməsi üçün program təminatları
- / Rsync, Unison və NFS istifadəsi
- / Vaxt, tarix və təqvim əməliyyatları
- / Müxtəlif məqsədlərdə istifadə edilən utilitlərin açıqlanması

Internet vasitəsilə hansısa informasiyanın əldə edilməsi üçün spesifik program təminatlarının yüklənməsi və istifadəsi qaydaları, spesifik ftp istifadəçi program təminatının yüklənməsi və idarə edilməsi, fərqli program təminatları ilə təyin edilmiş qovluqların bir serverdən digərinə əzaqdan sinxronizasiya edilməsi və şəbəkə disklerinin serverlər arasında paylaşımı, sistemdə olan vaxt və tarixlə əlaqəli olan bütün işlərin görülməsi və vaxt serverinin quraşdırılması, fərqli utilitlərin istifadə qaydaları (şəkil, video və musiqi) və adlandırılmış kanallar barədə bu başlıqda danışılır.

# Uzaq serverlərə data ötürülməsi və götürülməsi Üçün program təminatları

Başlığımızda uzaq serverlərə informasiyanın ötürülməsi, götürülməsi və ümumiyyətlə, serverimizin daxili qovluğunun uzaq serverə sinxronizasiya edilməsi imkanları açıqlanır.

## WGET və CURL

Hər iki program təminatının sayəsində fərqli üsullarla endirmə imkanları mövcuddur. Ancaq hər iki program təminatı sistemdə susmaya görə olmur və onları yükləmək lazımdır. Bu alt başlıqda hər iki program təminatının imkanları açıqlanır.

Yükləyirik:

<code>pkg_add -r curl, pkg_add -r wget</code>	- FreeBSD8.4
<code>pkg install curl wget</code>	- FreeBSD 9.3, 10.1

İmkanları:

1. Hər iki alət faylı transfer üçün istifadə edir: HTTP & FTP
2. Büyük hədlər üçün çəkmə alətləri.
3. Ümumi protokollardan məlumat alma metodu.

İşimiz:

1. 'curl' istifadəsini açıqlayaq.

**Qeyd:** Curl tanınmış protokollardan istifadə edərək fayl transfer edə bilir. (Məs: **SSH**, **SCP**, **SFTP**, **LDAP**, **DICT**, **Telnet**)

**Qeyd:** Curl-la bir neçə faylı birdən darta bilərik.

a. Digər bir serverdə apache qaldırıb, onun ev qovluğunda 100MB -lıq '**curl.size**' adlı bir fayl yerləşdirək və curl yüklediyimiz serverdən həmin faylı curl vasitəsilə endirək.

a1. **'curl http://192.168.1.123/curl.size'**

**Qeyd:** 'curl' susmaya görə götürdüyü faylı **STDOUT**-a yollayır. (Gərək onu çıxışa bir fayla yönəldək.)

a2. **'curl -O http://192.168.1.123/curl.size'** - Uzaq fayl adına uyğun olaraq, daxildə eyni adlı fayl yaradacaq.

a3. **'curl -O0**

**ftp:ftp@ftp.freebsd.org/pub/FreeBSD/releases/amd64/ISO-IMAGES/7.3/FreeBSD-7.3-RELEASE-amd64-disc{1,2,3}.iso**' - freebsd.org-dan 7.3-cü {1,2,3}-ün sayəsində release-ə aid olan 3-diski də endirəcək.

a4. **'curl ftp://ftp.freebsd.org/pub/FreeBSD/doc/en/books/'** - 'ftp.freebsd.org'-da olan kitab məzmununu siyahılaysaqq.

a5. **curl -O ftp://cavid:freebsd@192.168.0.105/file -Q '-DELETE file'** - 'cavid' istifadəçi adı və 'freebsd' şifrə ilə **192.168.0.105** IP ünvanlı serverə daxil olub, istifadəçinin ev qovluğundan '**file**' adlı faylı öz qovluğumuza endirib bitirdikdən sonra ftp serverdən silirik.

a6. **curl -T book.pdf ftp://cavid:freebsd@192.168.0.105/ -Q "-RNFR book.pdf" -Q "-RNTO freebsd-netbook.pdf"** - 'book.pdf' adlı faylı daxili qovluğumuzdan uzaq **'192.168.0.105'** IP ünvanlı serverə, 'salman' istifadəçi adı, 'freebsd' şifrə ilə yükləyin və serverdə adını dəyişib 'freebsd-netbook.pdf' edin.

a7. **'curl ftp://ftp.freebsd.org/pub/FreeBSD/doc/en/books/'** - FreeBSD-nin rəsmi ftp serverində kitabların məzmununu siyahılaysaqq.

2. 'wget' istifadəsini açıqlayaqq.

a. **'wget http://192.168.1.123/curl.size'**

a1. **'wget**

**http://upload.wikimedia.org/wikipedia/commons/2/2c/Chokladbollar.jpg'** - yalnız bir şəkli endiririk (**chokladbollar.jpg**).

a2. **'wget ftp://cavid:freebsd@192.168.0.105/file'** - ftp serverdən istifadəçi adı və şifrə daxil edərək (**file**) adlı faylı endiririk.

- a3. 'wget --user=cavid --password=freebsd  
**ftp://192.168.0.105/file'** - 192.168.0.105 IP ünvanlı serverdən 'file' adlı faylı cavid istifadəçi adı və freebsd şifrəsi ilə endiririk.
- a4. 'wget --user=cavid --password=freebsd  
**ftp://192.168.0.105/image.jpg'** - cavid adlı istifadəçi adı və freebsd şifrəsi ilə 'image.jpg' faylini 192.168.0.105 IP ünvanlı serverdən endiririk.
- a5. 'wget http://www.google.com' - 'www.google.com'-un ilk **index.html** səhifəsini endiririk.
- a6. 'wget -p http://www.google.com' - 'www.google.com'-un səhifəsini və bütün zədələnmiş linklərini endirəcək.

**Qeyd:** Biz 'index.html'-i bütün endirsek belə, şəkillər açılmayacaq, çünki şəkillərə gedən link qlobal IP ünvana müraciət edir. **index.html** endirilmiş ünvandan açıldıqda isə, o linklər **broken** (qırılmış) olur. Həmin linklərin daxildən açılmasını istəsək, aşağıdakı sintaksisi istifadə edirik.

- a7. 'wget -pk http://www.google.com' - 'www.google.com'-un "index.html"-ni və bütün '-p' (pages) şəkillərini endirin '-K' (local name), amma qlobal link ünvanlarını daxili adları ilə əvəz edin.
- a8. 'wget -pkK http://www.google.com' - 'www.google.com'-un "index.html"-ni endirin və qlobal linkləri daxili adlara əvəz edin, '-K' keep, ancaq orijinal 'index.html'i saxlayın.
- a8-1. 'wget --mirror -p --convert-links -P /disk xaknotdie.org' -  
'--mirror' (bütöv güzgü edir), -p --convert-links (Dünya linkləri daxili adla konvertasiya edir), '-P /disk'(disk qovluğuna)
- a9. 'wget -E http://phone.rabita.az' - Qeyd: Bəzi saytlar ola bilər ki, onların kodları '.asp', ya da '.cgi' yazılmış olur. Onu daxilə endirəndə browser aça bilməyəcək. Ancaq çəkdiyimiz faylin 'html' genişlənmədə çekilməsini 'wget'-ə "-E" opsiyası ilə deyə bilərik. 'phone.rabita.az' saytinın susmaya görə olan index faylini hər bir halda html kimi dərtəcəq.
- a10. 'wget -m http://www.linuxtoys.net' - 'www.linuxtoys.net' saytını rekursiv olaraq bütün fayl və qovluqlarını tamamilə öz daxili qovluğumuza nüsxələyirik. Ancaq qlobal linklər öz yerlərində qalacaq. Bunun üçün aşağıdakı əmri istifadə edirik.
- a11. 'wget -mEkK http://www.linuxtoys.net' - 'www.linuxtoys.net' saytını rekursiv olaraq bütün fayl və qovluqlarını daxili adla tamamilə endiririk.
- Qeyd: Biz endirdiyimiz hansısa ISO və ya DVD faylı "**Ctrl+c**" ilə dayandırıb, sonra davamını çəkə bilərik. Aşağıdakı sintaksisdən istifadə edirik. Əgər '-c' opsiyası istifadə edilməsə, çəkdiyimiz köhne fayl yerində qalacaq, faylin sonuna '.1' əlavə edilərək yenisidən çəkiləcək.

a12. `wget http://example.com/DVD.iso` - Bu linki "Ctrl+C" ilə kəssək, '-c' continue ilə davam edə bilərik. (`wget -c http://example.com/DVD.iso`)

**Qeyd:** 'curl'-lə 'wget'-in fərqi ondan ibarətdir ki, 'wget' susmaya görə aldığı informasiyanı STDOUT-a yollamır.

a13. `wget -o curl.log http://192.168.1.123/curl.size` - çıxışda 'curl.log'-adlı jurnal fayl yaradır.

b. `wget -o curl.log -O curcurl.size http://192.168.1.123/curl.size` - Uzaq HOST-dan daxilə seçdiyimiz ada endiririk.

### LFTP client

İmkanları:

1. Interactive & Non-interactive
2. Çoxlu protokolları dəstəkləyir: **FTP, HTTP(s), SFTP**
3. Avtomatlaşmaq üçün uyğundur.
4. Jobs-un fonda işləməsini rahatlaşdırır.
5. İslədiyi müddət **\$SHELL**-ə qayıtmaq olur.
6. Ötürmə sürətini mehdudlaşdırmaq olar.
7. İşi idarə etmək üçün müxtəlif dəyişən təyin etmək olar.
8. Oxuyur sistem daxilində: '**/etc/lftp.conf**'-dan, ya da istifadəçinin ev qovluğunda olan '**~/.lftpirc**' faylından.
9. Content Mirroring-i dəstəkləyir.
10. BASH-a uyğun olaraq tarixçəni dəstəkləyir.

`pkg install lftp` - FreeBSD9.3,10.1 yükleyirik.

İşimiz:

1. Interactive & Non-interactive interfeysi açıqlayaq.
  - a. `'lftp'` - əmrini daxil etdikdə interaktiv rejim işə düşür.
    - a1. `'set -a'` - Olduğumuz mühitin susmaya görə olan dəyişənlər və mənalarını çap edir.
    - a2. `'lftp'` əmrini daxil edib, sonra `'help'` əmrini daxil etsək, bütün dəstəklədiyi əmrləri görə bilərik.
    - a3. `'!bash'` - biz BASH **\$SHELL**-ə qaytaracaq.
    - a4. `'open ftp.freebsd.org'` - FTP serverə qoşuluruq.

- a5. '**set net:limit-rate 1000:1000**' və '**set net:limit-total-rate 10000:10000**' çəkmə və yükləmə sürətini saniyədə (**1kb**) təyin edirik.  
a6. '**Ctrl-Z**' – arxa fon rejiminə keçirir.

2. LFTP-ni SFTP kimi istifadə edək.

a. '**lftp -u cavid sftp://192.168.1.123**'

**Qeyd:** Əgər **SSH-PKI AUTH** quraşdırılıbsa, onda sintaksis aşağıdakı kimi olacaq.

- a1. '**lftp -u cavid,freebsd 192.168.0.105**' - cavid istifadəçi adı və **freebsd** şifrəsi ilə **192.168.0.105** IP ünvanlı serverə qoşuluruq.  
a2. '**lftp -u cavid 192.168.0.105**' - **cavid** istifadəçi adı ilə **192.168.0.105** IP ünvanlı serverə qoşulacaq.  
a3. '**lftp cavid@192.168.0.105**' - **cavid** istifadəçi adı ilə **192.168.0.105** IP ünvanlı serverə qoşuluruq.  
a4. '**lftp sftp://cavid@192.168.0.104**' - '**lftp**' client-lə **192.168.0.104** IP ünvanlı **sftp** serverə cavid istifadəçi adı ilə qoşuluruq.  
a5. '**lftp -u camal, sftp://192.168.1.123**' - Əmr öncəki ilə eyni işi görür. '**lftp -u camal,**' -istifadəçi adından sonra mütləq vergül olmalıdır.
- b. '**mirror -v**' - Uzaq qoşulduğumuz qovluğu daxili qovluğunuzla sinxronizasiya edir.  
b1. '**mirror -e -v**' - Uzaq serverdə olmayan faylları daxildən silir.  
b2. '**mirror -R -v**' - Daxili məzmunu uzaq hosta sinxronizasiya edir.

# Rsync, Unison və NFS istifadəsi

Bu başlığımızda uzaq fayl sistemlərin öz serverimizə, öz fayl sistemlərimizin uzaq məşinlərə sinxronizasiyası və ümumiyyətlə, fayl sistemlərin uzaq və ya mənbə məşinlərinə birdəfəlik mount edilməsi haqqında danışacaqıq.

## Rsync uzaq rezerv nüsxənin sinxronizasiyası

İmkanları:

1. Uzaq serverin məzmununu '**ssh**'-la sinxronizasiya bacarığı.
2. Syncronizes: Daxili və uzaq kontent - '**cp**' yerinə istifadə oluna bilər.
3. Susmaya görə faylları yeniləyir: Həcmə, ya da dəyişdirilmə vaxtına görə.
4. Çox sürətlidir.
5. HOST-lar və qovluqlar arası kontentin sinxronizasiyası üçün yaxşı alətdir.

İşimiz:

1. Əsas istifadə qaydası.
  - a. '**rsync -avv MƏNBƏ MƏNSƏB**'
  - b. '**rsync -avv \* /tmp/rsync\_temp**' - Daxili məşinimizdə olan iki ədəd qovluğu sinxronizasiya edirik.
  - c. '**rsync -avv /tmp/rsync\_temp .**' - **rsync\_temp** qovluğunun yerləşdiyimiz ünvana nüsxələyirik.

2. Kontentin şəbəkə HOST-ları arasında sinxronizasiya olunması.

a. `'rsync -avv * cavid@192.168.1.123:rsync_temp/'`

- Sinxronizasiya üçün SSH-i istifadə edir.

b. `'rsync -avvz * cavid@192.168.1.123:rsync_temp/'`

- (-z) Transfer müddətində bütün məlumatları sıxış göndərəcək.

c. `'rsync -avvz --log-file=rsync.log * cavid@192.168.1.123:rsync_temp/'`

- Burada isə görülən işi jurnal faylı `'rsync.log'`-a jurnallayacaq.

d. `'rsync -avv * --log-file=rsync.log --bwlimit=100 cavid@192.168.1.123:rsync_temp/'`

- köçürmə sürəti saniyədə 100KB olacaq.

e. `'rsync --delete -avvz --log-file=rsync.log cavid@192.168.1.123:rsync_temp/ .'`

- Əvvəlcə öz serverimizin daxilində hər şeyi sil, sonra da sinxronizasiya et.

**Qeyd:** Ehtiyatlı olun, `'--delete'` hər şeyi silir.

f. `'rsync -n --delete -avvz --log-file=rsync.log cavid@192.168.1.123:rsync_temp/ .'`

- '-n' opsiyası `'--delete'` opsiyasının yaratdığı extra faylı silib sinxronizasiya edir.

`rsync -avz --delete cavid@192.168.192.223:/home/cavid/ papka/`

- 'cavid' istifadəçi adı ilə '192.168.192.223' IP ünvanlı uzaq serverin "/home/cavid" qovluğunda olan bütün informasiyanı öz serverimizin daxilində olan 'papka/' qovluğuna sinxronizasiya edirik.

'-a' rekursiv olaraq uzaq serverdən bütün qovluq və faylları mənbə serverimizə arxiv edirik.

'-z' qəbul etdiyi informasiyanı sıxır.

'-v' verbose rejimdə işləyir.

'`--delete`' bu isə uzaq serverdə olmayan fayl və ya qovluqları mənbə serverdən silir.

**Qeyd:** Gündəlik rezerv nüsxə çıxarmaq istəsəniz, onu müəyyən "rsync" skripti ilə çıxara bilərsiniz. Aşağıdakı şəkildə.

`mkdir /disk/backups`

- Rezerv nüsxələrimiz üçün qovluq yaradaq.

```
rsync --delete --backup --backup-dir=/disk/backups/backup`date +%A` -avz  
cavid@10.0.0.1:/home/Personal/ /disk/backups/current-backup/
```

- '--delete' rezerv nüsxə çıxardıqdan sonra mənbə qovluğumuzda olan və uzaq serverdə olmayan artıq faylları siləcək.

"--backup" rezerv nüsxə çıxarın.

"--backup-dir" rezerv ediləcək nüsxə qovluğu "/disk/backups/backup`date +%A`" rekursiv olaraq, '-a' bütün faylları '-v' verbose rejimdə, '-z' sıxaraq, "cavid" istifadəçi adı ilə, 10.0.0.1 serverindən mənbə serverimizin "/disk/backups/current-backup/" ünvanına rezerv nüsxə edəcək. Birinci dəfə o, bütün faylları "/disk/backups/current-backup/" qovluğuna nüsxəleyəcək. Sonra isə hər yeni gün üçün dəyişiklik edilmiş bütün faylları "/disk/backups/" qovluğunda hər yeni gün üçün yeni qovluğa atacaq. Beləliklə, həftənin hər günü üçün yeni qovluğa ən yeni nüsxə götürüləcək.

```
rsync --delete --link-dest=/raid10/backup-old -avz cavid@10.0.0.1:/home/cavid/  
/raid10/backup-current/
```

rsync-lə qoşulma gedəndə mənbədə yaranan "/raid10/backup-current/" qovluğu "/raid10/backup-old" qovluğuna hard link edəcək. cavid istifadəçi adı ilə 10.0.0.1 IP ünvanlı serverin "/home/cavid/" qovluğundan "/raid10/backup-current" qovluğuna sinxronizasiya edəcək.

#### Unison uzaq rezerv nüsxənin sinxronizasiyası

**Qeyd:** Unison multiplatformda işləyən bir programdır. Onu həm Windows-da, həm də UNIX-də istifadə edə bilərsiniz.

Unison mənbə qovluqları da sinxronlaşdırıb ilir. O, eyni faylda dəyişiklik olduğunu görəndə onun barəsində ekrana mesaj çıxarıb ki, seçim edək.

```
unison /localfolder1 /localfolder2
```

- Sistemdə olan iki qovluq arasında sinxronizasiya olacaq.

```
unison /test1 ssh://salman@10.0.0.1//folder - "/test1" qovluğu ilə 10.0.0.1 IP ünvanlı serverin "folder" qovluğunun sinxronizasiya edəcək.
```

**Qeyd:** İnteraktiv rejimdə iki menyu çıxacaq. Və sual veriləcək ki, mənbə serverdən uzaq serverə və uzaq serverdən mənbə serverə sinxron nüsxələnmə işini görəkmə?

```
unison -help
```

- Əmr unisonun bütün imkanlarını çap edir.

```
unison -doc all | less
```

- Əmr unisonun manualını **text doc** kimi ekrana çap edəcək.

**Qeyd:** Qrafik rejimdə unison istifadə edildikdə hər profil üçün "**~/.unison/**" qovluğununda "**.prf**" genişlənməsində fayllar yaradılır. Həmin faylı unisonla oxumaq olur. Məsələn: "**unison fc-home.prf**"

### Network File System (NFS)

İmkanları:

1. Uzaq fayl sistemlərə şəffaf giriş icazəsi.

2. Teləbləri:

- '**rpcbind**' - müxtəlif portlarla müraciət edir.
- '**nfsd**' - Primary NFS servis.
- '**mountd**' - müraciətləri 'nfsd'-dən keçərək edir.
- '**rpc\_lockd**' - bağlamaq üçün.
- '**rpc\_statd**' - statistikalar.

3. Dəstəkləyir: **UDP** və **TCP**

İşimiz:

1. NFS Server quraşdırıq ki, export-u qovluq üçün edək: '**/projectx**'  
a. '**/etc/rc.conf**' - StartUP faylında lazım olan servisləri işə salaq.  
**rpcbind\_enable="YES"**  
**nfs\_server\_enable="YES"**  
**mountd\_enable="YES"**  
**mountd\_flags="-r"**  
**rpc\_lockd\_enable="YES"**  
**rpc\_statd\_enable="YES"**

b. Serverimizdə '/etc(exports' faylini yaradaq və içində aşağıdakı sətri əlavə edək. Susmaya görə bu fayl mövcud olmur:

'/projectx -ro -network 192.168.1.0/24' - icazə verilir 192.168.1.0/24-cü şəbəkəyə, yalnız oxumağa, '/projectx' qovluğuna.

'/projectx -maproot=root -network 192.168.1.0/24' - 192.168.1.0/24-cu şəbəkəyə icazə verilir, '/projectx' qovluğunda, həm oxuma, həm də yazmağa, root istifadəçi adından.

'/projectx -alldirs -maproot=root -network 192.168.1.0/24' - Qovluqda olan bütün qovluqlar root istifadəçi adından seçilmiş şəbəkəyə tam yetki alır.

**Qeyd:** Hər dəfə '/etc(exports' faylında edilən dəyişiklikdən sonra 'mountd' servisi restart olmalıdır.

**Qeyd:** Paylaşım etdiyimiz NFS serverin qovluq siyahisini 'showmount -e' əmri ilə görmək olar.

b1. **showmount -e** - NFS serverdə nə qədər paylaşım olduğunu siyahılıyırıq.

b1-1. **showmount -e 192.168.0.105** - Client maşından 192.168.0.105 IP ünvanlı serverin paylaşım siyahisini çap edirik.

c. Serverimizdə servisləri işə salaq.

```
/etc/rc.d/rpcbind start  
/etc/rc.d/nfsd start  
/etc/rc.d/mountd start  
/etc/rc.d/lockd start  
/etc/rc.d/statd start
```

**nfsd -u -t -n 4** - Əmr vasitəsilə console-dan işə sala bilərik.

2. Uzaq maşından(yəni client) NFS serverə qoşulmağa çalışaq.

a. FreeBSD NFS\_Client quraşdırıq.

a1. 'ee /etc/rc.conf' fayla yazırıq 'nfs\_client\_enbale="YES"'

a2. '**nfsiod -n 4**' - daxildən uzaq serverimizə 4 ədəd nfs\_client prosesin işləməsinə izin veririk.

- b. '`mkdir /projectx`' - Client maşında da eyni qovluğu yaradırıq.
- b1. '`mount -t nfs 192.168.1.122:/projectx /projectx`' - gördükümüz kimi, **192.168.1.122** IP ünvanlı serverin `/projectx` qovluğunu öz daxili `/projectx` qovluğumuza **NFS** fayl sistem kimi mount etdik.
- b2. '`mount -o rw,hard,intr 192.168.1.122:/projectx /projectx`' - '`rw`' **read write** (oxuma və yazmaq) olsun, '`hard`' **read write** olmasa, '`intr`' serverimizə interrupt siqnal ötürün.
- b3. Client maşında sistemin yenidənyüklənməsindən sonra **NFS** diskimizin avtomatik mount olmasını istəyiriksə, `/etc/fstab` faylına aşağıdakı sətrə uyğun olan sətri əlavə etməlisiniz.  
**192.168.121.130:/projectx /projectx nfs rw 0 0**

# Vaxt, tarix və təqvim əməliyyatları

Əməliyyat sisteminin üzərində olan vaxtin, təqvimin və tarixin düzgün olması çox önemlidir, çünkü eksər daemonlar öz işləmə prinsiplərində ilk olaraq vaxtin düzgülüünü yoxlayırlar. Misal üçün, DNS server Master/Slave sinxronizasiyası və ya Microsoft Domain controller-lə Samba serverin integrasiyasında vaxt düzgün olmazsa, serverimiz DC-yə üzv ola bilməyəcək. Bu başlığımız vaxt haqqında bütün imkanları açıqlayır.

## Network Time Protocol (NTPD)

İmkanları:

1. Vaxt sinxronizasiyası tək və ya bir neçə mənbədən.
2. Sinxronizasiya NTP serverdən sonra **1000** saniyə fərqi ilə.
3. 'ntpdate' – yüklənmə müddəti uyğunlaşmadan asılı olmayaraq, vaxtin sinxronizasiyası.  
`'ee /etc/rc.conf'` - Fayla aşağıdakı setirləri əlavə edirik.  
`ntpdate_enable="YES"` - Sistemin daxili tarixini dəqiqləşdirir.  
`ntpdate_hosts="0.asia.pool.ntp.org"` - Bu vaxt serveri ilə.
4. Dəqiqlik iyerarxiyası ki, **16** səviyyə istifadə edir. **1** - ən dəqiq, **16** - ən qeyri-dəqiq.

**Qeyd:** İyerarxiyalar arası vaxt millisaniyələrlə fərqlənir.

İşimiz:

1. Saati 'ntpdate' istifadə edərək təyin edək.

- a. 'ntpdate NTP\_SERVER'
2. NTPD quraşdırıq.
- a. '/etc/ntp.conf' - dolduraq: 'server IP|DNS\_NAME' bölmələrini
  - b. '/etc/rc.conf' - faylına əlavə edək: 'ntp\_enable="YES"' və  
'ntp\_sync\_on\_start="YES"'
  - c. 'ee /etc/ntp.conf' - lazımi sətirləri əlavə edirik.
- ```
server 116.193.170.30 iburst
server 202.112.10.36 iburst

driftfile /var/db/ntp.drift
logfile /var/log/ntp.log
restrict default ignore
restrict 116.193.170.30
restrict 202.112.10.36
```
3. NTPD aletlərini açıqlayaq.
- a. 'ntpq' - NTPD-yə müraciət yollamaq üçündür və cavabı STDOUT-a yollayır, həmçinin interaktiv rejimdə də işləyir.
    - a1. (-4n) - IPv4-ə (-n) adı İP ünvana çevirmədən müraciət yollayın.
    - a2. 'ntptrace' - trasirovka üçün istifadə olunur.

#### Network Time Protocol-un istifadəsi

/etc/ntp.conf - quraşdırma faylına aşağıdakı sətirləri əlavə edirik:

```
server 116.193.170.30 iburst burst
```

- **Qeyd:** mütləq ad əvəzinə İP yazın. Və mütləq qeyd etdiyiniz NTP server İP ünvanlarına restrict bölümündə izin verin.

```
server 202.112.10.36 iburst burst
```

```
server 180.211.88.211 iburst burst
```

```
server 78.111.50.2 iburst burst
```

```
driftfile /var/db/ntp.drift
```

```
logfile /var/log/ntp.log
```

```
restrict default ignore
```

```
restrict 116.193.170.30
```

```
restrict 202.112.10.36
```

```

restrict 180.211.88.211
restrict 78.111.50.2
# 192.168.121.0 və localhost-umuza sinxronizasiya etmək üçün icazə veririk.
restrict 192.168.121.0 mask 255.255.255.0 nomodify notrap
restrict localhost

iburst                                - Parametri ilk sinxronizasiya prosesini
  sürətləndirir.
burst                                 - NTP serverə sinxronizasiya cəhdı üçün 1 paket
  əvəzinə 8 paket yollamağı təyin edir.

touch /var/db/ntp.drift                - Drift faylı yaradırıq.
touch /var/log/ntp.log                  - Jurnal faylini yaradırıq.

/etc/rc.conf - NTPD servisi StartUP-a əlavə edirik.
ntp_enable="YES"                      - Network Time Protocol-u işə salırıq.
ntp_program="/usr/sbin/ntp"          - Proqramın binar fayl ünvanı
ntp_config="/etc/ntp.conf"            - ntpd(8) quraşdırma faylı
ntp_sync_on_start="YES"                - Vaxtı ntpd işə düşən kimi sinxronizasiya edin.
ntp_flags="-p /var/run/ntp.pid -f /var/db/ntpdrift" - NTPD proses ID faylı və baza faylı.

/etc/rc.d/ntp start                   - Daemonu işə salırıq.

ntpdate pool.ntp.org                 - Sistemi həmin anda da vaxtin götürməsi üçün
  "ntpdate" CLI-dan əmrindən istifadə edilir.

Qeyd: Ancaq "ntp" daemon söndürülmüş olmalıdır. "ntp" daemon ilə birgə işləmək
fürəcək. "ntp" əmrinin özü ilə vaxtı yeniləmək olar.

ntp -qg                                - '-q' opsiyası ntpd-yə deyir ki, daemonun özü
  işlək vəziyyətdə qalsın, saatı təyin edin, sonra
  çıxın. '-g' opsiyası onun üçündür ki, sistem
  dünya saatından 1000 saniyə fərqi ilə işləyirsə,
  ntpd sistemə panic siqnalı ötürüsün.

```

```
ntpq -p                                - Sinxronizasiya statusunu yoxlayırıq.
remote          refid      st t when poll reach   delay   offset   jitter
=====
*proxy04.ispros. 192.93.2.20  2 u    64   64    76  304.184   14.968   4.884
+gus.buptnet.edu 202.112.10.60 3 u    64   64    27  429.674   -2.025  15.024
-ns4.king.net.id 130.149.17.21 2 u    1    64    77  357.852   27.087  32.294
+host-2.net50.so 193.79.237.14 2 u    3    64    77  97.272   -38.792   3.276
```

Görünən sütunlarımızı açıqlayaq:

**remote**

- Uzaq ntp serverlərin adları (Bizim halda 0/1/2/3.asia.pool.ntp.org-dan götürülmüşdür.)

**refid**

- Serverdir, hansı ki, bu uzaq NTP serverlə vaxt sinxronizasiyasını edir.

**st**

- Stratum (səviyyə) uzaq serverdə.
- 1** - ən yüksək, bu serverlər dəqiq vaxtin hesablanması üçün bahalı avadanlıqlardan istifadə edir;
- 16** - adı maşın/istifadəçi.

**t**

- peer tipi (**u = unicast, m = multicast**)

**when**

- Serverlə nə qədər müddət öncə sinxronizasiya baş vermişdir.

**poll**

- Saniyelərlə olan tezlik, hansı ki, onunla NTP daemon peer-lə sinxronizasiya edir.

**reach**

- Serverlərin mövcudluğu statusu, əgər serverlə olan son **8** sinxronizasiya statusu cəhdilə ugurlu olarsa, bu mənə **377** səviyyəsində stabillaşdırılır.

**delay**

- Serverdən qayidian cavabın millisaniyələrlə olan gecikməsi.

**offset**

- Sistem vaxtı ilə uzaq server vaxtı arasında

millisaniyələrlə olan fərqi. Mənfi işaretə gecikmə, müsbət işaretə isə tələsmə deməkdir.

## jitter

- Uzaq serverdə vaxtin qarışması.

Peer-dən gələn işaretlər haqqında:

"\*" - Axırıncı dəfə bizimlə vaxt sinxronizasiyası edən peer.

"+" - "yaxşı" (yenilənmə üçün yararlıdır) server.

"-" - "pis" (yenilənmə üçün yararlı deyil) server.

"x" - Server cavab vermir.

**ntpdate -q localhost**

- Əmri NTP serverimizdə daxil edib test edirik.

Əgər aşağıdakı sətrə uyğun olaraq sətirlər görsəniz, demək, quraşdılmalarımız düzgündür.

**server 127.0.0.1, stratum 3, offset -0.000037, delay 0.02594**

**27 Mar 22:25:19 ntpdate[33806]: adjust time server 127.0.0.1 offset -0.000037 sec**

/var/log/ntp.log faylında aşağıdakı sətirlərə uyğun sətirləri görməlisiniz:

**27 Mar 22:18:38 ntpd[33642]: synchronized to 116.193.170.30, stratum 2**

**27 Mar 22:25:21 ntpd[33642]: synchronized to 78.111.50.2, stratum 2**

Artıq NTP serverimizin IP ünvanını istənilən resurs üçün NTP server kimi istifadə edə bilərsiniz.

**Qeyd:** Əgər sistem vaxtinin bios-dan götürülməsini istəsək, "/etc/wall\_cmos\_clock" ünvanında fayl yaratmaq lazımdır.

**touch /etc/wall\_cmos\_clock && chmod 0444 /etc/wall\_cmos\_clock**

- Fayl yaradıb yetki veririk.

**adjkerntz -a**

- Daxili vaxtı CMOS BIOS-dan götürürük.

## Sistem tarixi və vaxt aralığı

**cp /usr/share/zoneinfo/Asia/Baku /etc/localtime**

- Yerli vaxt enliyini təyin edirik (yəni AZST).

|                                                           |                                                                                                    |
|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>ln -s /usr/share/zoneinfo/Asia/Baku /etc/localtime</b> | - Və ya link edirik.                                                                               |
| <b>date "+TIME: %H:%M:%S%nDATE: %Y-%m-%d"</b>             | - Vaxt və tarixi yeni sətirlərdə çap edirik.                                                       |
| <b>date "+The date today is %F."</b>                      | - Tarix çıkışına söz əlavə etmişik.                                                                |
| <b>date '+%A %B %d %G'</b>                                | - Günü, ayı, ayın günlərini və ili çap edəcək.                                                     |
| <b>date -v +2H</b>                                        | - 2 saat sonrakı vaxtı çap edəcək.                                                                 |
| <b>date -v +2m</b>                                        | - 2 ay sonrəni çap edəcək.                                                                         |
| <b>cal</b>                                                | - Hal-hazırkı ay üçün təqvimini çap edəcək.                                                        |
| <b>cal 2012</b>                                           | - 2012-ci il üçün təqvimini çap edəcək.                                                            |
| <b>cal-j</b>                                              | - Julian tipli təqvimini çap edəcək. Təqvim yanvar ayının 1-dən <b>365</b> -ci günədək hesablanır. |

# Müxtəlif məqsədlərdə istifadə edilən utilit-lərin açıqlanması

Bu başlığımızda müxtəlif istiqamətlərdə lazım ola biləcək imkanlar açıqlanır. Buna eyni fonda bir neçə prosesin işə salınması, CD/DVD ilə işləmə, şəkillər, səslə işləmək üçün program təminatları, mail göndərilməsi, adlandırılmış kanal, mətn fayl konvertasiyası, iş masasının yayıllanması və rootkit-lərin yoxlanılması aiddir.

## Screen

İmkanları:

1. Bir görünüşdə bir neçə dəfə giriş etmək imkanı - Bir işləyən prosesin üstündə PTY rəqəmlər generasiya edir.
2. İşləyən görünüş və prosesin qoşulub/ayrılması imkanı var.
3. TTY informasiyasını \$TTY dəyişənində saxlayır.
4. Screen-ləri saxlayır: '/var/run/screen/S-\$USER' - qovluğunda

İşimiz:

1. 'screen'-i yükleyək.
  - a. 'cd /usr/ports/sysutils/screen && make install clean'

**Qeyd:** Əgər siz istifadəçi adından giriş edib, sonra 'su' ilə root yetkisi alsanız, 'screen' PTY sessiyası yaratmayıcaq.

2. 'screen' istifadəsi

- a. '**screen -ls**' - Bu screen-ə mənimsənilən bütün proses listləri çap edir.(attached/detached)
  - a1. '**screen --help**' - screen-in imkanlarını çap edir.
- b. '**CTRL-a**' - Bütün SCREEN-lər arası keçid üçün istifadə olunur.
  - b1. '**CTRL+a ?**' - help qaytarır.
  - b2. '**CTRL+a Ctrl+d**' - screen-ə detach edir.
  - b3. '**CTRL+A, SHIFT+A**' - Set window's title to: (Yəni screen başlıq adı yaza bilərik.)
  - b4. '**screen -ls**'-cavabı '**930.ttyp0.dd**' - '**930'-processID, 'ttyp0'-terminalID, 'dd'-hostname**
  - b5. '**screen -r 930**' - '**930'-process ID-ni ekrana qaytarır.**
    - b5-1. '**screen -r**' - detach etdiyimiz screen-ə reconnect edirik.
  - b6. '**CTRL+a CTRL+C**' - eyni screen-in daxilində yeni screen yaradır.
  - b7. '**CTRL+a SCREEN\_NUM**' - '**'CTRL-a 0'**' - 0-ci screen-ə daxil olur.
  - b8. '**CTRL+a " "**' - screen-lərin sayını rəqəmləyərək çap edir.
    - b8-1. "**CTRL+A " "**" - (Switch to window:) və pəncərəni seçirik. Məsələn: 1-ci screen-ə keçirik.
    - b8-2. '**CTRL+A, SHIFT+N**' - yerləşdiyimiz screen-in nömrəsini çap edir. (This is window 5 (csh).)
    - b8-3. '**CTRL+A, n**' - Növbəti screen-ə keçid edir.
    - b8-4. '**CTRL+A, p**' - Öncəki screen-ə qayıdır.
    - b8-5. '**CTRL+A, SHIFT+A**' - Hal-hazırkı screen-in adını dəyişir.
  - b8-6. '**CTRL+A, w**' - Bütün screen-ləri ID və adla 1 sətirdə siyahilayacaq. Məs: "**0 Sonuncu pwd2\*\$screenlist3-\$ csh 4\$ csh 5\$ csh 6\$ Yəni Screen name**"
  - b9. '**CTRL+a, SHIFT+k**' - hal-hazırkı screen-i öldürür.
  - b10. '**screen -d -r**' - screen-i detach edin və hal-hazırkı səhifədə (PTY) yenidən attach edin.
    - b10-1. '**screen -S screenname**' - Burada screen-i işə salan zaman ona default ad yox, öz seçdiyimiz adı veririk.
    - b10-2. '**screen -x mysession**' - Burada mysession adlı sessiyani paylaşırıq, amma paylaşımından istifadə etmək üçün mütləq eyni istifadəçi adı ilə daxil olmaq lazımdır.
    - b10-3. '**screen -x**' - Paylaşım edilmiş screen-ə qoşuluruq.
  - b11. '**screen -x PID**' - seçilmiş processID ilə olan screen digər istifadəçilərə paylaşılır.

**Qeyd:** Paylaşım olmuş screen-i detach etmək üçün: '**CTRL-a d**' - Orijinal screen üçün narahat olmayıñ.

**Qeyd:** 'screen' - istifadə edin ki, yeni 'screen' aça biləsiniz. Həmin Screen-in öz iyerarxiyası olacaq.

**Qeyd:** 'screen' - o vaxt qalır ki, PTY-lar heç nə etmir.

## ISO nüsxələrin hazırlanması və yazılması

**Qeyd:** Sistem nüsxəsi yaradıb onu diskə birbaşa yazmaq üçün "**cdrtools**" paketi yüklemək lazımdır.

```
pkg_add -r cdrtools          - FreeBSD8.4 yükleyirik.  
pkg install cdrtools         - FreeBSD9.3,10.1 yükleyirik.
```

**mkisofs**-la rezerv nüsxə çıxarılır, **cdrecord**-la CD-yə və DVD-yə yazılır. '**mkisofs**' əmri "**crdtool**" yükəndikdən sonra əmələ gəlir.

```
cd /tmp  
mkisofs -o home.iso /home      - Adı ISO9660 nüsxəsi yaradaq.  
mkisofs -o home2.iso -J -R /home Add - Bu ISO fayl yarananda Rock Joilet Ridge olur.  
   Windows tərifindən görünə bilən.
```

```
mkisofs -o home3.iso -J -R -hfs /home Also add HFS extensions  
   - -hfs MAC tərifindən oxunula bilən ISO nüsxə.
```

Bir obrazın daxilinə bir neçə qovluq və onların fayllarını əlavə edə bilərik.

```
mkisofs -o home.iso -R -J music/ docs/ chris.pdf /var/spool/mail
```

ISO nüsxə yaradır, hansı ki, **"/var/pics/"** qovluğundan "**Pictures**" qovluğuna gedir. "**Pictures**" isə **"/home/chris"** qovluğundadır.

```
mkisofs -o home.iso -J -R -graft-points Pictures=/var/pics/ /home/chris
```

ISO nüsxə üçün header informasiya əlavə edir, **"-p"** ISO Image üçün ID identifikasiator, **"-publisher"**-də **128** simvolluq açıqlanma təyin eləmək olar, **"-V"** disk identifikasiatoru təyin edir. Disk

ID önemlidir, çünkü bəzi OS-lar bunu CD mount eləmək üçün istifadə edir, '-A' opsiyası göstərir ki, program sadəcə adı ISO yaradıb, '-volset' opsiyası isə ISO nüsxə haqqında cümlə çap edir.

```
mkisofs -o /tmp/home.iso -R -J -p www.handsonhistory.com -publisher "Swan Bay Folk Art Center" -V "WebBackup" -A "mkisofs" -volset "All web site material on November 2, 2008" /home/chris
```

```
isoinfo -d -i home.iso
```

- 'home.iso' iso haqqında informasiyanı çap edir.

```
mkdir /mnt/myimage
```

- Yaratdığımız iso nüsxəni müəyyən bir qovluğa mount eləmək üçün qovluq yaradırıq.

```
mdconfig -a -t vnode -f home.iso -u 0
```

- 0-ci unit-də memory disk yaradırıq və home.iso-nu ona mənimsədirik.

```
mount -t cd9660 /dev/md0 /mnt/myimage
```

- Mount edəndə deyirik ki, md0 aləti cd9660 tiplidir.

```
ls -l /mnt/myimage
```

- Və nüsxənin məzmununa baxırıq.

```
umount /mnt/myimage
```

- İşimiz bitəndən sonra umount edirik.

**Qeyd:** ISO nüsxəni diskə yazmaq üçün isə kernel-in modulundan istifadə edirik.  
"kldload /boot/kernel/atapicam.ko".

```
cdrecord -scanbus
```

- CD/DVD alətimizin ROM və ya RW olduğunu baxırıq.

```
cdrecord -blank=fast ...
```

- CDRW/DVDRW diskini tez rejimində silin.

```
cdrecord -dummy home.iso ...
```

- Diski yazmadan yazma imkanını test edin.

```
cdrecord -v home.iso ...
```

- Susmaya görə olan quraşdırılmalarla diskini yazın.

```
cdrecord -v -e d speed=24 home.iso...
```

- "home.iso" faylini 24 süreti ilə yazın.

```
cdrecord -pad home.iso ...
```

- CD/DVD yazılımı bitdiğdən sonra diskini çıxarıın.

```
cdrecord -eject home.iso
```

- Diski alətin adı ilə identifikasiya edin.

```
cdrecord -/dev/cdrw home.iso...
```

- Aləti SCSI ad ilə identifikasiya edin.

```
cdrecord dev=0,2,0 home.iso
```

- Multisessiyalı disk yazın.

```
cdrecord -multi home.iso
```

- Növbəti disk yazılmaması üçün sessiyanın çıkışını yoxlayın.

```
cdrecord -msinfo
```

Ya da hər iki işi bir utilit-lə görə bilərik. **growisofs**-lə.

|                                                    |                                                          |
|----------------------------------------------------|----------------------------------------------------------|
| <b>cd /usr/ports/sysutils/dvd+rw-tools</b>         | - Growisofs "dvd+rw-tools" paketinin tərkibinə daxildir. |
| <b>make install clean</b>                          | - Yükləyirik.                                            |
| <b>growisofs -Z /dev/cd0 -R -J /home/chris</b>     | - Yaradın və diskə yazın, multisessiyalı diskdir.        |
| <b>growisofs -Z /dev/cd0 -R -J /home/francois</b>  | - Diskə yeni ISO əlavə edin.                             |
| <b>growisofs -M /dev/cd0=/dev/zero</b>             | - Yazını dayandırın.                                     |
| <b>growisofs -dvd-compat -Z /dev/cd0=image.iso</b> | - ISO image-i DVD diskə yazın.                           |

## Audio avadanlığının qoşulması və fərqli utilitlər

|                                                                                       |                                                |
|---------------------------------------------------------------------------------------|------------------------------------------------|
| <b>dmesg   grep -i audio</b>                                                          | - Sistemə qoşulmuş audio alətlərimizi axtarıq. |
| <b>pcm0: &lt;AudioPCI ES1373-B&gt; port 0xdf00-0xdf3f irq 6 at device 7.0 on pcil</b> |                                                |
| <br>                                                                                  |                                                |
| <b>lspci   grep -i audio</b>                                                          | - PCI audio alətlərin siyahısına baxırıq.      |
| <b>01:07.0 Multimedia audio controller: Ensoniq ES1371 [AudioPCI-97] (rev 06)</b>     |                                                |

**Qeyd:** Sonra **sound\_card**-ı və **sound\_card\_driver**-i aktivləşdiririk. Bunun üçün önce **/boot/defaults/loader.conf**-dan aşağıdakı sətirləri götürürük və **/boot/loader.conf** faylinə əlavə edirik.

|                                                |                                                                                                                                                                                                                                                                                               |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>sound_load="YES" # Digital sound system</b> |                                                                                                                                                                                                                                                                                               |
| <b>  snd_es137x_load="YES" # es137x</b>        | - Əgər biz audio driver-i təyin edə bilməsək, "es137x"-in sətri əvəzinə dəstəklənən bütün driver-ləri yükleyə bilərik. Sistem yenidənyüklənməsindən sonra sistemdə " <b>/dev</b> " ünvanında " <b>dsp, dsp0.0, audio, audio0.0,mixer0</b> " və " <b>sndstat</b> " adlı alətlər əmələ gələcək. |
| <br>                                           |                                                                                                                                                                                                                                                                                               |
| <b>ll /dev/sndstat</b>                         | - Sistemdə yüklenmiş audio alətimizi görə bilərik.                                                                                                                                                                                                                                            |
| <b>chmod 666 /dev/acd0</b>                     | - Bütün istifadəçilərə CDROM-dan istifadə etmə yetkisini açırıq.                                                                                                                                                                                                                              |

**Qeyd:** Audio player kimi, bu paketlərdən "sox, vorbis-tools, libvorbis, mpg123, və aumix" birini yükleyə bilərik. Audio-nu kəsmək və enconding-i üçün "cdparanoia" və "flac" istifadə etmək olar.

```
pkg_add -r mpg321 flac  
pkg install mpg321 flac
```

- FreeBSD8.4
- FreeBSD9.3,10.1

```
mpg321 -longhelp  
mpg321 new.mp3 -w new.wav  
mpg321 *.mp3
```

- İmkanlarının tam siyahısını çap edir.
- Mp3 faylini konvert edirik wav faylinə.
- Yerləşdiyimiz qovluqda olan bütün MP3-lərə qulaq asın.

```
cat playlist | mpg321 -@ -  
mpg321 -z *.mp3  
mpg321 -Z *.mp3
```

- Playlist-i kanalla "mpg321"-ə ötürürük.
- Bütün mp3-ləri təsadüfi ardıcılıqla işlədir.
- Bütün mp3-ləri təsadüfi ardıcılıqla işə salın və həmişəlik təkrar edin.

```
pkg install vorbis-tools
```

- Paketlərdən, ya da portlardan yükleyirik.

```
cd /usr/ports/audio/vorbis-tools/  
make install clean
```

- ogg123 paketini yükleyirik.
- Yükleyirik.

```
ogg123 http://vorbis.com/music/Lumme-Badloop.ogg
```

- "ogg123"-u rəsmi saytında olan musiqi ilə yoxlayırıq. (Əla musiqidir).

```
ogg123 -z *.ogg
```

- Bütün ".ogg" faylları təsadüfi ardıcılıqla işə salırıq.

```
ogg123 -Z *.ogg
```

- Bütün ".ogg" faylları təsadüfi ardıcılıqla işə salırıq və sonsuz təkrarlanan edirik.

```
ogg123 /var/music/
```

- Olduğumuz qovluqdakı bütün "ogg" fayllarını işə salacaq.

```
ogg123 -@ playlist
```

- Musiqini "playlist"-dən işə salacaq.

**Qeyd:** Musiqilər playlist-dən, ya da qovluqda oxunduqda, xoşa gəlməyən musiqi arasında keçid etdikdə "Ctrl+C" əmri kifayətdir. İki dəfə "Ctrl+C" isə, ümumiyyətlə, çıxış deməkdir.

Səs və görüntünün səviyyələrini kontrol etmək üçün "mixer" və "aumix" paketlərindən istifadə olunur.

```
pkg install mixer
```

- Mixer paketini yükleyirik.

```
mixer
```

- Susmaya görə olan bütün səs səviyyələrini çap edir.

```
mixer vol -10
```

- Səsi 10% azaldaq.

```
mixer vol +20
```

- Səsi 20% artırıq.

```
mixer vol 80:60
```

- Sol səs dinamikinin səsini 80%, sağı isə 60% təyin edirik.

```
mixer =rec cd
```

- Yazmaq üçün aleti "cd" təyin edirik.

```
pkg install aumix
```

- aumix paketi həm screen rejimdə, həm də CLI rejimdə işləyir.

```
aumix -q
```

- Bütün sol sağ kanalların statusunu çap edir.

```
aumix -l q -m q
```

- "microfon" və "line" kanalların quraşdırılmalarını çap edir.

```
aumix -v 80 -m 0
```

- "mikrofon"-u sıfır və "volume"-u 80% edir.

```
aumix -m 80 -m R -m q
```

- "microfon"-un səsini 80% edib, record-a qoşun və mikrofonun statusunu çap edin.

**Qeyd:** CD-nin encoding olunması və bölünməsi üçün əla qrafik alet "grip" və "sound-juicer"-dir. Amma bununla belə, "oggenc", "flac" və "lame" adlı paketlər də var.

```
pkg install flac
```

- "flac"-i yükleyirik.

```
pkg install grip
```

- "grip"-i yükleyirik.

Ya da portlardan

```
cd /usr/ports/audio/lame
```

- Yükleyirik.

```
make install clean
```

**cdparanoia -B**

- Bütün track-ləri wav genişlənmədə öz adları ilə sərt diskə rip edəcək.

**cdparanoia -B -- "5-7"**

- 5-7 aralıqda olan track-ləri ayrı-ayrı fayllara yazacaq.

**cdparanoia -- "3-8" abc.wav**

- 3-dən 8-dək olan track-ləri **abc.wav** adlı bir fayla yazacaq.

**cdparanoia -- "1:[40]-"**

- İlk yolu 40-ci saniyədən sonadək ayır və yaz.

**cdparanoia -f -- "3"**

- 3-cü trackı "aiff" formatında yaz.

**cdparanoia -a -- "5"**

- 5-ci trackı AIFC formatında yaz.

**cdparanoia -w -- "1" my.wav**

- 1-ci yolu "my.wav" adı ilə yazın.

#### Musiqinin konvertasiyası

**oggenc ab.wav**

- wav formatdan ogg formatına konvert edəcək.

**oggenc ab.flac -o new.ogg**

- flac formatdan ogg formatına konvert edəcək.

**oggenc ab.wav -q 9**

- Konvert keyfiyyətini 9-la edin, ogg formatına.

Susmaya görə "ogg" formatının keyfiyyəti 3-lə olur. Ancaq biz "-1"-dən "10"-dək artırı bilərik.

**Qeyd:** Biz, həmçinin konvertasiyaya müəyyən qisim informasiya da əlavə edə bilərik.

**oggenc NewSong.wav -o NewSong.ogg \**  
**-a Bernstein -G Classical \**  
**-d 06/15/1972 -t "Simple Song" \**  
**-l "Bernsteins Mass" \**  
**-c info="From Kennedy Center" \**

- Konvert edirik wav-dan ogg-a  
- "-a" artistin adı, "-G" ümumi açıqlama,  
- "-d" tarix, "-t" başlıq,  
- "-l" albomun adı,  
- "-c" şərh informasiyası.

**ogginfo NewSong.ogg**

- "ogginfo" əmri "NewSong.ogg" musiqisi haqqında detallı informasiya çap edir.

**flac hotsong.wav -o hotsong.flac \**  
**--picture=cover.jpg**

- "wav"-i "flac"-a konvert et  
- və "cover.jpg" adlı şəkli tik.

**lame in.wav**

- "wav" faylından eyni ad ilə "mp3" faylinə konvert et.

```
lame tune.aiff -o tune.mp3  
lame -h -b 64 -m m in.wav out.mp3
```

- "aiff" faylından mp3-e konvert et.
- "wav" faylından yüksək keyfiyyətlə 64-bitlə mono modda konvert et.

```
lame -q 0 in.wav -o abcHQ.mp3
```

- "wav"-dan "mp3"-ə 0 keyfiyyətlə konvert et.

**Qeyd:** "lame"-də keyfiyyəti 0-la 9 arası təyin eləmək olar. Sısmaya görə 5-dir.

```
lame NewSong.wav NewSong.mp3 \  
--ta Bernstein --tg Classical \  
--ty 1972 --tt "Simple Song" \  
--tl "Bernsteins Mass" \  
--tc "From Kennedy Center"
```

- Konvert edirik wav-dan mp3-ə
- "--ta" artistin adı, "--tg" ümumi açıqlama,
- "--ty" tarix, "--tt" başlıq,
- "--tl" albomun adı,
- "--tc" şərh informasiyası.

### Sox Audio konverter

/usr/ports/audio/sox/ portlarda olan ünvanıdır. Sox VoIP-də istifadə olunan ən rahat utilit-dir, hansı ki, sayəsində səsləri konvertasiya edirik.

```
pkg install sox
```

- Yükləyirik.

```
sox head.wav tail.wav output.wav
```

- "head.wav" və "tail.wav" faylini "output.wav" faylında birləşdirir.

```
sox -m sound1.wav sound2.wav output.wav
```

- "sound1.wav" və "sound2.wav" faylini mix edib "output.wav" faylına yazır.

```
sox sound1.wav -e stat
```

- "sound1.wav" faylinin statusuna baxırıq.

```
sox sound1.wav output.wav trim 4
```

- "sound1.wav" faylini "output.wav" faylinə konvert edirik, lakin ilk 4 saniyəni yazmırıq.

```
sox sound1.wav output.wav trim 2 6
```

- "sound1.wav" faylini "output.wav" faylinə ancaq 2-ci və 6-ci saniyə aralığında konvert edin.

## Şəkillərlə işləyək

Şəkillərlə işləmək üçün "ImageMagick" paketi mövcuddur.

```
cd /usr/ports/graphics/ImageMagick
```

- Paketi yükleyirik (Bu paketin daxilində avtomatik olaraq, "convert" və "identify" utilit-ləri gələcək.)

```
make install clean
```

```
pkg install ImageMagick
```

- Ya da paketlərdən yükleyirik.

```
identify p2090142.jpg
```

- Şəkil haqqında məlumat çap edir.

```
identify -verbose p2090142.jpg | less
```

- Şəkil haqqında detallı məlumat çap edir.

**Qeyd:** "konvert" bu genişlənmələri(JPG, BMP, PCX, GIF, PNG, TIFF, XPM, XWD) konvert edə bilir.

**Qeyd:** Biz şəkilləri 0-360 dərəcə arasında bucaq dəyişməsi də edə bilərik.

```
convert tree.jpg tree.png
```

- "tree.jpg" formatından "tree.png" formatına konvert edəcək.

```
convert icon.gif icon.bmp
```

- "icon.gif" formatından "icon.bmp" formatına konvert edəcək.

```
convert photo.tiff photo.pcx
```

- "photo.tiff" formatından "photo.pcx" formatına konvert edəcək.

```
convert -resize 1024x768 hat.jpg hat-sm.jpg
```

- "hat.jpg" şəklinin ölçüsünü "1024X768" edib, "hat-sm.jpg" faylına konvert edir.

```
convert -sample 50%x50% dog.jpg dog-half.jpg
```

- "dog.jpg" şəklinin ölçüsünü uzunluq və endən 50% kiçildib "dog-half.jpg" faylına yazacaq.

```
convert -rotate 270 sky.jpg sky-final.jpg
```

- "sky.jpg" şəklini 270 dərəcə sağa fırladacaq və "sky-final.jpg" faylına yazacaq.

```
convert -rotate 90 house.jpg house-final.jpg
```

- "house.jpg" şəklini 90 dərəcə sağa döndərir və "house-final.jpg" faylına yazır.

Bu misalda "p10.jpg" faylından "p10-cp.jpg" faylına konvert etdikdə **60** fontsize-la və "helvetica" font tipində, mətn ölçüsü **10** və **80** diapazonunda bu mətni -> "Copyright NegusNet Inc." şəklə yazacaq.

```
convert -fill black -pointsize 60 -font helvetica -draw 'text 10,80  
"Copyright NegusNet Inc."' p10.jpg p10-cp.jpg
```

**convert -thumbnail 120x120 a.jpg a-a.png** - "a.jpg" şəklinin həcmini kiçildib "a-a.png" şəklinə "**120x120**" ölçüsündə yazırıq.

**convert -thumbnail 120x120 -frame 40x40 a.jpg a-b.png**  
- "a.jpg" şəklinin həcmini kiçildib "a-b.png" şəklinə "**120x120**" ölçüsündə və "**40x40**" çərçivəsində yazırıq.

"a.jpg" şəklinin həcmini kiçildib "a-c.png" şəklinə "**120x120**" ölçüsündə və "**40x40**" çərçivəsində edir və **10** dərəcə sağa döndərib arxa fonu qırmızı rəng edir.

```
convert -thumbnail 120x120 -frame 40x40 -rotate 10 -background red a.jpg a-  
c.png
```

**convert -sepia-tone 75% house.jpg oldhouse.png**  
- "house.jpg" faylini **75%-lik** tonla "sepia-tone" stili ilə "oldhouse.jpg" faylına konvert edəcək.

**convert -charcoal 5 house.jpg char-house.png**  
- Eyni ilə burada ağac oyuğu kimi konvert edəcək.

**convert -colorize 175 house.jpg color-house.png**  
- Eyni ilə burada da şəkil lenti kimi konvert edəcək.

**convert -swirl 300 photo.pcx weird.pcx** - "photo.pcx" şəklini "weird.pcx" faylına konvert edəndə su burulğanı yaradır.

## Mail göndərilməsi

Əməliyyat sistemimizdə susmaya görə SendMail işlək vəziyyətdə olur. Biz hansısa bir programımız və ya sistemə aid olan statistik məlumatları SendMail vasitəsilə email-lə yollaya bilərik. Ancaq göndərdiyiniz email ünvanı daxili olmalıdır. Serverimizin IP ünvanı heç bir email yoxlanışında olmalı deyil. Əks halda, göndərilən mail server tərəfindən tutulacaq və ünvana ötürülməyəcək.

```
uname -a | mail -s 'My BSD version' qebuleden@gmail.com
```

- Sistem versiyasını "My BSD version" subject-lə **qebuleden@gmail.com** email ünvanına yollayırıq.

```
ps aux | mail -s'My Processl List' qebuleden@gmail.com
```

- Sistem proseslərini "My Process List" subject-lə **qebuleden@gmail.com** email ünvanına yollayırıq.

```
echo $MAIL
```

```
mail
```

- Daxili istifadəçinin **MAIL** faylini çap edir.
  - Gələn mail-ləri çap edir.
  - >N** - Ən yeni email-i çap edir.
  - N** - Yeni email-ləri çap edir.
  - U** - Oxunulmamış email-ləri çap edir.
  - &** - Email-lər bitmiş və '**mailq**' növbəti əmrləri gözləyir.
  - ?** - '**mailq**' əmrinin imkanlarını çap edir.
  - v1** - 1-ci email-i vi rejimdə açacaq.
  - h1** - 1-ci email-dən başlayaraq bütün email-lərin header-lərini çap edəcək.
  - r1** - 1-ci email-ə cavab verəcək.
  - d2** - İkinci email-i silir.
  - d4-9** - 4-dən 9-dək bütün email-ləri siləcək.
  - !bash** - mail mühitindən bash mühitinə qayıdacaq.  
Sonra exit əmrini daxil etsək, yenə '**mail**'-ə qayıdacaq.
  - x** - Bütün mail-ləri mailbox-da saxlayaraq çıxacaq.
  - q** - Bütün mail-ləri yadda saxlayıb, sonra çıxacaq.
- 
- ```
mail -f /var/mail/cavid
```
- '**cavid**' istifadəçisinin mail-lərini root adından oxuyuruq.

Həmçinin gördüyüümüz bu işlərin hamısını mutt vasitəsilə edə bilərsiniz. Ancaq öncə onu portlardan, ya da paketlərdən yükləmək lazımdır. Port ünvanı:

**/usr/ports/chinese/mutt.**

**pkg install mutt**

- Yükləyirik.

-s subject, -a əlavədə olan PDF faylı, salamfile faylı isə email-in kontentidir.

**mutt -s "My BSD Version" -a /home/cavid/book.pdf cavid@192.168.0.105 < salamfile**

-s subject, -a attach '/dev/null' isə boş email kontentidir.

**mutt -s "My BSD Version" -a /home/cavid/book.pdf cavid@192.168.0.105 < /dev/null**

**mutt**

- Bu əmr dən sonra istifadəçinin öz istifadəçi məktublarına baxıb, mail-ləri idarə etmək olar.

**?**

- İmkanlar menyusunu açıqlayacaq.

## **Adlandırılmış kanal. FIFO (First in First Out)**

Istənilən proses adlandırılmış kanalı aça və bağlaya bilər. Kanalın hər bir tərəfində olan proseslər bir-birilə əlaqəli olmalı deyil. Yaradılmış kanalda mütləq oxuma və yazma hüququ olmalıdır. Çünkü bir tərəfdən proses yazır, o biri tərəfdən oxuyur.

**mkfifo fayl**

- "**mkfifo**" əmri "**fayl**" adı ilə adlandırılmış kanal yaradır. Yaradılmış yetkilərində '**p**' simvolu yaranır. Bu da pipefayl deməkdir.

**prw-r--r-- 1 root wheel 0B Mar 28 22:47 fayl**

Test edək.

Yaratdığımız fayl yerləşən serverdə iki ayrı sessiya ilə qoşulaq və fayl yerləşən qovluğa daxil olaq.

1. İlk seansda "**cat < fayl**" əmri daxil edirik.

Bu, o deməkdir ki, "**fayl**" faylından gələn konteyner cat əmrinə ötürülür.

2. İkinci seansda "**cat > fayl**" əmri daxil edirik.

Bu, o deməkdir ki, "**cat**" əmri fayla müəyyən konteyner ötürəcək. Və ardınca müəyyən sözlər yazıb **ENTER** sıxırıq. Sonra da ilk sessiyamızın ekranına baxsaq, həmin yazıları orada görə bilərik. İşi bitirmək üçün "**Ctrl+D**" əmrini daxil edirik.

"/dev/zero" və "/dev/null" fərqləri

**/dev/null**

- Tamam boşluqdur.

**/dev/zero**

- Spesifik fayldır, hansı ki, **null** simvollar yaradır.

**dd if=/dev/zero of=/home/nullfile count=1** - Burada "nullfayla" 1 dəfə 512 baytlıq informasiya yollayıraq.

**od -vt x1 /home/nullfile**

- **Od (Octal Dump)** 8-lük say sistemini açıb oxuyur. Faylimizi onunla oxuyacağıq.  
"-v" təkrarlanan sətirləri çap etməyin.  
"-t" çıxışının formatını təyin edin.  
"x1" nəticə hexadecimall olsun.

### Mətn faylinin müxtəlif formatlara konvert olunması

UNIX əməliyyat sistemlərində mətn faylları sətrin sonunu "\n" simvolu ilə bitirir. Windows/DOS isə sətrin sonuna "\r\n" simvolunu artırır. O isə müəyyən quraşdırılmaların işləməsində problem çıxarır.

**cp /etc/hosts file.txt**  
**od -c -t x1 file.txt**

- "**hosts**" faylini adı txt faylına nüsxələyək.  
- **Od (Octal Dump)** 8-lük say sistemini açıb oxuyur. Faylimizi onunla oxuyacağıq.  
"-c" Escape simvollarını C-stilində çap edin.  
Məsələn: "\n". "-t" çıxışının formatını təyin edin.  
"x1" nəticə hexadecimall olsun. Nəticədə, siz hexadecimall formatda "\n" simvollarını sətirlərin sonlarında görəcəksiniz.

UNIX2DOS-dan DOS2UNIX-ə konvert edək.

**pkg install unix2dos**

- "unix2dos" paketini internetdən yükləyək.

**unix2dos < file.txt > dosfile.txt**

- "**file.txt**" faylinin məzmununu yollayıraq  
"unix2dos" əmrinə, ondan alınan məzmunu isə "dosfile.txt" faylına ötürürük.

**cat dosfile.txt | dos2unix > unixfile.txt** - "dosfile.txt" məzmununu çap edib kanalla

yollayıraq "dos2unix" emrinə, o isə nəticəni "unixfile.txt" faylına yönəldir.

```
file dosfile.txt unixfile.txt
```

- Hər iki faylin tipinə baxırıq.

## IRC CHATS

FreeBSD əməliyyat sistemində dostlarla chat-laşmaq üçün çox client-lər var.

Məsələn: **xchat** (qrafik interfeysli), **ircII**, **PhoEniX**, **Vassago**, **BitchX**, **irssi**.

IRC proxy client-lər: **bouncher**, **dircproxy**

```
pkg install xchat
```

- Xchatı yükleyirik, ya da

```
pkg install irssi
```

- yükleyirik.

```
$ irssi -n Qabriel
```

- Seçdiyimiz nickname-lə daxil oluruq.

(Göy rəngli fon açılacaq.)

```
/connect chat.freenode.net
```

- **freenode.net**-in chat serverinə daxil oluruq.

```
/join #freebsd
```

- '#**freebsd**' qrupuna üzv olmağa çalışırıq.

```
Alt+1,Alt+2
```

- Irssi pəncərələri arasında keçid.

```
Ctrl+N,Ctrl+P
```

- Irssi pəncərələri arasında keçid.

```
/help
```

- Köməkçi menyu açılır.

```
/part
```

- Qrupdan çıkışdır.

```
/quit
```

- Chat-dan çıkış.

## Sharing Desktop Using VNC

**VNC** (Virtual Network Computing)

- Uzaq məsaflədən iş masasına yetki vermək üçün sistemdir. FreeBSD əməliyyat sistemimizin qrafik interfeysinə uzaqdan yetki almaq istəsək, bu alt bölümə mövzu açıqlanır.

```
pkg install vnc
```

- VNC-ni yükleyirik.

```
mkdir $HOME/.vnc
```

- Hər bir istifadəçinin ev qovluğunda '.vnc' qovluq yaradıb **vnc**-nin quraşdırılmasını saxlayaqq.

**vncpasswd**

- Daxil olduğumuz istifadəçi üçün vnc şifrəsi yaradırıq.

**VNC server**

- VNC server-i root consol-dan işə salırıq, aşağıdakı sətirlər çap olunur. Şifrə təyin edirik ki, iş masamızda daxil olduqda tələb olunsun. root-un ev qovluğunda '**Xauthority**' faylı, '**.vnc**' qovluğu və onun altında '**xstarttup**' adlı fayllar yaradır.

You will require a password to access your desktops.

Password:

Verify:

xauth: creating new authority file /root/.Xauthority

New 'boot.az:1 [salman]' desktop is boot.az:1

Creating default startup script /root/.vnc/xstartup

Starting applications specified in /root/.vnc/xstartup

Log file is /root/.vnc/boot.az:1.log

**Qeyd:** Əgər siz firewall istifadə edirsinizsə, nəzərə alın ki, VNC server **5900**-cu portdan başlayaraq, ardıcılıqla istifadə edir. Məsələn: **TCP** ilə **5900+N**, N-display-in nömrəsidir, Məsələn: **display 1 5901**-ci portda qulaq asacaq.

**Qeyd:** 'VNC server' əmri "**Xvnc**" prosesini o halda işə salır ki, ilk display-i mənimşəyib dinləyir.

**Qeyd:** '**\$HOME/.vnc**' qovluğunda vnc-nin '**passwd**' faylini görə bilərik, hansı ki, yaratdığımız şifrə bu faylda saxlanılır.

**VNC server -kill :1**

- VNC server-i **1-ci** display-də öldürürük.

**Qeyd:** 'VNC servere' Browser vasitəsilə və client-lə də qoşula bilərik. "<http://192.168.0.104:5801/>" Web-lə qoşuluruq.

**vncviewer 192.168.0.104:1**

- Müştəri programı vasitəsi ilə adı istifadəçi ilə **192.168.0.104** IP ünvanlı serverin **1-ci** display-inə qoşuluruq.

Desktop-un tam işləməsi üçün VNC serveri işə salan istifadəçi öz ev qovluğunda '**xstartup**' adlı fayl yaratmalıdır.

Biz öz iş masamızı VINO ilə də paylaşa bilərik.

**pkg install vino**

- VINO paketini yükleyirik.

**Qeyd:** VINO serverin tələb etdiyi şifrə minimum **8** simvol olmalıdır.

**vino-preferences**

- Bu əmri VNC serveri işə salacaq istifadəçinin CLI-ndən daxil edirik. Unutmayın ki, istifadəçi də **X11-forward** aktiv olmalıdır. Qrafik rejimdə VNC serveri quraşdırırıq.

## CHKROOTKIT quraşdırması

**RootKit** – proqramlar yığımıdır, hansı ki, sayəsində obyektləri(driver, fayl, proses və qovluq) gizlətmək mümkün olur. Sistemdə idarə etmə və informasiya yığımı ilə məşğul olur. Adətən Hacker-lər bunu sistem üzərində root yetkisini aldığı andan etibarən yükləyirlər. Bir sözlə, rootkit Hacker-in izlərini təmizləmək və gizlətmək üçün alətdir. chkrootkit isə onu araşdırıb üzə çıxarmaq üçün alətdir.

**cd /usr/ports/security/chkrootkit  
make install clean**

- ChkRootKit-i yükleyirik.  
- Yükləyirik.

**chkrootkit**

- Yoxlanış edirik.

Burada susmaya görə olan parametrlərlə işə salınır. Başlanğıc üçün tam kifayətdir.

Bütün imkanlar siyahısını **-h** opsiyası ilə əldə edə bilərsiniz. Skan etdikdən sonra aşağıdakı prefiksrlə detallı hesabat çap olunur:

**not infected**

- Heç bir rootkit program təminatının tapılmadığını göstərir.

**INFECTED**

- Yoxlanış göstərir ki, bu program pis niyyətli kod siyahısına daxildir.

**not\_found**

- Proqram tapılmadı, uyğun olaraq yoxlanışdan keçmədi.

**Vulnerable but disabled**

- Proqram zədəlidir, amma istifadə edilmir.

**not tested**

- Test yerinə yetirilmədi (opsiyadan söndürülmüşdür, ya da mümkün deyil).

**Qeyd:** Sonda silməyi unutmayın. Çünkü Hacker özü belə bu utilit-dən istifadə edib öz alətlərini sile bilər.

Link: [www.chkrootkit.org](http://www.chkrootkit.org)

Link: <http://ramzess.ru/bezopasnost-servera-freebsd-proverka-na-rootkity-chast-i/>

# BÖLÜM 9

## PF, İPFW, IPFilter FireWall-ların açıllanması, İPFW- PF Fail2Ban

- / FreeBSD Firewall
- / PF FIREWALL
- / PF ALTQ tam açıqlama
- / PF Bridge Firewall
- / IPFW Firewall
- / IPFW Squid Transparent
- / IPFW PF Fail2Ban
- / IPFilter firewall

Başlığımızda FreeBSD əməliyyat sisteminin dəstəklədiyi fərqli tip firewall-lar açılınır. Firewall vasitəsilə istifadəcidən və dünyadan gələn istənilən trafik kontrol edilir. İstifadəcidən gələn trafikin ikinci və üçüncü səviyyədə kontrolu açılınır. İkinci səviyyəli firewall quraşdırılır və izah olunur. İPFW vasitəsilə SSH daemona edilən hücumların avtomatik bağlanması quraşdırılır. Korporativ istifadə üçün Squid daemon işə salınır və İPFW firewall-la transparent rejimdə integrasiya edilir. Həmçinin IPFilter firewall imkanları göstərilir.

# FreeBSD Firewall

Firewall – təhlükəsizlik məqsədləri üçün istifadə edilir və imkanları aşağıdakılardır:

- Daxili şəbəkədə olan programları, servisləri və maşınları Internet tərəfindən gələn arzu olunmayan trafikdən qoruyur.
- Daxili şəbəkədə olan hostlardan internet üzərində olan servislərə çıxışı limitləyir və ya dayandırır.
- NAT (Network Address Translation) dəstəkləyir. Sayəsində daxili IP, IP qrupunu və ya tam şəbəkəni dünyaya yeganə PUBLIC IP ünvanla yayımlamaq olur.

Firewall tam şəkildə **TCP/UDP** üzərində trafiki kontrol edir. Protokollar haqqında ətraflı oxumaq üçün <http://www.ipprimer.com/#/tcpudp> linkinə müraciət edə bilərsiniz. Hər bir TCP/IP servisi təyin edildiyi protokoldan və qulaq asdığı portdan ibarətdir. Tanımadığınız portların siyahısını **/etc/services** faylından, ya da [http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers) linkindən əldə edə bilərsiniz. Troyanların istifadə etdiyi portların siyahısını bu <http://www.sans.org/security-resources/idfaq/oddports.php> linkdən əldə edə bilərsiniz.

FTP server iki rejimdə işləyir: aktiv və passiv. Fərqləri data kanalının necə əldə olunmasından ibarətdir. Passiv rejim daha təhlükəsizdir, ona görə ki, verilənlərin ötürülməsi kanalı səliqə ilə hər dəfə FTP seansdan sorușularaq əldə olunur. FTP rejimləri haqqında ətraflı oxumaq istəsəniz, <http://www.slacksite.com/other/ftp.html> linkinə müraciət edə bilərsiniz.

- **Aktiv** FTP rejimində istifadəçi təsadüfi yetkisi olmayan (**N > 1023**) portla FTP serverin əmr qəbul edən portu **21**-ə qoşulur. Sonra istifadəçi **N+1** portunda qulaq asmağa başlayır və FTP serverə əmr ötürür **PORT N+1**. Sonra isə server istifadəçi təyin etdiyi data portuna öz data portu **20** ilə qoşulur.
- Səliqə ilə FTP serverin istifadəciyə qoşulmasının yaradılması üçün yeni FTP metod olan **Passiv** yaradıldı. Buna **PASV** də deyilir (Müştəri tərəfindən bu əmr yerinə yetirilən kimi serverə deyilir ki, passiv rejimə keçid alınsın). Passiv rejimdə isə serverə gedən qoşulmanın hər ikisini istifadəçi özü edir. Müştəri FTP qoşulması etdikdə özündə iki fərqli yetkisi olmayan port (**N>1023** və **N+1**) açır. İlk port FTP serverin **21**-ci portu ilə əlaqələnir, ancaq **PORT** əmrinin yerinə yetirilməsi və serverə geriye bu data porta qoşulması əvəzinə, istifadəçi **PASV** əmrini yerinə yetirəcək. Bunun nəticəsinə FTP server qeydiyyatda olmayan portlardan təsadüfi (**P>1023**) seçərək birini açır və **P**-ni geriye istifadəciyə PSV əmrinin cavabı olaraq ötürür. Ardınca istifadəçi **N+1** portundan FTP serverin **P** portuna data portuna data ötürmək üçün qoşulma yaradır.

Firewall "**exclusive**", ya da "**inclusive**" ola bilər. Exclusive - firewall qaydaları ilə üst-üstə düşən trafikdən başqa, digərlərini susmaya görə buraxır. Inclusive - əksinə, firewall qaydaları ilə üst-üstə düşən trafikləri buraxır, digərlərini susmaya görə bağlayır.

Təhlükəsizlik "**stateful firewall**" sayesində sıxılı bilər. Bu tip firewall açıq qoşulmaların sessiyasını saxlayır və yalnız mövcud qoşulmaya uyğun olan trafikə izin verir, ya da yeni izin verilmiş qoşulma açır. Stateful sessiyada olan axını **bi-directional** (iki istiqamətli) olaraq süzgəcdən keçirir. Sessiyada hər bir gözlənilən paket statusu firewall-umuzun qaydasına uyğun olarsa, avtomatik olaraq daxili dinamik qayda generasiya ediləcək. O, kifayət qədər imkana malikdir ki, paketin keçərli sessiyaya aid olub-olmamasını təyin edə bilsin. Sessiya nüsxəsinə aid olmayan istənilən qalan paketlər blok ediləcək. Seans bitən kimi dinamik cədvəldəki qayda silinir.

FreeBSD əməliyyat sisteminin daxilində **PF**, **IPFW** və **IPfɪltə**(IPF) tipli firewalllar mövcuddur. Keçirilmənin idarəsi üçün isə iki trafik boğucu (shaper)-dan istifadə edir. Bunlar **altq(4)** və **dummynet(4)**-dir. Ənənəyə uyğun olaraq, **altq** tam şəkildə **PF** və dummynet isə **IPFW** ilə bağlıdır.

## Packet filtering

- Daxil edilən - Qaydalarda olan bütün trafiki buraxır, qalanlarını tutur.
- Çoxşaxəli - Qaydalarda olmayan bütün trafiki qəbul edir və başqa üsulla təyin edir.

İşimiz:

1. PF-i qurmaq. Haqqında daha ətraflı oxumaq istəsəniz, <http://home.nuug.no/~peter/pf/> və <http://pf4freebsd.love2party.net/#resources> linklərindən oxuya bilərsiniz.

**Qeyd:** PF susmaya görə kernel-də yüklenən deyil.

**Qeyd:** PF (`kldload`) modullardan yüklenilə bilər.

**Qeyd:** İstifadə edin: 'kldstat' yüklenmiş modulları görmək üçün.

- a. `'ee /etc/rc.conf'` StartUP faylına əlavə edirik -> '`pf_enable="YES"`'
- b. Nümunə faylı `'/usr/share/examples/pf/pf.conf'` nüsxələyirik '`/etc/pf.conf`' faylına.

2. `'/etc/pf.conf'` faylında əsas dəyişənləri dəyişdirin.

a. Əsasən, şəbəkə kartı adları, IP ünvanlar və bizə lazım olan portlardır.

b. `IPv{4|6}` ünvanların cədvəli aşağıdakı kimi qurulur: '{ }'

Məsələn: `'table <home-net> [const|persist] {192.168.1.4, 192.168.1.7}'`

`'const'` - Cədvəl dəyişilməz kimi təyin olunacaq, `'pfctl'` onu dəyişdirə bilməz.

`'persist'` - Bu cədvəl qaydalardan kənardır, əgər kernel onun istifadə olunmadığını gərsə, siləcək. `'table <home_net> persist {192.168.1.0/24, 10.0.0.0/24}'`

c. **Opsiylar** - PF-in təsiri, PF-in qlobal quraşdırılmaları, `'Block Policy'`

**Qeyd:** Əgər qadağa üçün açıq-aydın qayda yoxdursa, susmaya görə olan '**block policy**' trafiki buraxır.

**Qeyd:** Son yazıya uyğun olan trafiki PF tapır, əgər '**quick**' istifadə olunmursa.

d. Növbələşmə (**Queueing**) - **QoS**

e. Filtr qaydaları.

3. Qaydaların yüklenməsini test edək.

a. `'pfctl -vnf /etc/pf.conf'`

**Qeyd:** Susmaya görə PF bütün qaydalara flag təyin edir: '`flags S/SA keep state`'

4. Əgər qaydalar uğurludursa, PF-i işə salaq.

a. `'/etc/rc.d/pf start'` - əmr kernel modulunu yükleyəcək - `'kldstat'`-la baxa bilərik.

b. `'pfctl -sa'` - Bütün PF informasiyasını çap edir, yüklenmiş cədvəllər, filtrlər və saygaclar daxil olmaqla.

**Qeyd:** '`pass out all`' - Sətir statusu yoxlayaraq çıkış paketlərinə icazə verir.

**Qeyd:** 'pfctl -f /etc/pf.conf' - qayda dəyişikliklərini fayldan oxuyub yenidənyüklənmə edir.

## 5. 'pfctl'

- a. '-d' - PF-i söndürür.
- b. '-e' - PF-i işə salır.
- c. '-ss' - Cədvəlin statusunu çap edir.
- d. '-sr' - Hal-hazırkı qaydaları çap edir.
- e. '-sn' - NAT haqqındaki informasiyanı çap edir.
- f. '-t home\_net -T show' - <home\_net> cədvəlin məzmununu çap edir.
- g. '-t home\_net -T delete 10.0.0.0/24' - <home\_net> cədvəldən göstərilən şəbəkəni silir.
- h. '-t home\_net -T add 10.0.0.0/24' - <home\_net> cədvəlinə göstərilən şəbəkəni əlavə edir.

**Qeyd:** Dinamik 'pfctl' vasitəsi ilə əlavə edilən qaydalar yalnız RAM-da qalır. '/etc/pfconf' -u reload eləsəniz, o silinəcək.

- i. '-vnf /etc/pf.conf' - sintaksisi yoxlayır.

## 6. PF qaydalar:

- a. 'action' = 'pass', 'drop'
- b. 'direction' = 'in'(from external host), 'out' (locally-generated)
- c. '[log]' - mənası 'pflogd', əgər uyğun qayda aktivdirse.
- d. '[quick]' - hal-hazırkı qaydanı son qayda kimi qəbul edir.

**Qeyd:** PF susmaya görə son üst-üstə düşən paketi uyğun olaraq qəbul edir.

- e. 'on interface' - hal-hazırkı paketin yeri - 'on \$int0'
- f. 'proto protocol' - 'tcp|udp|icmp/etc/protocols/protocol number(0-255)|list'
- g. 'from src\_addr [port src\_port]'
- h. 'to dst\_addr [port dst\_port]'
- i. '[flags tcp\_flags]' - susmaya görə = 'S/SA', və ACK
- j. '[state]' - [no state|keep state(default)|modulate state|synproxy state]

## 7. Permit ALL from <home-net> EXPERT - '192.168.1.101'

- a. Yenilə cədvəli üçün: '!192.168.1.101'

8. MACRO istifadə edərək girişə olan portlara hədd qoyaq.
  - a. `'trusted_ports={22,80}'`

İşimiz:

1. Girişdə ICMP(6)-ni bloklayaq.
  - a. `'block in proto icmp'` - Girişə bütün 'icmp' paketləri bağlayır.
  - b. `'block in proto icmp6'` - Girişə bütün 'icmp6' paketləri bağlayır.
  - c. `'block in proto {icmp icmp6}'` - icmp icmp6-ni girişə bağlayır.
2. Çıxışa ICMP[6]-ni bloklayaq.
  - a. `'block out proto {icmp icmp6}'` - 2 qaydanı birində yazdıq.
3. UDP trafikə hədd qoyaq.
  - a. `'netstat -an -p udp'` – serverimizin qulaq aslığı udp portları sadalayaq.
  - b. `'trusted_udp_ports={53, 123, 514}'`
  - c. `'pass in on $int0 proto udp to any port $trusted_udp_ports'` - girişə \$int0 şəbəkə kartına istənilən yerə göstərilən qrupda olan portlara icazə var.
4. Global Block yaradaq TCP|UDP|ICMP.
5. İnzibatçılar üçün ayrı qayda yazaq.
  - a. `'pass in all from <admin_hosts>'` - İnzibatçılar üçün girişə hər şey açıqdır.
  - b. `'pfctl -t admin_hosts -T add 192.168.1.10'` - `admin_hosts` cədvəlinə yeni inzibatçı IP ünvanı əlavə edirik.

**Qeyd:** Bu yalnız RAM-da qalır, PF-in yenidənyüklənməsindən sonra itəcək. Hər bir yeni qaydanı mütləq '`/etc/pf.conf`' faylinə da əlavə edin.

### PF Log

İmkanları:

1. Qaydaların hansısa birinin üstüne düşərsə, jurnallaşdırır.
2. TCPDUMP-a uyğun olan paketləri '`/var/log/pflog`' faylinə generasiya edir.

**Qeyd:** '`newsyslog`' auto-rotate edir '`/var/log/pflog`'-u

3. Qurum Firewall səviyyəsində packet-level jurnallaşmanı dəstəkləyir.
4. Kernel-in modulu kimi qurulub: '`kldstat`'-i istifadə edərək görə bilərik.

5. '/var/log/pflog' - real rejimdə istənilən snifferlə yoxlanıla bilər. Məsələn: 'wireshark', 'tcpdump'

İşimiz:

1. Jurnallaşmanı aktivləşdirək: '/etc/rc.conf'-da
  - a. 'pflog\_enable="YES"'
  - b. 'pflog\_logfile="yeni\_log\_faylin\_ünvanı"'
  - c. 'pflog\_flags=" "' - standart pflog fayl opsiyaları əlavə etmək olar.
2. Log Traffic
  - a. 'block|pass| in|out [log] [quick]' – Qaydalara üçüncü sütun kimi qoşmalıdır.

**Qeyd:** Əmin olun ki, siz seçdiyiniz qaydanı jurnallaşma edirsiniz.

**Qeyd:** Susmaya görə olan flush vaxt 60 saniyədir, yəni '/var/log/pflog' jurnal faylı hər dəqiqədən bir yenilənir.

- b. 'tcpdump -v -Ae -r /var/log/pflog BPF' - tcpdump əmri jurnal faylini görünən sintaksislə oxuyur.

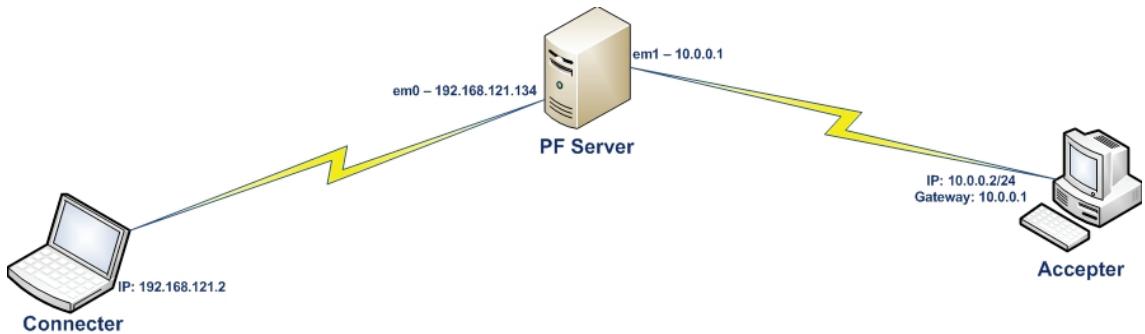
**Qeyd:** Həmçinin açıqlaya bilər: 'wireshark /var/log/pflog'

**Qeyd:** 'log all' - bütün 'syn, syn:ack' paketlərini jurnal edəcək.

- c. 'log (all)...' - yazdığımız qaydalardan keçən bütün paketləri jurnallayacaq.

# PF Firewall

Məqsədimiz PF vasitəsilə portun yönləndirilməsidir. Quruluş aşağıdakı şəkildəki kimi olacaq:



Şəkildə **Acceptor** adlı maşında **Windows7**-dir və **RDP(3389)** portu qulaq asır. **PF server** maşında isə qayda yazmışıq ki, **192.168.121.0/24** şəbəkəsindən **3389**-cu port ilə gələn istənilən müraciəti **10.0.0.2** IP ünvanlı serverə yönləndirsin.

PF-in işləməsi üçün kernel aşağıdakı siyahı ilə kompilyasiya edilməlidir:

```
device      pf          # PF virtual alət
device      pflog       # PFlog virtual alət
device      pfsync      # PFSync virtual alət
options     ALTQ        # Növbələrin idarəedilməsi üçün alternativ platforma
options     ALTQ_CBQ    # Klaslara əsaslanan növbələşmə (CBQ)
options     ALTQ_RED    # Təsadüfi erkən təyinat (RED)
options     ALTQ_RIO    # RED In/Out
options     ALTQ_HFSC   # İerarxik paket zamanlayıcısı (HFSC)
options     ALTQ_PRIQ   # Üstünlük növbələşməsi (PRIQ)
```

İşə salmaq üçün aşağıdakı sətirləri **/etc/rc.conf** StartUP faylına əlavə edirik:

**pf\_enable="YES"**

**pf\_rules="/etc/pf.conf"**

**pflog\_enable="YES"**

**pflog\_logfile="/var/log/pflog"**

- PF öz qaydalarını fayldan oxuyacaq.

**tcpdump -n -e -ttt -r /var/log/pflog**

- Pflog faylı oxuyuruq.

**tcpdump -n -e -tttt -v -r /var/log/pflog**

- Detallı açıqlama rejimində jurnal faylini oxuyuruq.

**/etc/pf.conf** faylinin içində aşağıdakı sətirləri əlavə edirik:

**#pfctl -d # PF-i dayandırır.**

**#pfctl -e # PF-i işə salır.**

**#pfctl -F all -f /etc/pf.conf**

- Bütün qaydaları sıfırlayır.

**#pfctl -s [ rules | nat | state ]**

- Filtr, nat qaydaların hesabatı,

**#pfctl -vnf /etc/pf.conf**

- **/etc/pf.conf** faylı səhvlerini yoxlayır, ancaq qaydaları mənimsətməyəcək.

**ports = "{ 20,21,3389 }"**

- Ports adlı qrup yaradırıq və lazımi portları əlavə edirik.

**table <clientler> { 192.168.121.0/24 }** - Client-lər adlı hansısa bir cədvəl yaradırıq və bize lazım olan şəbəkəni həmin cədvələ əlavə edirik.

**rdr on bcel inet proto tcp from <clientler> to 192.168.121.134 port \$ports -> 10.0.0.2**

- Bu sintaksisə deyirik ki, **bcel** şəbəkə kartı üzərindən TCP protokolu ilə client-lər cədvəlindən 192.168.11.134 IP ünvanına **\$ports** qrupunda olan portlardan birilə qoşulduğda 10.0.0.2 IP ünvanına yönləndirsin.

**pass in quick all**

- Girişə bütün trafikə izin veririk.

**pass out quick all**

- Çıxışa bütün trafikə izin veririk.

**pfctl -F all -f /etc/pf.conf**

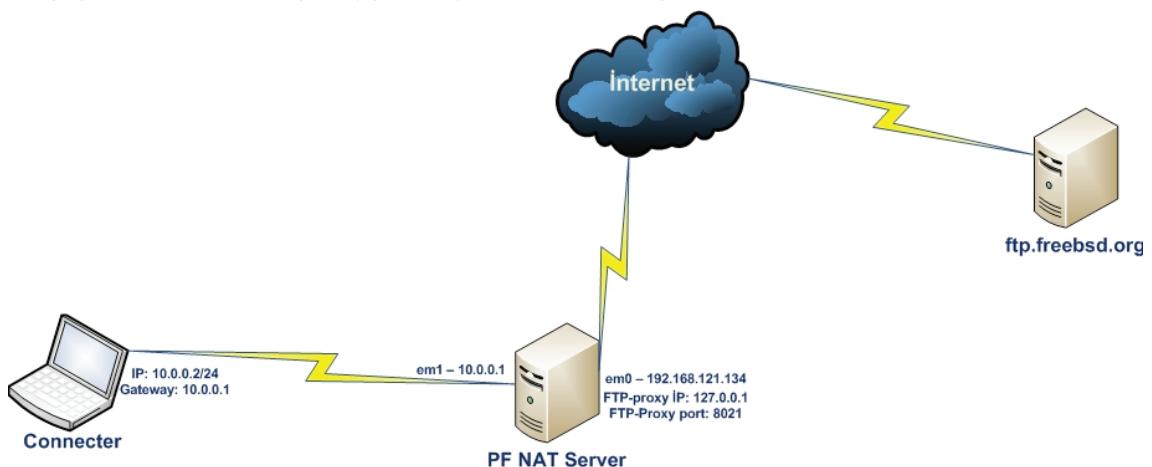
- Qaydaları sıfırlayıraq və işə salırıq.

Sonda Connecter adlı maşindan RDP ilə 192.168.121.134 IP ünvanına qoşulub Acceptor maşınının üstünə düşməliyik.

**Qeyd:** Əgər biz ftp port-u başqa host-a yönəlsək, unutmayaq ki, passiv rejimdə dinamik portlar istifadə olunacaq. Bunun üçün ftp-proxy istifadə ediləcək.

### PF FTP-Proxy

FTP-Proxy Aktiv/Passiv FTP qoşulmalarının çıkışına izin verir. Məqsədimiz PF NAT server arxasında olan daxili şəbəkə istifadəçilərinin dünya FTP serverlarına problemsiz çıxışlarını təmin etməkdir. Bunun üçün PF NAT qurulmuş serverimizdə ftp-proxy servisini aktiv etmək lazımdır. FTP-proxy servisi **127.0.0.1** IP ünvanı üzərində **8021**-ci portda qulaq asır. LAN şəbəkəmizdən istənilən istifadəçidən dünyada olan hansısa FTP (**ftp.freebsd.org**) serverə müraciət gedərsə, o, FTP-proxy üzərindən dünyaya çıkış edəcək. Topologiya aşağıdakı kimi olacaq:



Bu quruluşda **/etc/rc.conf** quraşdırma faylımız aşağıdakı kimi olacaq:

**hostname="ps"**

**ifconfig\_em0="inet 192.168.121.134 netmask 255.255.255.0"**

**ifconfig\_em1="inet 10.0.0.1 netmask 255.255.255.0"**

**defaultrouter="192.168.121.2"**

**gateway\_enable="YES"**

**sshd\_enable="YES"**

**pf\_enable="YES"**

**pf\_rules="/etc/pf.conf"**

**pflog\_enable="YES"**

**pflog\_logfile="/var/log/pflog"**

**ftpproxy\_enable="YES"**

**/etc/rc.d/ftp-proxy start**

- FTP-proxy-ni işə salırıq.

- FTPProxy servisini işə salırıq.

```

/etc/pf.conf firewall-umuzun quraşdırma faylı aşağıdakı kimi olacaq:
ext_if = "em0" # Internet
int_if = "em1" # lan
proxy="127.0.0.1"      # ftp proxy IP
proxyport="8021"        # ftp proxy port

##### Normallaşdırma
scrub in all

##### NAT və RDR işə salırıq.
nat-anchor "ftp-proxy/*"
nat on $ext_if from $int_if:network to any -> ($ext_if)

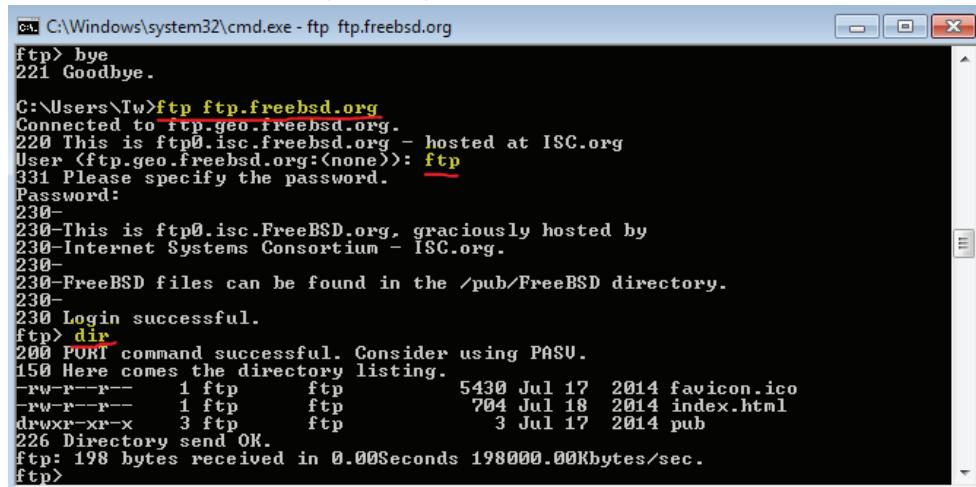
# ftp trafiki proxy-nin üstünə yönlendiririk.
rdr-anchor "ftp-proxy/*"
rdr pass proto tcp from any to any port ftp -> $proxy port $proxyport

# ftp-proxy üçün buna ehtiyacımız var.
anchor "ftp-proxy/*"
pass in quick all
pass out quick all
pfctl -f /etc/pf.conf

```

- Yeni qaydaları işə salırıq.

Sonda client maşından qoşuluruq:



```

C:\Windows\system32\cmd.exe - ftp ftp.freebsd.org
ftp> bye
221 Goodbye.

C:\Users\Tu>ftp ftp.freebsd.org
Connected to ftp.geo.freebsd.org.
220 This is ftp0.isc.freebsd.org - hosted at ISC.org
User <ftp.geo.freebsd.org:<none>>: ftp
331 Please specify the password.
Password:
230-
230-This is ftp0.isc.FreeBSD.org, graciously hosted by
230-Internet Systems Consortium - ISC.org.
230-
230-FreeBSD files can be found in the /pub/FreeBSD directory.
230-
230 Login successful.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 ftp      ftp          5430 Jul 17  2014 favicon.ico
-rw-r--r-- 1 ftp      ftp          704 Jul 18  2014 index.html
drwxr-xr-x  3 ftp      ftp           3 Jul 17  2014 pub
226 Directory send OK.
ftp: 198 bytes received in 0.00Seconds 198000.00Kbytes/sec.
ftp>

```

### TCPDump (Şəbəkədə olan problemlərin araşdırılması)

TCPDump spesifik alətdir, hansı ki, sayəsində şəbəkənin bütün səviyyələrinə görə trafiki açıb analiz edə bilərik. Serverimizin məsuliyyətində olan bütün şəbəkə səviyyələrinə görə, problemimizin araşdırılması və təyin edilməsində susmaya görə mövcud olan əvəz olunmaz alətdir.

**tcpdump -D**

- Hansı şəbəkə kartlarından istifadə edə biləcəyini göstərir.

**tcpdump -v**

- Daha detallı məlumatlarla ilk gördüyü şəbəkə kartında SNIFF edir. Məsələn: TTL, identifikasiya, ümumi uzunluq və IP paketində olan opsiyalar da hər sətirdə çap ediləcək.  
- **vvv** lap detallı opsiyalarla çap edəcək.

**tcpdump -v -w tcp.out**

- Bütün gələn paketleri detallı şəkildə **sniff** edin və **tcp.out** adlı fayla yazın.

**tcpdump -v -c 5**

- **5** paketi sniff edin və çıxın.

**tcpdump -i em0 | lo0 |em1**

- İstədiyimiz şəbəkə kartı ilə sniff edək.  
- Nəticədə çıxan paketlərdə adı IP ünvanına çevirməyəcək.

**tcpdump -r tcp.out**

- Öncədən sniff edilib **tcp.out** faylinə yazılmış məlumatı oxuyuruq.

**tcpdump -w tcp.out src host 10.0.1.7**

- **tcp.out** faylinə yalnız **10.0.1.7** IP ünvanından gələn paketleri sniff edin.

**tcpdump -w tcp.out src host 10.0.1.7 and dst port 22**

- **tcp.out** faylinə yalnız **10.0.1.7** IP ünvanından **22-ci** port ilə gələn paketləri sniff edib yazacaq.  
- Kanal səviyyəsində verilənləri çap edir (Məsələn: MAC ünvan).

**tcpdump -n -e -i em0 not port 22 and not port 137**

- **em0** şəbəkə kartımızda **22-ci** və **137-ci** portdan başqa bütün portları sniff edəcək.

**tcpdump -s 65535**

- Qəbul ediləcək paketlərin həcmi 65535 bayt edirik.

TCP flag-lar ilə filtrasıyanın geniş imkanları mövcuddur. TCP paketin başlığı **20** oktetdən ibarətdir. TCP-də flag-ların idarə edilməsi sütunu biti **13**-cü oktetin altındadır. Sözsüz ki, **13**-cü baytda flag-ların altında növbəti bit mövqeləri aparılır (Artıma görə **2-ci seviyyə**): **FIN(1)**, **SYN(2)**, **RST(4)**, **PSH(8)**, **ACK(16)**, **URG(32)**, **ECE(64)**, **CWR(128)**

Beləliklə, məsələn, SYN flag-ı ilə 13-cü başlıqda olan TCP paketlər üçün mənə **2** olacaq. Filtr isə aşağıdakı kimi olacaq:

```
tcpdump -n -i em0 'tcp[13] == 2'
```

SYN və ACK(Bit təyinatları 2 və 16-dır) flag-lı paketlərin tutulması üçün aşağıdakı quruluş istifadə ediləcək:

```
tcpdump -n -i em0 'tcp[13] == 18'
```

Bundan əlavə, **tcpflags** opsiyası istifadə edilə bilər. İstənilən flag-ı olan paket:

```
tcpdump -n -i em0 'tcp[tcpflags] != 0'
```

SYN flag-ı olan paketlər:

```
tcpdump -n -i em0 'tcp[tcpflags] & tcp-syn != 0'
```

TCP protokolu ilə ötürülən verilənləri analiz etmək üçün **tcpflow** istifadə etmək olar (ancaq öncə yüklemək lazımdır):

```
pkg install tcplflow
```

- Yükləyirik.

```
tcpflow -c -i em0 host 195.12.69.199 and port 80
```

Şəbəkədə ən çox qarşılıqlı əlaqədə olan istifadəçilərin reytinqini əldə etmək üçün aşağıdakı əmrədən istifadə edə bilərik:

```
tcpdump -tn -c 10000 -i em0 tcp or udp | awk -F ":" '{print $1"."$2"."$3"."$4}' | \ sort | uniq -c | sort -nr | awk '$1 > 100'
```

SYN+ACK-la daxil olanların sayına SYN flag-la çıxan paketlərin sayının münasibətinin(əlaqəsinin) hesablanması yolu ilə TCP anomaliyalarının üzə çıxardılması:

```
SYN_ONLY_FROM_ISP=`tcpdump -n -l -r $1 'src net 192.168.0.0/24' and 'dst net not 192.168.0.0/24' and 'tcp[13] == 2' | wc | awk '{print $1}'`  
SYN_ACK_TO_ISP=`tcpdump -n -l -r $1 'src net not 192.168.0.0/24' and 'dst net 192.168.0.0/24' and 'tcp[13] == 18' | wc | awk '{print $1}'`  
RATIO=$(echo "scale=3; $SYN_ACK_TO_ISP/$SYN_ONLY_FROM_ISP" | bc)
```

# PF ALTQ tam açıqlama

Bütün çıkış trafiki aşağıdaki növbələrə bölünmüştür.

- |                             |                         |
|-----------------------------|-------------------------|
| - DNS müraciətləri          | - növbə <b>u_dns</b>    |
| - TCP ACK paketlər          | - növbə <b>u_ack</b>    |
| - Yüksək prioritetli trafik | - növbə <b>u_hpri</b>   |
| - Normal prioritetli trafik | - növbə <b>u_pri</b>    |
| - Aşağı prioritetli trafik  | - növbə <b>u_lowpri</b> |
| - Bütün qalan trafik        | - növbə <b>u_other</b>  |
| - Standart növbə            | - növbə <b>u_std</b>    |

Növbənin 1-i **default** olmalıdır. Bura trafik düşür və heç bir digər qaydada artıq analiz edilmir.

Eynilə client-in daxili şəbəkəsinə gedən trafikin bölgüsü uyğun şəkildədir, ancaq **u\_\*** əvəzinə **d\_\*** istifadə edilir.

Trafik servis/protokol (**HTTP**, **FTP**, **Torrent**)-lara görə standartlaşdırıldığı kimi istifadəçiye görə də standartlaşdırılacaq. Misal üçün, əgər mühasibatlıqdə bir maşına prioritet versək, onun bütün paketləri heç bir növbə gözləmədən kənara çıxış edə biləcək(həmçinin torrent,FTP). DNS müraciətlərdən və TCP ACK paketlərdən ibarət olan verilənlər isə yüksək prioritetə malikdir və dəyişməzdır. Misal üçün, hansısa komp-dan gələn və ya gedən trafik aşağı prioritetli qrupda olsa belə, bu **DNS** və **TCP ACK** paketlərinə şamil edilməyəcək.

Öncədən macros-larımızı təyin edək.

```
mst="modulate state"  
str="source-track rule"  
ext_if="em0"  
int_if="em1"
```

Aşağıdakı cədvələ əlavə edilən IP ünvanları yüksək prioritətlidir.

```
table <pc_hipri> persist {10.11.1.2}
```

Adı prioritetli PC-lər üçün.

```
table <pc_pri> persist {10.13.1.2 10.13.1.10 10.13.1.13 10.13.1.14 10.13.1.15}
```

Aşağıdakı ünvanlarla olan cədvələ yetkilər bağlanacaq.

```
table <ban> persist file "/etc/pf.ban"
```

İnanacağımız client-lərin cədvəli.

```
table <trust> persist {123.10.456.0/24 193.196.125.0/24}
```

Inzibatçının IP ünvanı

```
table <me> persist {210.211.13.84}
```

SIP provider-lərin IP ünvanları

```
table <sip_peers> persist {212.15.65.122 75.16.127.118}
```

PF-in özünü aparmasının dəyişdirilməsi üçün opsiyaları təyin edək. Bizim halda aşağıdakılardır:

- LoopBack adapter-də yoxlanış etməyəcək.
- Basic optimizasiya qaydasını yazırıq.
- Hər bir şəbəkə kartı üçün qoşulma statuslarının öyrənilməsini təyin edirik.
- Statusun maksimal sayını təyin edirik.

Əgər şəbəkə kartlarının və ya istifadəçilərin sayı çox olarsa, onda bu mənaları tələbatla uyğun dəyişməlisiniz.

```
set skip on lo0  
set ruleset-optimization basic  
set state-policy if-bound  
set limit states 20000
```

Trafikin normallaşdırılması(Scrub-laşma), scrub-laşma firewall-umuzun təhlükəsizliyinin artırılmasına şərait yaradır. Bizim halda trafikin normallaşması external interfeysdə baş verir, IP paketin identifikasiya sütununda təsadüfi ardıcılıqlı generasiya edilir, uzunluq TTL=128 təyin edilir, IP paketlərin defragmentasiyası və TCP qoşulmaların normallaşması baş verir.

```
scrub on $ext_if all random-id no-df min-ttl 128 fragment reassemble reassemble tcp
```

## ALTQ

Bütün fiziki şəbəkə kartlarında prioritələşməni mənimsədəcəyik. External şəbəkə kartında çıxan trafiki, Internal şəbəkə kartında isə daxil olan trafiki. Dissiplinanın seçimi çox önemli hissədir. Mümkün olan variantlar: **priq**, **cqfq**, **hfsc**

Biz **hfsc** istifadə edəcəyik, hansı ki, seçilmiş istifadəçi və ya servis üçün ayrılmış şəbəkəni təmin edir. **hfsc** planlaşdırılanın əsas işi bizim halda belə olacaq: əgər növbələşmənin açıqlanmasında növbəti **realtime** parametri göstərilibse, onda ona heç bir şeydən asılı olmayaraq, göstərilmiş şəbəkə hissəsi (**realtime**) veriləcək. **realtime** trafikdən öndə olan trafikin həcmi isə **bandwidth** həcmindən çıxarıllaraq hesablanır. **bandwidth** əsas yox, asılı parametrdir.

**bandwidth** həcmindən çıxan hər bir növbə üçün **realtime**-dan öndə olan trafikin həcmi hesablanır o vaxtadək ki, **upperlimit** parametrinə çatacaq (bu parametr sərt olaraq şəbəkəyə məhdudiyyət qoyur). **hfsc** planlaşdırmasında prioritet istifadə edilmir.

Öncə yazdıqlarımızdan başqa, **hfsc**-də **qlimit** parametri var, hansı ki, slotların sayını təyin edir. Slotlar eyni anda bu queue-dən keçən paket sayıdır. Bu limit həddinə çatdıqda paketlər **drop** (məhv) ediləcək. Buna görə də istifadəçinin sürəti azaltmasına şərait yaradır. Biz **RED**, ya da **ECN** istifadə etməyəcəyik, bunun əvəzinə **qlimit**-in sayını artıracaqız.

External şəbəkə kartımızda növbəni təyin edirik. Şəbəkə keçirməsinin həcmi ondan önce dayanan router-in ən azı 96%-ni təşkil etməlidir. Həmçinin elə burada kiçik səviyyəli növbələri təyin edirik.

```
altq on $ext_if hfsc bandwidth 800Kb queue  
{u_std,u_ack,u_dns,u_hipri,u_pri,u_lowpri,u_other}
```

Susmaya görə olan növbə

Heç nədən asılı olmayaraq, **25 kbit/s** istifadə edəcək. Bandwidth **1Kb** o deməkdir ki, əgər kanal hər şey və ya digər hansısa növbə ilə məşğul olsa, onda **u\_std** növbəsi, demək olar ki, **25 kbit/s**-dən yuxarı heç bir şey almayıacaq.

```
queue u_std bandwidth 1Kb qlimit 50 hfsc [default realtime 25Kb]
```

**u\_ack** növbəsi uzaq məsafədə olan host-a göndəriləcək **TCP ACK** paketləridir, hansı ki, ondan yüklenmə TCP protokolu ilə baş verir. Vacibdir ki, bu paketlər gecikmə olmadan keçsin. Provider-dən gələn **4Mbit/s** maksimal sürət üçün qənaətə alınmış **125Kbit/s** sürət tələb edilir.

```
queue u_ack bandwidth 1Kb qlimit 200 hfsc (realtime 125Kb)
```

DNS müraciətlər

Təmin edilmiş **25Kbit/s** tam olaraq kifayətdir. Çox lazım deyil. Ona görə də **1Kb** kifayət edər.

```
queue u_dns bandwidth 1Kb qlimit 50 hfsc (realtime 25Kb)
```

Yüksek prioriteti olan növbə

```
queue u_hipri bandwidth 300Kb qlimit 250 hfsc (realtime 200Kb)
```

Adı prioriteti olan növbə

```
queue u_pri bandwidth 300Kb qlimit 400 hfsc (realtime 150Kb)
```

Aşağı prioriteti olan növbə

```
queue u_lowpri bandwidth 100Kb qlimit 100 hfsc (realtime 75Kb)
```

Qalan bütün TCP və UDP trafiki üçün tələb edilən növbə

```
queue u_other bandwidth 97Kb qlimit 50 hfsc (realtime 25Kb)
```

Növbəni daxili interfeysdə təyin edirik - daxili trafikə prioritet təyin edirik.

Provider **5Mbit/s** verir, ona görə də **inetq** növbəsini **96%** həcmdə təyin edirik. Həmçinin daxili şəbəkə kartında çoxlu daemon-lar işləyir, məsələn, daxili **FTP**. Ona görə də **Lokal** trafiki **Internet** trafiklə qarışdırmaq lazım deyil. Daxili şəbəkə kartımızın şəbəkə kartı **100Mbit** dəstəklədiyinə görə **bandwidth-i 100Mb** edirik. İki növbə təyin edirik: biri daxili trafik, ikincisi isə, bala növbələri ilə birlikdə internet trafik.

```
altq on $int_if hfsc bandwidth 100Mb queue {etherq, inetq}
```

Daxili trafik üçün növbə

Bu növbəyə daxili şəbəkə kartından daxili istifadəçilərə gedən bütün paketlər düşəcək. **upperlimit** bu növbə üçün maksimal rəqəmi təyin edir. Nəzərə alın ki, bu növbəyə internetdən gələn cavablar düşməyəcək. Misal üçün, WWW serverdən gələnlər. Bu növbə kəskin şəkildə daxili trafik üçündür.

```
queue etherq bandwidth 95Mb hfsc (upperlimit 95Mb)
```

Internet trafik üçün növbə

Bu növbəyə düşən paketlər Internet şəbəkəsinə gedənlərdir. PUBLIC şəbəkə kartının analogiyasında olan alt növbələrə malikdir.

```
queue inetq bandwidth 4800Mb hfsc (upperlimit 4800Mb)
```

```
{d_std,d_ack,d_dns,d_hipri,d_pri,d_lowpri,d_other}
queue d_std bandwidth 1Kb qlimit 50 hfsc (default realtime 25Kb)
queue d_ack bandwidth 1Kb qlimit 50 hfsc (realtime 50Kb)
queue d_dns bandwidth 1Kb qlimit 50 hfsc (realtime 25Kb)
queue d_hipri bandwidth 1297Kb qlimit 500 hfsc (realtime 1000Kb)
queue d_pri bandwidth 2000Kb qlimit 500 hfsc (realtime 2000Kb)
queue d_lowpri bandwidth 1000Kb qlimit 500 hfsc (realtime 500Kb)
queue d_other bandwidth 500Kb qlimit 500 hfsc (realtime 240Kb)
```

#### Daxili ünvanların translyasiya qaydaları (NAT)

Ünvanların translyasiyası istenilən daxili şəbəkə kartından gələn IP ünvanın kənarə istenilən IP ünvanla çıxışıdır (Router-lerdə təyin edilən IP ünvanlardan başqa).

Bu həm fiziki, həm də VPN şəbəkə kartları ola bilər (**tun**,**gif**).

(**\$ext\_if**) - dairəvi mötərizənin daxilindədir, çünki external şəbəkə kartının IP ünvanı dinamik təyin edilir. (**\$int\_if:network**) və (**self**) dairəvi mötərizədədir, ona görə ki, **pfctl -sn** əmrinin çıxışında real ünvan və şəbəkələrin çıxışı olmasın. Bu, daxili şəbəkə kartında bir neçə alias və şəbəkə yerləşdiyi zaman çox rahat olur.

```
nat on $ext_if inet from ($int_if:network) to !(self) -> ($ext_if) port 1024:65535
```

**pc\_hipri** və **pc\_pri** istifadəçilərinə proxy-ni aşaraq keçmələrinə icazə veririk.

```
no rdr on $int_if inet proto tcp from <pc_hipri> <pc_pri> to !(self) port 80
```

Redirektor qaydası, hansı ki, daxili istifadəçilərin bütün DNS müraciətlərini **PUBLIC DNS** serverə yönləndirir. Bu, google və ya şirkətinizin öz daxili DNS serveri ola bilər.

```
rdr on $int_if inet proto {tcp udp} from ($int_if:network) to !(self) port 53
-> 85.132.57.58 port 53
```

Proxy serverə redirect

```
rdr on $int_if inet proto tcp from ($int_if:network) to !(self) port 80 -> 127.0.0.1
port 3128
```

Internet üzərindən daxildə olan Windows maşınınə RDP ilə yönləndirmə.

```
rdr on $ext_if inet proto tcp from any to ($ext_if) port 3389 -> 10.11.1.2 port 3389
```

Trafikin filtrasiyası qaydaları. Qaydaları aşağıdakı ardıcılılıqda qruplaşdıracaqıq.

Edəcəyimiz iş - interfeys, istiqamət, protokol, mənbənin ünvanı, mənbənin port-u, mənsəbin ünvanı və mənsəbin portu.

## **Antispoofing**

```
    antispoof quick for {$int_if lo0} inet
```

Yönlendirilə bilinməyən ünvanları block edək.

```
    block in quick inet from no-route to any
```

Broadcast-ları bağlayırıq.

```
    block in quick on $ext_if inet from any to 255.255.255.255
```

**ban** cədvəlində olan IP ünvanları block edirik. **return** opsiyası **TCP RST** qaytarır, hansı ki, **timeout** olmadan bağlayır. Yaxşıdır, o halda ki, reklam şəbəkələri block edilir və browser block edilən məzmunu gözləmədən düzgün datanı aça bilir.

```
    block return out quick on $ext_if inet from any to <ban>
```

Aşağıdakı iki qayda Firewall-ın tipini təyin edir ki, özümüz təyin elədiyimizdən başqa hər şey bağlıdır.

```
    block in all  
    block out all
```

Artıq izin verilən qaydalar.

Lazımı effekt və düzgün işləməsi üçün uyğun ardıcılıqlıda düzülmüşdür.

Daxili interfeys.

LoopBack şəbəkə kartına gedən TCP paketlərə izin veririk, hansı ki, orada Transparent Proxy serverimiz qulaq asır. Növbələri təyin edirik. Biz daxil olan trafikə prioritət təyin edə bilmərik, yalnız çıxana edirik. **d\_pri** prioritətinə düşən paketlər serverdən client-ə qaydan cavab paketləridir, beləliklə, biz endirmə sürətini stabillaşdırırıq.

```
pass in log quick on $int_if inet proto tcp from ($int_if:network) to 127.0.0.1  
port 31280 queue (d_pri, d_ack)
```

Növbəti 3 qayda daxili olmayan IP ünvanlara təyin edilmiş portlara TCP qohumlarını açır. Uyğun olan növbələr təyin edilir. **Tag** təyin etmək ona görə yaxşıdır ki, sonra biz çöl interfeysdə bunu digər trafikdən ayırib, lazımı növbəyə yerləşdirə biləcəyik. Nəzərə alın ki, **quick** opsiyası istifadə edilmir, ona görə də bu qaydaları uyğun ardıcılıqlıda düzəkmə lazımdır: az məhdudiyyətlidən çoxa doğru, ona görə ki, **quick** opsiyası olmadan ilk uyğun olan qayda yox, son uyğun olan qayda işləyir.

Quick opsiyası ona görə istifadə edilmir ki, uyğun olan qaydaya düşən paket növbəyə yerləşdirilib buraxılacaq və protokollara, ünvanlara əsaslanan izin verilmə, prioritət təyin etmə qaydalarınınadək

düşməyəcək (bizim halda bu, **pc\_hipri** və **pc\_pri**-da olan komplardır.). **TCP** protokol olduğuna görə **TCP ACK** paketlərinə də növbə əlavə edəcəyik. Mənbə ünvanı kimi qaydalarımızda !(self:network) sintaksisini istifadə edəcəyik. Bu, o deməkdir ki, yalnız şəbəkə kartlarımızda olmayan subnet və ya IP adreslər, həmçinin router-ə qoşulmuş IP ünvanlardan başqa, bütün paketlərə izin veriləcək və uyğun olaraq tag ediləcək. Misal üçün, bu, bizim daxili şəbəkəmizdən istənilən şəxsin PUBLIC IP ünvanına qoşulmasına izin verməyəcək.

```
pass in log on $int_if inet proto tcp from ($int_if:network) to !(self:network)
$dst queue (d_other d_ack) tag INET_OTHER
```

```
pass in log on $int_if inet proto tcp from ($int_if:network) to !(self:network)
port {20 21 25 110 143 5190 8080 081} $dst queue (d_lowpri d_ack) tag INET_LOWPRI
```

```
pass in log on $int_if inet proto tcp from ($int_if:network) to !(self:network)
port 443 $dst queue (d_pri d_ack) tag INET_PRI
```

Növbəti işləyəcək iki qayda həm bütün istifadəçilərə, həm də VPN-qrupda olan bütün istifadəçilərə mənimsədiləcək (**pc\_hipri** və **pc\_pri**). Ona görə də burada **quick** opsiyasından istifadə edəcəyik.

```
pass in log quick on $int_if inet proto tcp from ($int_if:network) to !(self:network)
port {22 3389} $dst queue (d_hipri d_ack) tag INET_HIPRI
```

```
pass in log quick on $int_if inet proto tcp from ($int_if:network) to !(self:network)
port 53 $dst queue (d_dns d_ack) tag INET_DNS
```

Növbəti iki qayda **TCP** analogiyasındadır və **UDP** protokolunda izin vermələri və növbələri təyin edir. Həmçinin onları tag edirik. İkinci opsiya **quick** bütün kateqoriyalı istifadəçilərə işləyəcək.

```
pass in log on $int_if inet proto udp from ($int_if:network) to !(self:network)
queue d_other tag INET_OTHER
```

```
pass in log quick on $int_if inet proto udp from ($int_if:network) to !(self:network)
port {53 123} queue d_dns tag INET_DNS
```

ICMP-ni açırıq.

```
pass in log on $int_if inet proto icmp from ($int_if:network) to !(self:network)
queue d_lowpri tag INET_LOWPRI
```

Növbəti qayda istifadəçilərdən yüksək və normal prioritetlə gələn trafikə izin verir. Bütün trafik ya çox yüksək (**hipri**), ya da yüksək (**pri**) prioriteti olacaq (**DNS** və **TCP ACK**-dan başqa). Burada biz protokol təyin etmirik, ancaq modulate state təyin edirik, hansı ki, ancaq TCP üçün mənimsədir. Bu, səhv olmayacaq, PF həddən artıq ağıllıdır. O, **modulate state-i TCP** protokoluna və keep state-i isə qalan protokollara mənimsədəcək.

```
pass in log quick on $int_if inet from <pc_hipri> to !(self:network) $mst queue (d_hipri, d_ack) tag INET_HIPRI
```

```
pass in log quick on $int_if inet from <pc_pri> to !(self:network) $mst queue (d_pri, d_ack) tag INET_PRI
```

Daxili Ethernet-ə Router-in daxili interfeysinə izin veririk.

```
pass in quick on $int_if inet from ($int_if:network) to ($int_if) queue etherq
```

İndi isə router-in daxili interfeysindən daxili şəbəkəyə gedən qaydaları açıqlayaq. Əgər sizdə router-in üstündə daxili şəbəkəyə yetki istəyən internetdən daxili şəbəkəyə yönləndirmə yoxdursa, ya da heç bir server/client yoxdursa, onda bu qaydalara ehtiyac yoxdur.

İlk qayda daxili şəbəkədə olan IP ünvanının üzərinə internet trafikin gəlməsinə izin verir və daxil olan trafikə çox yüksək prioritet təyin edir. Bu, daxili şəbəkəyə port yönləndirmə üçün üçüncü (birinci - **rdr**, ikinci - çöl interfeysdə izin verir) qaydadır.

```
pass out quick on $int_if inet proto tcp from !(self) to 10.11.1.2 port 3389 queue (d_hipri d_ack)
```

Router-in daxili şəbəkə kartından daxili şəbəkəyə gedən trafikin keçməsinə izin verən qaydadır. Hansı protokol olması önemli deyil. Bu, bütün **etherq** növbəsinə ötürülür.

```
pass out quick on $int_if inet from ($int_if) to ($int_if:network) queue etherq
```

PUBLIC Interfeys. Daxil olan trafik üçün izin vermə qaydası.

Vacibdir ki, hər bir daxili qaydaya görə maksimal state sayını təyin edək, hansı ki, onlar da qaydalar yarada bilər. Du DoS baş verən halda statusların hamısının bitməsinin qarşısını almaq üçün istifadə edilir. Ardıcıl 6 qaydanın sayesində router-in seçilmiş portlarına giriş üçün TCP qoşulmalarına izin verilir və həmçinin lazımi növbələr təyin edilir. Bu növbələr daxil olan trafik yox, çıxan trafikə prioritetlər təyin edəcək. Həmçinin ilk iki qaydaya diqqət yetirmək lazımdır, hansı ki, Router-də olan FTP serverə yetkini açır. 21-ci port ilə ötürülen əmrlər böyük prioritetlə **u\_pri** prioritetinə, verilənlər isə kiçik prioritetlə **u\_lowpri**-a ötürüləcək.

```
pass in quick on $ext_if inet proto tcp from <trust> to ($ext_if) port 21 $mst (max 100) queue (u_pri u_ack)
```

```
pass in quick on $ext_if inet proto tcp from <trust> to ($ext_if) port >=49152  
$mst (max 100) queue (u_lowpri u_ack)

pass in quick on $ext_if inet proto tcp from <me> to ($ext_if) port 22 $mst (max  
10) queue (u_hipri u_ack)

pass in quick on $ext_if inet proto tcp from <me> to ($ext_if) port 80 $mst (max  
100) queue (u_pri u_ack)

pass in quick on $ext_if inet proto tcp from <me /> to ($ext_if) port 5900 $mst  
(max 10) queue (u_hipri u_ack)
```

Internet üzərindən router-in daxili interfeysinə yox, daxildə olan istifadəçinin maşınınə RDP ilə girişi açan qayda.

```
pass in quick on $ext_if inet proto tcp from <me> to !(self) port 3389 $mst (max  
10) queue (u_hipri u_ack)
```

Növbəti qayda VPN-router-ə dünyanın istənilən nöqtəsindən gələn **PUBLIC** İP ünvanlarına girişə izin verir. Təhlükəsizlik məqsədilə **source-track** təyin edilmişdir.

```
pass in quick on $ext_if inet proto tcp from any to ($ext_if) port 5500 $mst  
max 10,$str, max-src-nodes 2, max-src-states 3, max-src-conn-rate 3/60) queue  
(u_hipri u_ack)
```

Növbəti iki qayda **UDP**-də olan seçilmiş portlara girişə açır.

```
pass in quick on $ext_if inet proto udp from any to ($ext_if) port 1194 (max  
20) queue u_pri
```

```
pass in quick on $ext_if inet proto udp from <sip_peers> to ($ext_if) port  
5060 (max 20) queue u_hipri
```

PUBLIC interfeysə ping-i açırıq.

```
pass in quick on $ext_if inet proto icmp from any to ($ext_if) icmp-type  
echoreq (max 100) queue u_other
```

PUBLIC interfeys. Çıxan trafik üçün izin verən qaydaların yazılması.

Növbəti 5 qayda daxili şəbəkə kartında qeyd edilmiş bütün trafiki çöl interfeysindən kənara çıxarıır. Bu, sadəcə verilənlərdir, hansı ki, daxili şəbəkədə olan istifadəçilərdən internetə gedir.

```
pass out quick on $ext_if inet from ($ext_if) to any $mst queue (u_dns u_ack)
tagged INET_DNS
```

```
pass out quick on $ext_if inet from ($ext_if) to any $mst queue (u_hipri
u_ack) tagged INET_HIPRI
```

```
pass out quick on $ext_if inet from ($ext_if) to any $mst queue (u_pri u_ack)
tagged INET_PRI
```

```
pass out quick on $ext_if inet from ($ext_if) to any $mst queue (u_lowpri
u_ack) tagged INET_LOWPRI
```

```
pass out quick on $ext_if inet from ($ext_if) to any $mst queue (u_other
u_ack) tagged INET_OTHER
```

Və son qaydalar, hansı ki, çöl şəbəkə kartından çıxan qoşulmalara izin verir. Bu, router-in özünün trafiki olacaq. Bütövlükdə bu trafikin prioriteti istifadəçilərdən daxili şəbəkəyə gedən trafikdən böyükdür.

```
pass out quick on $ext_if inet proto tcp from ($ext_if) to any port 53 $mst
queue (u_dns u_ack)
```

```
pass out quick on $ext_if inet proto tcp from ($ext_if) to any $mst queue
(u_pri u_ack)
```

```
pass out quick on $ext_if inet proto udp from ($ext_if) to any port {53 123}
queue u_dns
```

```
pass out quick on $ext_if inet proto udp from ($ext_if) to <sip_peers> port
5060 queue u_hipri
```

```
pass out quick on $ext_if inet proto udp from ($ext_if) to any queue u_pri
```

```
pass out quick on $ext_if inet proto icmp from ($ext_if) to any $mst queue
u_lowpri
```

Bir önemli hissə var, hansı ki, dərindən araşdırılmalıdır. İkinci qayda router-ə və **80**-ci porta **TCP** ilə qoşulmaya izni açacaq. Bu qaydanın üstünə həmçinin daxili istifadəçilərdən **HTTP** trafik gələcək

(Proxy serverə gedən trafik). Bizim halda bu iki trafikin prioriteti (**u\_pri**) eyni olduğuna görə hər şey yaxşıdır. Ancaq müxtəlif növbələr təyin etmək gərək olsa(məsələn, daxili istifadəçilərdən gələn **HTTP** - **u\_lowpri** növbəsi, router-in çöl interfeysindən gələnlər isə - **u\_pri**), onda proxy server üçün nəzərdə tutduğumuz qayda üçün **user uid** opsiyasından istifadə etmək lazımdır və onu router-in çöl interfeysinə aid olan qaydadan sonra yazmaq lazımdır. Misal üçün, proxy nobody istifadəçi adından işə düşsə, aşağıdakı kimi olacaq:

```
pass out quick on $ext_if inet proto tcp from ($ext_if) to any user nobody  
$mst queue (u_lowpri u_ack)
```

```
pass out quick on $ext_if inet proto tcp from ($ext_if) to any $mst queue  
(u_pri u_ack)
```

Uyğun olaraq, həmçinin daxili şəbəkədən proxy-ə gedən izin yetkisi də dəyişəcək. **d\_pri** növbəsini **d\_lowpri**-a dəyişmək lazımdır.

```
pass in log quick on $int_if inet proto tcp from ($int_if:network) to 127.0.0.1  
port 31280 queue (d_lowpri, d_ack)
```

Həmçinin **user** opsiyasının istifadəsində BUG olmasını qeyd eləmək lazımdır. Bu haqda **pf.conf**-un manualında belə yazılmışdır. Ancaq testlərimdə bu, bug olmadı. Hər hal üçün **user** opsiyasını istifadə etməsəniz, yaxşı olar.

**Debugging.** PF-in özünün monitoring eləməsi üçün çox yaxşı statistik cədvəlləri olur. Yüklənmiş növbələrə baxmaq üçün:

```
pfctl -sq
```

Real vaxtda növbələrin yüklənməsinə baxmaq üçün aşağıdakı əmri istifadə edin:

```
pfctl -vvsq
```

Həmçinin PF statistikalarına onlayn rejimdə baxmaq istəyirsinizsə, **pftop** programını portlardan (**/usr/ports/sysutils/pftop**) yükleməyiniz daha düzgün olar.

# PF Bridge Firewall

Əgər müəyyən bir serveri olan şəxs öz Internet xidmətləri təchizatçısına müraciət edərək soruşa:

1. Mən istəyirəm ki, PUBLIC IP ünvanı serverimin özündə olmaq şərtilə eyni anda Firewall arxasında da qalsın.
2. Həmçinin mənə lazım olan port, protokolları həmin FireWall-la filter etmək imkanım olsun.

Cavab ağır olacaq ☺, çünkü bu işi NAT-sız görmək mümkün deyil. O halda da serverə verilən IP ünvanı **Private** aralığından olacaq. Bellidir ki, NAT hər bir halda Private aralığı NAT edib kənara çıxarıv və daxildə filtrlədir. Bu isə istifadəçini qane eləmir. Bu iş üçün bəhəli PIX və ya ASA avadanlıqları var.

Ancaq FreeBSD və PF Firewall bu işi Bridge rejimində görə bilir.

Biz indi bu işi VmWare Workstation 11-ci versiya üzərində edəcəyik.

Tələbatlar: **FreeBSD 10.1 x64**, 1 Windows7 Desktop son istifadəçi və bir ədəd Windows8.1(Şəxsi istifadə olunan Desktop)

Virtual məşinlərimizin virtual şəbəkələri aşağıdakı kimi olacaq:

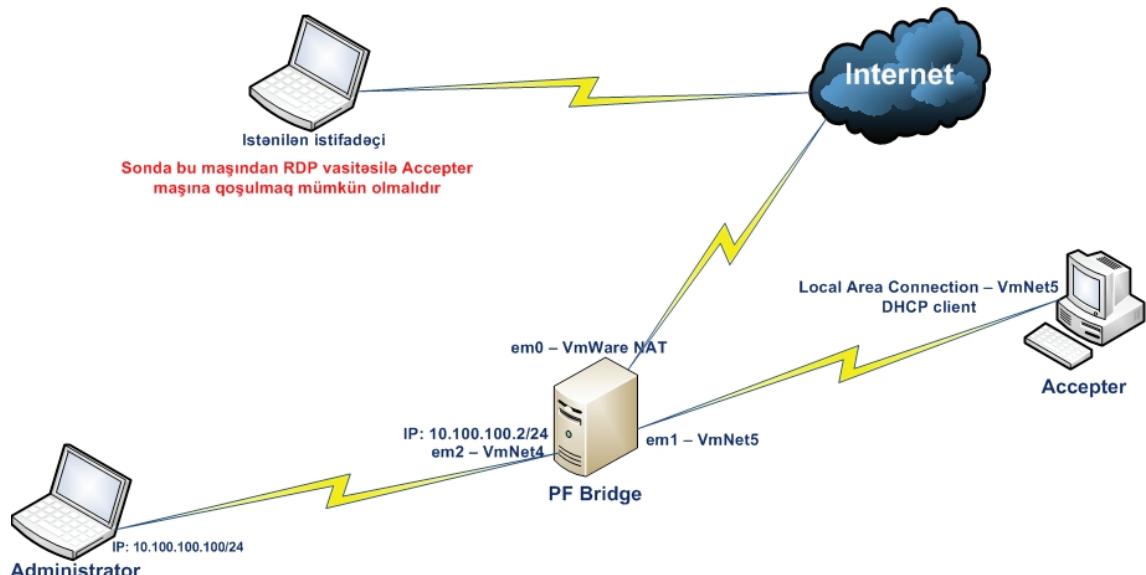
**FreeBSD 10.1 x64:** **em0** - NAT (Windows8.1 virtual şəbəkə kartı vasitəsilə bu şəbəkəni görəcək.)  
**em1** - **VmNet5**  
**em2** - **VmNet4** (Windows8.1-in LoopBack şəbəkə kartına təyin edilmişdir.  
İdarə etmə üçündür. IP ünvanı **10.100.100.2/24**)

## Windows7: Local Area Connection

## Windows8.1: - LoopBACK(10.100.100.100/24)

- **VmNet5** (FreeBSD10.1 maşının em1 şəbəkə kartı ilə qoşudur). Sonda Vmware Workstationda olan NAT şəbəkəsindən DHCP vasitəsilə IP ünvan eldə etməlidir.
- **vmnet4** (FreeBSD 10.1 x64 maşın üçün idarəetmə şəbəkəsidir)
- NAT** – Vmware workstation yükləndikdə yaranan virtual şəbəkə kartıdır.

Məqsədimiz ayrı-ayrı şəxsi virtual şəbəkələrdə olan Desktop məşinlərin Firewall üzərindən keçərək bir-birini görməsini təmin etməkdir. Sonda PF Firewall vasitəsilə Windows7 maşına gedən RDP portu bağlayıb test edəcəyik. Şəbəkə quruluşumuz aşağıdakı şəkildəki kimi olacaq:



FreeBSD məsələlərimizi edək

/etc/rc.conf StartUP quraşdırma faylımızda Bridge şəbəkə kartları üçün lazımi quraşdırmları edirik.

```
hostname="ps"
ifconfig_em2="inet 10.100.100.2 netmask 255.255.255.0"
- İdarəetmə üçün tələb edilən şəbəkə kartımızdır.
```

```

sshd_enable="YES"

# Bridge konfiqurasiyası
cloned_interfaces="bridge0" - Bridge virtual şəbəkə kartımızı yaradırıq.

ifconfig_bridge0="addm em0 addm em1 up" - Yaratdığımız virtual 'bridge0' şəbəkə kartımıza iki ədəd fiziki 'em0' (NAT) və 'em1' (VmNet5) şəbəkə kartlarını əlavə edirik.

ifconfig_em0="up" - Bridge üçün mənimsətdiyimiz şəbəkə kartlarını işə salırıq.

ifconfig_em1="up" - Bridge üçün mənimsətdiyimiz şəbəkə kartlarını işə salırıq.

pf_enable="YES" - PF FireWall-ı işə salırıq. Ancaq bundan önce kernel-i lazımi opsiyalarla kompilyasiya etməyi unutmayın.

pf_rules="/etc/pf-bridge.conf" - PF-in Startup vaxtı işə salacağı qaydalar olan faylın ünvani.

pflog_enable="YES" - pflogd-ni işə salır.

pflog_logfile="/var/log/pflog" - pflogd-nin jurnal faylının ünvani.

```

PF üçün lazım olan Startup Firewall faylını öncədən hazırlayıraq.

```

ee /etc/pf-bridge.conf
##### PF əmrlər #####
# pfctl -F all (Bütün qayda və statusları sıfırlayıraq).
# pfctl -f /etc/pf.conf (Qaydaları PF-in qayda faylından oxuyun).
# pfctl -F all -f /etc/pf.conf (Öncəki iki işi bir əmrlə yerinə yetirir).
# pfctl -sa (Bütün informasiyanı çap edin -sa>Show All").
# pfctl -sr (Hal-hazırkı qaydaları göstərin).
# pfctl -sr -v (Hal-hazırkı qaydaları statuslarla göstərin).
# pfctl -vsq (pftop-a oxşar nəticə çap edir).

```

```

# Macros: Dəyişənləri elan edirik.
mgt_if="em2" # İdarəetmə üçün tələb edilən şəbəkə kartı.
ext_if="em0" # VmWare Workstation NAT şəbəkə kartı.
int_if="em1" # Windows7 ilə eyni VMNet5-də olan şəbəkə kartı.

```

```

# Normallaşdırma:
# Fraqmentleri yiğib aydınlaşdırmaq və ya
# ikitərəfli lazımsız trafikləri azaltmaq.
scrub in all

# Filtrləmə: Test üçün hər 3 şəbəkə kartımıza hər şeyi açırıq.
pass log on $ext_if all
pass log on $int_if all
pass log on $mgt_if all

# Loopback şəbəkə kartımıza girişini açırıq.
pass in quick on lo0 all

# Hər bir şeyi bağlayıb və jurnallama işi aparmaq üçün
# aşağıdakıları edə bilərik. Ya da konkret port üçün girişini bağlayaraq,
# test edə bilərik. Ancaq sonra qaydanın qarşısına komentariya yerləşdirib,
# PF-in qaydalarını yenidən yükləməyi unutmayın. Çünkü əks halda, RDP ilə
# Windows7 maşına qoşula bilməyəcəksiniz.
block log on em0 proto { tcp udp } from any to any port { 3389 }
#block log on $mgt_if all
#block log on $ext_if all
#block log on $int_if all

# İdarəetmə tərəfdən giriş üçün konkret yetkilər verə bilərik.
#pass out on $mgt_if all keep state
#pass in on $mgt_if proto tcp from any to $mgt_if port 80 keep state
#pass in on $mgt_if proto tcp from any to $mgt_if port 22 keep state

```

Kernel-in lazımı modullarını kernel-in startup quraşdırma faylına əlavə edirik ki, sistemin yenidənyüklənməsindən sonra işləsin.

<b>ee /etc/sysctl.conf</b>	
<b>net.link.bridge.pfil_bridge=1</b>	- Bridge rejimdə paket filtrasiyasını aktivləşdiririk.
<b>net.link.bridge.pfil_member=1</b>	- Şəbəkə kartlarımızda Packet Filtrasiyasını <b>in</b> (giriş) və <b>out</b> (çıxış) üçün aktivləşdiririk.

Kernel-imizi tələb edilən opsiyalarla kompilyasiya edək:

```
cd /sys/`uname -p`/conf/  
cp GENERIC kernel  
  
ee kernel
```

- Kernel-in quraşdırma qovluğuna daxil olurq.  
- Susmaya görə olan **GENERIC** kernel faylini '**kernel**' adlı fayla nüsxələyirik.  
- Kernel faylinin ən sonunda yeni sətir açıb aşağıdakı sətirləri əlavə edirik.

<b>options</b>	<b>SMP</b>	- Symmetric MultiProcessor Kernel (Bu sətir prosessorlar arası keçid üçün mütləq tələb edilir).
----------------	------------	---

```
# Kernel-in Bridge şəbəkə kartları dəstəkləməsi üçün aşağıdakı alət tipini  
# əlavə edirik.
```

```
device      if_bridge
```

```
# PF Firewall, jurnallama və sinxronizasiya imkanlarını əlavə edirik.
```

```
device      pf  
device      pflog  
device      pfsync
```

```
# ALTQ (Alternate Queries) QoS üçündür.
```

```
options      ALTQ  
options      ALTQ_CBQ  # Class Bases Queuing (CBQ)  
options      ALTQ_RED  # Random Early Detection (RED)  
options      ALTQ_RIO  # RED In/Out  
options      ALTQ_HFSC # Hierarchical Packet Scheduler (HFSC)  
options      ALTQ_PRIQ # Priority Queuing (PRIQ)  
options      ALTQ_NOPCC # SMP modulu program təminatının prosessorların core-larına  
                      keçid üçün istifadə edilir. (multiprocessor)
```

**Qeyd:** Bunun üçün kernel-də '**options SMP**' mütləq olmalıdır.

```
# Digər imkanlar
```

```
options      IPSTEALTH # TTL sütunlarının redakte edilməsi üçündür.
```

```
# Mexanizm traceroute-dan müdafiə üçündür.
```

```
options      HZ=1000
```

```
cd /usr/src
```

- Kernel-i kompilyasiya etmək üçün lazımi ünvana daxil oluruq.

```
make buildkernel KERNCONF=kernel
```

- Kernconf-a yeni '**kernel**' adlı faylimizi kompilyasiya etməsi üçün ünvani göstəririk.

```
make installkernel KERNCONF=kernel
```

- Kernconf-a yeni '**kernel**' adlı kernel-i sistemə yükleməsini deyirik.

```
reboot
```

- Sonda sistemimizə yenidən yüklənmə edirik ki, yeni kernel işə düşsün.

Görülən işin araşdırılması üçün TCPDUMP-i aşağıdakı sintaksislə istifadə edə bilərsiniz.

```
tcpdump -e -tttt -i pflog0
```

- '**pflog0**'-un jurnal faylini onlayn analiz etmək üçün istifadə edilir.

```
tcpdump -n -e -tttt -v -r /var/log/pflog - Jurnal faylı oxumaq üçün istifadə edilir.
```

```
tcpdump -nexttti pflog0
```

- Jurnalları şəbəkə kartından onlayn analiz etmək üçün istifadə edilir. (Bu, '**pflogd**'-nin işinə qarışır).

Monitoring üçün PFTOP yükləyə bilərsiniz.

```
cd /usr/ports/sysutils/pftop
```

- Port ünvani daxil oluruq.

```
make install clean
```

- Yükleyirik.

```
rehash
```

- Əmr bazasını yeniləyirik.

```
pftop
```

- Bütün qoşulma statuslarını göstərir.

Sonda hansısa uzaq maşından, ya da elə Windows8.1 maşınınızdan Windows7 maşına RDP vasitəsilə qoşularaq test edin. Eyni işi hər iki tərəf üçün edə bilərsiniz.

# IPFW Firewall

IPFW Stateful (Statusa əsaslanan) firewall-dır. Paketi hesablama, jurnallama, NAT, dummynet(4) şəbəkənin boğulması, yönlədirmə, bridge rejimi və ipstealth imkanlarına sahibdir.

FreeBSD-nin tərkibində susmaya görə olan IPFW üçün qaydalar faylı `/etc/rc.firewall` vardır, hansı ki, fərqli ssenarilərə əsaslanan firewall tipini təyin edib, fərqli qaydaları yazır.

IPFW Firewall tipləri aşağıdakılardır:

- **open:** Bütün trafiki buraxır.
- **client:** Yalnız bu maşını qoruyur.
- **simple:** Bütün şəbəkəni qoruyur.
- **closed:** LoopBack şəbəkə kartı üçün İP trafiki tamamilə bağlayır.
- **workstation:** Dinamik qaydalar istifadə edərək, yalnız bu maşını qoruyur.
- **UNKNOWN:** Firewall qaydalarının yüklənməsinin qarşısını alır.
- **filename:** Firewall qaydaları olan faylin tam ünvanı.

**Qeyd:** Əgər bu firewall şablon tiplərin quruluşu sizin tələbə uyğun olmazsa, istədiyiniz zaman onları `/etc/rc.firewall` faylında dəyişə bilərsiniz.

Öz tələblərinizə uyğun olan qaydaların yüklənməsini istəsəniz, `/etc/rc.conf` StartUP faylinə

`firewall_script="/etc/ipfw.rules"` sətrini əlavə edib, lazım olan qaydaları isə "`/etc/ipfw.rules`" faylına yazmağınız kifayətdir.

Və ya jurnallamani StartUP-a əlavə etmək istəsəniz, `/etc/rc.conf` faylına "`firewall_logging="YES"`" əlavə etməyiniz kifayətdir. Ancaq jurnallamaya limit təyin etmək istəsəniz, "`/etc/sysctl.conf`" faylına `net.inet.ip.fw.verbose_limit=3` (3 - eyni jurnal sətri maksimum 3 dəfə təkrar oluna bilər) əlavə etmək lazımdır.

`service ipfw start` - Əmrli konsol-dan işə salırsınız.

**Qeyd:** Ancaq unutmayın ki, `IPFW default_to_accept` rejimində işləməzsə, avtomatik olaraq, 65535-ci qaydada bütün trafik hər yerə bağlanacaq və SSH ilə qoşulmuş olsanız, sessiyanız atacaq.

`sysctl net.inet.ip.fw.verbose_limit=3` - CMD vasitəsilə jurnallanmani işə salırıq.

`IPFW` adı halda sistemdə aktiv olmur. Onu ya sistem modullarından yüklemək, ya da kernel-ə əlavə edib yenidən kompilyasiya etmək lazımdır.

İlk olaraq modullardan yüklenməni açıqlayacaq

`kldload ipfw`

- Kompilyasiya etmədən modullardan yükleyirik.

`kldload ipdivert`

- NATD modulunu kompilyasiya etmədən yükleyirik.

`kldload dumynet`

- Şəbəkə boğucu modulu kompilyasiya etmədən yükleyirik.

Eyni qayda ilə kerneldə olmayan və yüklənmiş modulları sistemdən çıxarmaq olar.

**Qeyd:** Ancaq sistem o modulları istifadə edirsə, çıxarmağa izin verməyəcək.

`kldunload ipfw`

`kldunload ipdivert`

`kldunload dumynet`

**Qeyd:** Ancaq sistemdə yenidən kompilyasiya etmədən firewall-in istifadəsi məsləhət deyil. Şəxsi təcrübəmdə bir neçə dəfə problemlı işlədiyinin şahidi olmuşam.

**Qeyd:** Sistem yenidənyüklənməsindən sonra modulların işləməsini istəsəniz, aşağıdakılari etməyiniz kifayətdir.

**/etc/rc.conf** faylına StartUP-da işlemesi için aşağıdaki sətirləri əlavə edirik:

<b>firewall_enable="YES"</b>	- Firewall-ımızı işə salırıq.
<b>firewall_type="OPEN"</b>	- Tipini <b>Open</b> seçirik.
<b>dummynet_enable="YES"</b>	- Şəbəkə boğucunu aktiv edirik.
<b>natd_enable="YES"</b>	- Əgər firewall aktivdirse, NATD işləyəcək.
<b>natd_interface="em0"</b>	- NAT edəcək şəbəkə kartını təyin edirik.
<b>tcp_drop_synfin="YES"</b>	- 'SYN+FIN' TCP paketlərin <b>FIN</b> flag-ı ilə gələnləri bağlayırıq.
 <b>/boot/loader.conf</b>	- Modul StartUP faylımiza aşağıdaki sətirləri əlavə edirik.
 <b>ipfw_load="YES"</b>	- Firewall

Sistem yenidənyüklənməsindən sonra **ipfw show** əmrini daxil etsək, aşağıdakı nəticəni əldə etmiş olacaqıq:

```
00050 0      0 divert 8668 ip4 from any to any via em0
00100 0      0 allow ip from any to any via lo0
00200 0      0 deny ip from any to 127.0.0.0/8
00300 0      0 deny ip from 127.0.0.0/8 to any
65000 300 31534 allow ip from any to any
65535 0      0 deny ip from any to any
```

### İkinci halda kernelə əlavə edək

IPFW firewall-ı kernel daxilinə əlavə edib kompilyasiya etmək istəsək, aşağıdakı addımları yerinə yetiririk:

<b>cd /sys/`uname -p`/conf</b>	- Platformamıza uyğun ünvana daxil oluruq.
<b>cp GENERIC ipfw kern</b>	- Susmaya görə olan kernel faylini bizi rahat olaraq başqa fayla nüsxələyirik.

**ipfw kern** faylında bizə lazım olan sətirləri dəyişirik. **options** olan sətirləri faylin sonuna əlavə edirik. Həmçinin lazım olmayan imkanları və driver-ləri fayldan silsəniz, kernel-inizi normal kiçildə bilərsiniz. Bu, sizin sistemin sürətli işləməsinə gətirən səbəblərdən biridir.

<b>ident ipfw kern</b>	- ident-in qarşısında olan faylin adını yaratdığımız faylin adı ilə mütləq dəyişirik.
------------------------	---

<b>options</b>	<b>IPFIREWALL</b>	- IPFW firewall-in özünü işe salırıq.
<b>options</b>	<b>IPFIREWALL_VERBOSE</b>	- Jurnallamanı kernel-ə əlavə edirik.
<b>options</b>	<b>IPFIREWALL_VERBOSE_LIMIT=3</b>	- Eyni jurnal sətri 3 dəfədən artıq təkrarlanması.
<b>options</b>	<b>IPFIREWALL_FORWARD</b>	- IPFW yönləndirməni bacarsın (Yalnız daxili şəbəkə kartları arası). Ancaq <b>FreeBSD10.1</b> -dən etibarən susmaya görə sistemdə olur.
<b>options</b>	<b>DUMMYNET</b>	- Şəbəkənin boğulması bacarığı olsun.
<b>options</b>	<b>IPDIVERT</b>	- Sistem NAT bacarığına malik olsun.
<b>options</b>	<b>IPFIREWALL_NAT</b>	- Tək sistemin NAT bacarığı ilə yox, həmçinin IPFW bacarığı ilə NAT-ın olsun.
<b>options</b>	<b>LIBALIAS</b>	- IPFW imkanları ilə NAT bu modulsuz MASQUERADING edə bilmir.
<b>options</b>	<b>IPFIREWALL_DEFAULT_TO_ACCEPT</b>	- Firewall susmaya görə olan vəziyyətdə hər yerə açıqdır. (Heç bir halda məsləhət deyil.)
<b>cd</b> ..../..		- 3 ünvan geri qayıdırıq, yəni <b>/usr/src</b>
<b>make buildkernel KERNCONF=ipfwkern</b>		- kernel-ə deyirik ki, kompilyasiya ediləcək faylin adı <b>ipfwkern</b> -dir. Kompilyasiya gedəcək və bitdikdən sonra aşağıdakı sətri işe salırıq.

**Qeyd:** Kompilyasiya müddəti ən azı **30** dəqiqə olur.

<b>make installkernel KERNCONF=ipfwkern</b>	- Əmrələ deyirik ki, ipfwkern adla kompilyasiya edilmiş kernel və ona aid olan bütün fayllar yüklənsin.
 2-ci üsulda yalnız <b>/etc/rc.conf</b> StartUP faylinə aşağıdakı sətirləri əlavə edirik:	
<b>firewall_enable="YES"</b>	- Firewall-in sistem yenidənyüklənməsindən sonra avtomatik işə düşəcək.
<b>firewall_type="OPEN"</b>	- Firewall-un tipi açıqdır.
<b>natd_enable="YES"</b>	- NATD modulu işə salırıq.
<b>natd_interface="em0"</b>	- PUBLİC Nat şəbəkə kartını elan edirik.
<b>reboot</b>	- Sistemə yenidənyüklənmə əmri ötürürük ki, yeni kernel IPFW ilə işə düşsün.
<b>/etc/rc.d/ipfw restart</b>	- Əmr IPFW Firewall-a aid olan servisləri yenidən yükleyir. Kernel-də olan <b>net.inet.ipfw.enable</b> -i əvvəl <b>sıfır</b> , sonra bir edir. NAT Modullarını reload edir.

## IPFW ilkin əmrlər

**ipfw list**

**ipfw -at list**

**ipfw zero**

- Firewall-ın hal-hazırkı qaydalarını çap edir.

- Hal-hazırkı qaydaları '-t' timestamp və '-a' statistika ilə çap edir.

- Bütün qeydə alınan paket sayılarını sıfır edir.

Unutmayaq ki, ipfw yuxarıdan aşağıya oxuyur və paketlə üst-üstə düşən ilk qaydanı qəbul edir. Hər qaydanın müxtəlif rəqəmlər altında yazılıması vacibdir.

**00050 divert 8668 ip4 from any to any via em0**

- ipv4 versiya olan IP ünvanlarının hamısını NAT ilə **em0** şəbəkə kartının vasitəsilə dünyaya **50** rəqəmi ilə olan qaydanın üstüne yönəldir.

**00100 allow ip from any to any via lo0** - **100** rəqəmi ilə olan qaydanın altında, hər yerdən **100** şəbəkə kartına gələn bütün trafiki qəbul edin.

**00200 deny ip from any to 127.0.0.0/8** - **200** rəqəmi ilə olan qaydanın altında, hər yerdən **127.0.0.0/8** şəbəkə aralığının üstünə gedən trafiki bağlayırıq.

**00300 deny ip from 127.0.0.0/8 to any** - **300** rəqəmi ilə olan qaydanın altında, **127.0.0.0/8** şəbəkə aralığından hər yerə çıxışı bağlayırıq.

**65000 allow ip from any to any**

- **65000** rəqəmi ilə olan qaydanın altında, istənilən ünvandan gələn istənilən trafiki istənilən ünvan üçün açırıq.

**65535 deny ip from any to any**

- **65535** rəqəmi ilə olan qaydanın altında, istənilən ünvandan gələn istənilən trafiki istənilən ünvan üçün bağlayırıq.

**ipfw add 101 divert 8668 ip4 from any to me in via em0**

- 101-ci rəqəm altında, em0 şəbəkə kartının girişində istənilən ünvandan IPv4 ilə mənə müraciət gələrsə, NAT edirik.

```
ipfw delete 500  
ipfw delete 500 600 1000
```

- 500-cü rəqəmin altında olan qaydanı silirik.
- 500,600 və 1000 rəqəm altında olan qaydaları silirik.
- Bütün qaydaları təmizləyirik.

```
ipfw f
```

#### Bir az sintaksisdən danışaq

<b>ip from</b>	- hansı IP ünvan(və ya kimin) tərəfindən
<b>any</b>	- istənilən
<b>allow</b>	- qəbul
<b>deny</b>	- qadağa
<b>divert</b>	- Socket (IP:port)-ə yönləndirmək üçün istifadə olunur.(yönləndirmə)

```
skipto 400
```

- Görünən qayda ilə üst-üstə düşən paket olduqda, 400-cü qaydayadək digər heç bir qayda tərəfindən oxunmayacaq və paket birbaşa 400-cü qaydaya ötürüləcək.

Məsələn: **add 100 skipto 400 all from any to any**

- Mövcud qoşulma statusuna baxır, əgər paketin girişinə izin verilibsə, həmin paketin çıxışı üçün dinamik qayda yaradır. Dinamik qayda bu qoşulmalar üzərindən paketlərin gedisi dayanmayanadək, ya da qaydanın yaşama müddəti bitməyənədək silinməyəcək. **timeout**-lar **sysctl** tərəfindən idarə edilir.

```
keep-state
```

İllkin olaraq dinamik qaydaların saxlanması üçün istifadə edilən, quraşdırılmış və hal-hazırkı keş cədvəl. Cədvəlin mənasını yalnız boş olduğu halda dəyişmək olar. Ona görə də yerində dəyişmək üçün flush tələb edilə bilər.

```
net.inet.ip.fw.dyn_buckets: 256
```

```
net.inet.ip.fw.curr_dyn_buckets: 256
```

Dinamik qaydaların hal-hazırkı siyahısı (Yalnız oxumaq olar)

```
net.inet.ip.fw.dyn_count: 3
```

Dinamik qaydaların maksimal sayı. Bu həddə çatdıqdan sonra öncəki qaydalar köhnəlməyənədək yeni dinamik qaydalar əlavə etmək olmayıcaq.

```
net.inet.ip.fw.dyn_max: 1000
```

Göstərilən dəyişənlər dinamik qaydaların mövcud olma vaxtlarını saniyelərlə təyin edir. SYN paketlərin ilkin mübadiləsində mövcudluq vaxtı böyük təyin edilmir, ancaq hər iki SYN paketin alınmasından sonra böyüdüür və bitmə paketləri **FIN**, ya da **RST** gelişində yenidən kiçildilir.

```
net.inet.ip.fw.dyn_ack_lifetime: 300  
net.inet.ip.fw.dyn_syn_lifetime: 20  
net.inet.ip.fw.dyn_fin_lifetime: 20  
net.inet.ip.fw.dyn_rst_lifetime: 5  
net.inet.ip.fw.dyn_short_lifetime: 30
```

Sayıları yalançı TCP paketlərdən qorumaq üçün dinamik qaydaların istifadə edilməsi uyğundur. Aşağıdakı qaydalar sizin Firewall-a izin verir ki, dinamik qaydaları yalnız bizim şəbəkədən SYN paketləri ilə çıxarsa, yaradılsın. Dinamik qaydalar ilk qaydada yazılan **check-state**, ya da **keep-state** vasitəsilə ilk emalda yoxlanılır.

**check-state** qaydasını ilk rəqəmlər altında yazmaq məsləhətdir, ona görə ki, qaydalara baxdıqda iş azalır.

Nəzərə alın ki, statusu yoxlayan qaydalar **DoS** hücumuna məruz qala bilirlər (çoxlu SYN paketlərin göndərilməsi çoxlu dinamik qayda yaradılmasına gətirir.)

Bu tip hücumları qismən **sysctl** vasitəsilə azaltmaq olur.

```
ipfw add check-state  
ipfw add deny tcp from any to any established  
ipfw add allow tcp from my-net to any setup keep-state
```

#### **check-state**

- Paketi dinamik yığım qaydalarına görə yoxlayır. Əgər uyğunluq tapılırsa, növbəti axtarış dayandırılır. Əks halda, növbəti qaydaya kecid alınır. Əgər check-state qaydası tapılmazsa, dinamik qayda yığımı ilk keep-state qaydası ilə yoxlanılır.

#### **established**

- Yalnız TCP paketlər üçün. RST, ya da ACK bitləri təyin edilmiş paketlərə uyğun gəlir.

#### **setup**

- SYN bit-i olan TCP paketlərinə uyğundur, ancaq ACK tullanılmış paketlərdə. Bu, 'tcpflags syn,!ack' sintaksisin qısa formasıdır.

#### **fwd**

- Yönləndirir (Əsasən, squid üçün istifadə olunur.

Yalnız daxili şəbəkə kartlarına forward edir.)

**via**

- Üzərindən keçid üçün istifadə olunur.

**limit {src-addr | src-port | dst-addr | dst-port} N**

- **keep-state** ilə eynidir, ancaq dinamik qaydalar təyin edilmiş hədd aşılarsa, yaranmağa başlayacaq. Bu üsulla asanlıqla seçilmiş IP ünvanı, ya da porta olan eynivaxtı qoşulma sayını limitləmək olar.

**count**

- Jurnallama üçün daxil olan paketləri sadəcə sayı. Bu, billing sistemlərində istifadə edilən qaydalarda yazılır. Paketləri hesablaşdırmaq üçün istifadə edilir.

**pipe**

- Kanalı boğmaq üçün istifadə olunur.

**/etc/sysctl.conf kernel StartUP** faylında firewall-ımızın işləmə səviyyəsini təyin edə bilərik:

**net.inet.ip.fw.enable=1**

- 3-cü səviyyə süzgəci

**net.link.ether.ipfw=1**

- 2-ci səviyyə süzgəci

Əməliyyat sistemimizin daxilində olan 'cavid' adlı istifadəçinin **80**-ci port ilə hər yerə çıxışını bağlayırıq.

**ipfw add 1000 deny ip from me to any dst-port 80 uid cavid**

Aşağıdakı sətir avtomatik olaraq dinamik qaydalar yaradacaq. **10.0.0.10** IP ünvanlı istifadəçiye **stateful inspection**la qoşulmağa icazə veririk.

**ipfw add 04000 allow ip from 10.0.0.10 to any keep-state**

Növbəti sətirlə 10.0.0.0/24 şəbəkəsini hər yerə bağlayırıq.

**ipfw add 40000 deny ip from 10.0.0.0/24 to any**

İşlək vəziyyətdə olan dinamik qaydaları görə bilərik.

**ipfw -d show**

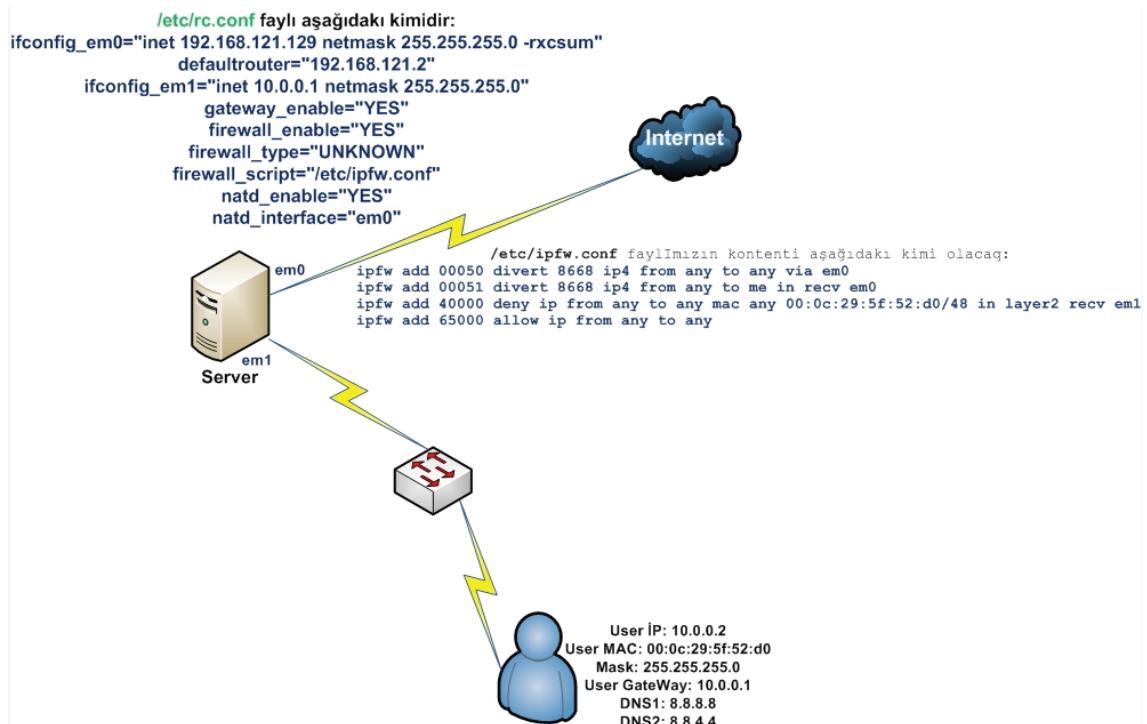
```
## Dynamic rules (1):
64000 78 7072 (ls) STATE tcp 192.168.121.1 50533 <-> 192.168.121.129 22
```

**ipfw -de show**

- Vaxtı bitmiş '**-e**' expired və işlək '**-d**' dinamik qaydaları aşağıdakı əmrlə görə bilərik.

### IPFW qaydalarımızın fayldan oxunması və NATD

Yazdığımız Firewall qaydalarının sistem yenidənyüklənməsindən sonra avtomatik işə düşməsini istəsek, onu lazımi təyinatı olan fayla əlavə etmək lazımdır. Deyək ki, sadə NAT quraşdırırıq və bütün trafiki istənilən istiqamətə buraxırıq. Təsəvvür edək ki, serverimizin iki şəbəkə kartı var və PUBLİC İP ünvanımız **192.168.121.129**-dur. Public tərəf **em0** Internet, daxili tərəf **em1** istifadəçilər üçün gateway. Serverimiz Router rejimində çalışır. Topoliyamız aşağıdakı şəkildəki kimidir:



**/etc/ipfw.conf** firewall quraşdırma faylimizn məzmunu aşağıdakı kimi olacaq:

**ipfw add 00050 divert 8668 ip4 from any to any via em0**

**ipfw add 00051 divert 8668 ip4 from any to me in recv em0** - İstənilən IPv4 trafiki **em0** şəbəkə kartının girişində özümüzə qayıdarsa, NAT edin.

```
ipfw add 00100 allow ip from any to any via lo0
ipfw add 00200 deny ip from any to 127.0.0.0/8
ipfw add 00300 deny ip from 127.0.0.0/8 to any
ipfw add 40000 deny ip from any to any mac any 00:0c:29:26:28:c2/48 in layer2 recv em1
- Client-imizi 2-ci səviyyədə MAC ünvanına görə bağlayırıq. Ancaq /etc/sysctl.conf faylına net.link.ether.ipfw=1 sətrini əlavə etməyi unutmayın. Sonra bu qaydanı silin ki, NAT-in işlədiyini client-dən test edə biləsiniz.
```

```
ipfw add 65000 allow ip from any to any
```

/etc/rc.conf StartUP quraşdırma faylımızda aşağıdakı sətirləri əlavə edirik:

```
ifconfig_em0="inet 192.168.121.129 netmask 255.255.255.0 -rxcsum"
- em0 geri qayidan paketlərin düzgünlüyünü sayırmış
defaultrouter="192.168.121.2"           - Serverimiz üçün gateway
ifconfig_em1="inet 10.0.0.1 netmask 255.255.255.0"
- Müştərilər üçün Gateway
gateway_enable="YES"                    - Serverimizi Router rejiminə keçiririk.
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/ipfw.conf"         - IPFW startup-da bütün qaydalarını göstərdiyimiz
natd_enable="YES"                         fayldan yükleyir.
natd_interface="em0"
```

#### Static NAT və PAT

Static NAT – əgər sizin PUBLİC şəbəkə kartınızda bir neçə dünya IP-si mövcud olarsa və siz daxili şəbəkənizdə işləyən bir istifadəçiye bütövlükdə dünya IP ünvanı təyin etmək istəsəniz, bu işi o görəcək.

PAT(Port Address Translation) – bu halda isə sizin daxili şəbəkədə olan hansısa IP ünvanının seçilmiş portunu dünyadan görmək tələbi yaranarsa, bu, sizin köməyinizə çatacaq.

```
/etc/rc.conf StartUP faylında aşağıdakı sətri əlavə edirik ki, NATD servisimiz üçün quraşdırmları əlavə fayldan oxuyaq:
natd_flags="-f /etc/natd.conf"
```

`/etc/natd.conf` faylımızın məzmunu isə aşağıdakı kimi olacaq:

`use_sockets yes`

`same_ports yes`

`interface em0`

`dynamic yes`

`unregistered_only yes`

`#redirect_address 10.0.0.2 192.168.121.129`

`redirect_port tcp 10.0.0.3:10003 22`

`redirect_port tcp 10.0.0.2:3389 192.168.121.129:3389`

`#redirect_port tcp 192.168.1.3:3000-4000 3000-4000`

- Socket(2)-ni səliqə ilə yerləşdirir ki, FTP Data, ya da İRC DCC qoşulmalarını yollasın. Bu, çoxlu sistem resursu istifadə edəcək, ancaq qoşulmada portlarda konflikt yaranarsa, uğurlu nəticəyə təminat verir.

- Paketlərin çıxışında eyni port rəqəmlərini istifadə etməyə cəhd edəcək. Bu opsiya ilə RPC kimi protokolların yaxşı işləmə şansı artır.

- Alias ünvanı təyin etmək üçün istifadə olunur. Əgər şəbəkə kartında İP ünvanın dəyişmə imkanı olarsa, **-dynamic** opsiyası da istifadə edilməlidir. Əks halda, **-alias\_address** opsiyası istifadə edilməlidir.

- **-n**, ya da **-interface** opsiyası istifadə edilərsə, natd şəbəkə kartına keçən socket yönləndirməsini izləyəcək. Əgər İP ünvan dəyişərsə, natd öz alias ünvan konsepsiyasını dinamik olaraq dəyişəcək.

- Ancaq qeydiyyatdan keçməyən və çıxışda olan mənbə paketlərini dəyişdir. RFC **1918**-ə əsaslanaraq qeydiyyatdan keçməyən mənbə ünvanları **10.0.0.0/8**, **172.16.0.0/12** və **192.168.0.0/16**.

- **10.0.0.2** daxili ünvanın bütün portlarını **192.168.121.129** İP ünvanının üstüne yönəldəcək.

- **em0** şəbəkə kartımızın ilk İP ünvanı üzərinə **22** portla gələn müraciətləri **10.0.0.3:10003** porta yönəltsin.

- **192.168.121.129** İP ünvanına 3389-cu port ilə gələn müraciətləri **10.0.0.2:3389** üzərinə yönləndirəcək.

- Dünyada görünən ilk İP ünvanımızın üzərinə **3000-4000** aralığındakı portlara gələn müraciətləri daxili **192.168.1.3:3000-4000** port aralığına yönəldəcək.

## IPFW PIPE

IPFW vasitəsi ilə şəbəkənin boğulması üçün PIPE-dən istifadə edilir. PIPE haqda yalnız təcrübədə tez-tez istifadə edilən məqamları açıqlayırıq.

```
ipfw pipe show  
ipfw pipe 10 delete  
ipfw pipe 1 config bw 5KByte/s
```

- Mövcud PIPE sıralamasını çap edir.
- 10-cu PIPE-yə aid olan qaydanı silir.
- 1-ci PIPE quraşdırması altında tutduğumuz şəbəkə sürətini 5 kilobayt edirik.
- 1-ci PIPE quraşdırması altında tutduğumuz şəbəkə sürətini 5 kilobit edirik.

```
ipfw pipe 1 config bw 5Kbits
```

```
ipfw add 500 pipe 1 ip from 192.168.1.2 to any
```

- 500-cü qayda altında olan 192.168.1.2 IP ünvanının istənilən istiqamətə gedişini PIPE 1 sürətinə təyin edirik.

```
ipfw add 12 queue 10 ip from 192.168.1.2 to any via em0
```

- Seçilmiş IP ünvandan istənilən istiqamətə em0 şəbəkə kartından keçdikdə növbə təyin edirik.
- queue 10** – növbədə olacaq paketlərin sayını təyin edirik.

```
ipfw queue 10 config weight 6 pipe 10
```

- Burada isə 10 paket keçəsi olan növbələşməyə çəki 6 veririk.
- weight** – Növbənin üstünlüyünü təyin edir (1-100).
- Daha yüksək rəqəm daha üstün sayılır.

# IPFW Squid Transparent

İsimizi görmek için FreeBSD makinimizin **em0** şəbəkə kartı PUBLİC IP ünvanı ilə dünyaya çıkış edir. **em1** şəbəkə kartı isə daxili istifadəçilərimiz üçün susmaya görə olan şəbəkə yoludur və daxili IP ünvanı üzərində proxy server işləyəcək.

Portlarınıza yeniləyirik və lazımi paketləri yükləyirik.

Əgər siz FreeBSD firewall olan serverinizdə portlarınızın NAT vasitəsilə yönləndirilməsini istəməsəniz, **rinetd** paketi vasitəsilə bu işi görə bilərsiniz.

<b>cd /usr/ports/net/rinetd</b>	- Port ünvanına daxil oluruq.
<b>make install clean</b>	- Yükləyirik.

Squid IPFW ilə işləməsi üçün kernel kompilyasiya etməli, ya da modullardan IPFW-nu yüklemək lazımdır. Kernel-ə aşağıdakı modulları əlavə etməyiniz kifayətdir:

<b>options</b>	<b>IPFIREWALL</b>
<b>options</b>	<b>IPFIREWALL_VERBOSE</b>
<b>options</b>	<b>IPFIREWALL_VERBOSE_LIMIT=3</b>
<b>options</b>	<b>DUMMYNET</b>
<b>options</b>	<b>IPDIVERT</b>

**/etc/rc.conf** StartUP quraşdırma faylımız aşağıdakı kimi olacaq. X.X.X.X yazılan yerde siz öz PUBLİC İP ünvanınızı qeyd etməlisiniz:

```
ifconfig_em0="inet X.X.X.X netmask 255.255.255.0"
ifconfig_em1="inet 10.0.0.1 netmask 255.255.255.0"
defaultrouter="X.X.X.1"
sshd_enable="YES"
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/firewall.conf"      - Transparent proxy üçün qaydalar faylı
gateway_enable="YES"                      - Serverimiz router rejimindədir.
natd_enable="YES"                          - NAT-ı aktiv edirik.
natd_interface="em0"                       - NAT şəbəkə kartı em0
```

**/etc/firewall.conf** faylı transparent və NAT quraşdırmları üçün qaydalar faylidir (X.X.X.X yazılan yerlərdə PUBLİC İP ünvanla əvəz etməlisiniz):

```
ipfw add 00001 allow ip from any to me dst-port 22 keep-state
ipfw add 00002 allow ip from any to me dst-port 80 keep-state
ipfw add 00010 allow ip from any to any via em1
ipfw add 00020 allow ip from any to any via lo0
ipfw add 00030 deny ip from any to 127.0.0.0/8
ipfw add 00040 deny ip from 127.0.0.0/8 to any
ipfw add 00050 fwd 10.0.0.1,3129 tcp from 10.0.0.0/24 to any dst-port 80 out via em0
ipfw add 00060 divert 8668 ip from any to any in via em0
ipfw add 00070 check-state
ipfw add 00100 skipto 400 icmp from any to any keep-state
ipfw add 00105 skipto 400 udp from any to any dst-port 123 out via em0 keep-state
ipfw add 00110 skipto 400 udp from any to any dst-port 53 out via em0 keep-state
ipfw add 00111 skipto 400 tcp from any to any dst-port 53 out via em0 setup keep-state
ipfw add 00140 skipto 400 all from 10.0.0.0/24 to any 4899 out via em0 setup keep-state
ipfw add 00150 skipto 400 all from 10.0.0.0/24 to any 3389 out via em0 setup keep-state
ipfw add 00160 skipto 400 all from 10.0.0.0/24 to any 25 out via em0 setup keep-state
ipfw add 00170 skipto 400 all from 10.0.0.0/24 to any 110 out via em0 setup keep-state
ipfw add 00180 skipto 400 tcp from 10.0.0.0/24 to any dst-port 443 out via em0 setup keep-state
ipfw add 00190 skipto 400 all from X.X.X.X to any out via em0 setup keep-state
ipfw add 00200 deny ip from 192.168.0.0/16 to any in via em0
ipfw add 00201 deny ip from 172.16.0.0/12 to any in via em0
ipfw add 00202 deny ip from 10.0.0.0/8 to any in via em0
```

```
ipfw add 00203 deny ip from 127.0.0.0/8 to any in via em0
ipfw add 00204 deny ip from 0.0.0.0/8 to any in via em0
ipfw add 00205 deny ip from 169.254.0.0/16 to any in via em0
ipfw add 00206 deny ip from 192.0.2.0/24 to any in via em0
ipfw add 00207 deny ip from 204.152.64.0/23 to any in via em0
ipfw add 00208 deny ip from 224.0.0.0/3 to any in via em0
ipfw add 00215 deny tcp from any to any dst-port 113 in via em0
ipfw add 00220 deny tcp from any to any dst-port 137 in via em0
ipfw add 00221 deny tcp from any to any dst-port 138 in via em0
ipfw add 00222 deny tcp from any to any dst-port 139 in via em0
ipfw add 00223 deny tcp from any to any dst-port 81 in via em0
ipfw add 00300 allow icmp from any to X.X.X.X in via em0 icmptypes 0,8,11 limit src-addr 2
ipfw add 00310 allow tcp from any to X.X.X.X dst-port 80 in via em0 setup limit src-addr 2
ipfw add 00320 allow tcp from any to X.X.X.X dst-port 22 in via em0 setup limit src-addr 2
ipfw add 00330 allow tcp from any to X.X.X.X dst-port 25 in via em0 setup limit src-addr 2
ipfw add 00340 allow tcp from any to X.X.X.X dst-port 110 in via em0 setup limit src-addr 2
ipfw add 00350 allow tcp from any to X.X.X.X dst-port 4899 in via em0 setup limit src-addr 2
ipfw add 00360 allow ip from any to any established
ipfw add 00399 deny log logamount 100 ip from any to any
ipfw add 00400 divert 8668 ip from any to any out via em0
ipfw add 00410 allow ip from any to any
ipfw add 00999 deny log logamount 100 ip from any to any
```

AMP yüklenməsini 10-cu bölümün "**FreeBSD 10.1 x64 AMP(Apache MySQL PHP)**" başlığında oxumalısınız.

İstifadəçilərin internet trafikinin program təminatı səviyyəsində şügəcdən keçməsini idarə etmək istəsek, bizim köməyimizə Squid çatacaq. Squid istifadəçilərin saytlara girişini, sürətini, axtarış motorlarını və s. idarə edə bilir. Squid adı proxy, reverse proxy və transparent proxy rejimlərində işləyə bilir. Ancaq Squid həddən artıq böyükdür və onun bütün imkanlarının tam açıqlanması bu kitabın yerləşə bilməz. Bu səbəbdən biz Squidi susmaya görə işlek vəziyyətə gətirib Transparent rejimdə işlədəcəyik. Transparent proxy-nin adı proxy serverdən fərqi ondan ibarətdir ki, adı proxy serverdə istifadəçinin WEB broswerində proxy serverin IP ünvanı və portu sərt şəkildə öncədən qeyd edilməlidir. Transparent proxy-də istifadəçi tərefdən heç bir iş görülmür.

**Qeyd:** Unutmayın ki, transparent proxy-də yalnız 80-ci porta gedən müraciətlər şügəcdən keçir, adı proxy-də isə bütün trafik.

```
cd /usr/ports/www/squid
make config
```

- Port ünvanına daxil oluruq.  
- Modullardan mütləq IPFW seçilməlidir, ona görə ki, transparent proxy-mizin üzərinə trafiki onun sayəsində yönləndirəcəyik.

squid-3.4.12	
x [x] <b>ARP_ACT</b>	<b>ARP/MAC/EUI based authentication</b>
x [x] <b>AUTH_KERB</b>	Install Kerberos authentication helpers
x [ ] <b>AUTH_LDAP</b>	Install LDAP authentication helpers
x [x] <b>AUTH_NIS</b>	Install NIS/YP authentication helpers
x [ ] <b>AUTH_SASL</b>	Install SASL authentication helpers
x [ ] <b>AUTH_SMB</b>	Install SMB auth. helpers (req. Samba)
x [ ] <b>AUTH_SQL</b>	Install SQL based auth (uses MySQL)
x [ ] <b>CACHE_DIGESTS</b>	Use cache digests
x [x] <b>DEBUG</b>	Build with extended debugging support
x [x] <b>DELAY_POOLS</b>	Delay pools (bandwidth limiting)
x [x] <b>DNS_HELPER</b>	Use external dnsserver processes for DNS
x [x] <b>DOCS</b>	Build and/or install documentation
x [ ] <b>ECAP</b>	Loadable content adaptation modules
x [ ] <b>ESI</b>	ESI support
x [x] <b>EXAMPLES</b>	Build and/or install examples
x [ ] <b>FOLLOW_XFF</b>	Support for the X-Following-For header
x [x] <b>FS_AUFS</b>	AUFS (threaded-io) support
x [x] <b>FS_DISKD</b>	DISKD storage engine controlled by separate service
x [ ] <b>FS_ROCK</b>	ROCK (unstable)
x [x] <b>HTCP</b>	HTCP support
x [ ] <b>ICAP</b>	the ICAP client
x [x] <b>ICMP</b>	ICMP pinging and network measurement
x [x] <b>IDENT</b>	Ident lookups (RFC 931)
x [ ] <b>IPV6</b>	IPv6 protocol support
x [x] <b>KQUEUE</b>	Kqueue(2) support
x [ ] <b>LARGEFILE</b>	Support large (>2GB) cache and log files
x [ ] <b>LAX_HTTP</b>	Do not enforce strict HTTP compliance
x [ ] <b>NETTLE</b>	Nettle MD5 algorithm support
x [x] <b>SNMP</b>	SNMP support
x [x] <b>SSL</b>	SSL gatewaying support
x [x] <b>SSL_CRTD</b>	Use ssl_crtd to handle SSL cert requests
x [ ] <b>STACKTRACES</b>	Enable automatic backtraces on fatal errors
x [ ] <b>TP_IPF</b>	Transparent proxying with IPFilter
x [x] <b>TP_IPFW</b>	Transparent proxying with IPFW
x [x] <b>TP_PF</b>	Transparent proxying with PF
x [ ] <b>VIA_DB</b>	Forward/Via database
x [x] <b>WCCP</b>	Web Cache Coordination Protocol
x [x] <b>WCCPV2</b>	Web Cache Coordination Protocol v2

```
make install clean
```

- Yükləyirik.

**Qeyd:** Yüklənmədən sonra diqqətlə son sətirləri oxuyuruq.

Squidi işə salmazdan öncə onun keş qovluqlarını yaratmaq lazımdır. Ancaq squidin **3.4** versiyasında susmaya görə olan quraşdırma faylında (yəni **/usr/local/etc/squid/squid.conf**) **11** və **12**-ci sətirlərin (boş olan **acl localnet src** sətirləri) qarşısına şərh yerləşdirib yadda saxlayaraq çıxməq lazımdır. Bunu etməsəniz, keş qovluqlar yaranmayacaq və səhv çap ediləcək.

**squid -z** - cahce qovluqlarını yaradırıq.

**/usr/local/etc/squid/squid.conf** faylında **http\_port 3128** sətrini tapırıq və qarşısına şərh yerləşdiririk. Həmin sətirdən sonra aşağıdakı sətri əlavə edirik ki, transparent rejimi işə düşə bilsin (**10.0.0.1** İP ünvanı istifadəçilərin susmaya görə olan gateway-idir). Həmçinin jurnallarımızın yığılması üçün **access.log** faylini işə salırıq.

**http\_port 10.0.0.1:3129 transparent**  
**access\_log daemon:/var/squid/logs/access.log squid**

**touch /var/squid/logs/access.log** - İstifadəçilərimizin jurnal sətirlərini özündə toplayacaq jurnal faylini yaradırıq.

**chown squid:squid /var/squid/logs/access.log** - Squid-in fayla hüququ olması üçün yetkini veririk.

**squid -f /usr/local/etc/squid/squid.conf -k parse** - Squid quraşdırma faylında sintaksisin düzgünüyü yoxlayırıq. Nəticə səhvsiz olmalıdır.

**echo 'squid\_enable="YES"' >> /etc/rc.conf** - Squid daemonu StartUP faylımiza əlavə edirik ki, yenidənyüklənmədən sonra avtomatik işə düşsün.

**/usr/local/etc/rc.d/squid start** - Squid daemonu işə salırıq.

**squidclient -h 10.0.0.1 -p 3129 -g 1 mail.ru** - Test üçün proxy serverimizin üzərindən 1 ədəd paket yollayıq və aşağıdakı kimi nəticə əldə edirik.

Sending HTTP request ... done.

2015-04-03 04:38:31 [1]: 0.000 secs, -1.000000 KB/s  
1 requests, round-trip [secs] min/avg/max = 0.000/0.000/0.000

**Qeyd:** Squid Transparent proxy serverimizin istifadəçilərini süzgəcdən keçirəcək və hər kəsə lazım olan yetkiləri mənimsədəcək, ancaq proxy server istifadəçilər haqqında hesabatı bizə verə bilmir. Squid istifadəçilər haqqında jurnalları **access.log** faylinə toplayır. Bu jurnal faylinı digər programlar vasitəsilə analiz edib, web serverimizə ötürə bilərik. Misal olaraq, onlardan **LightSquid**(`/usr/ports/www/lightsquid`) və **SARG** deyə bilərik. Mövzumuzda SARG-ı açıqlayıraq.

```
cd /usr/ports/www/sarg/
```

- Port ünvanına daxil oluruq.

**make config**

- Lazımi modulları seçirik.

sarg-2.3.9

**make install**

- Yüklevirik.

Adı halde Squid tərəfindən əldə etdiyimiz access.log faylini SARG tərəfindən emal etsək, yalnız IP ünvanlı istifadəçi nəticəsi əldə etmiş olasıyıq. Lakin inzibatçı üçün bu, diskomfortdur, ona görə ki, WEB browserdə çap edilən IP ünvanlarının hansı şəxse mənsub olduğunu hər dəfə araşdırmaq lazımlı olacaq. Bu səbəbdən də biz səliqə ilə jurnal faylında olan istifadəçiləri aid olduqları şöbələrə görə access.log faylından filtr edib, ayrı adla başqa fayla yazacaqıq. Nəticədə, fərqli şöbələr üçün əldə etdiyimiz faylları ayrı-ayrı analiz edib WEB serverimizin **PUBLIC HTML** qovluquna ötürürəcəyik.

```
cd /usr/local/etc/sarg
```

- SARG-ın əsqs qurqasdırma qovluğuna daxil olaq.

```
cp sarg.conf sarg.conf.shovel
```

- Əsas quraşdırma faylini **shobel** adı ilə başqa fayla nüsxələyirik.

cp sarg.conf sarg.conf-shoe?

- Fyni işi **shöhe?** üçün edirik

`/etc/crontab` zamanlayıcı faylımızda 3 adəd fərqli faylin təyin etdiyimiz zamanlarda işə düşməsi üçün aşağıdakı setirleri əlavə edirik:

# Sarı 108 işa salan (iñurnalları süzgacdan kecirir)

root /root/logs script.sh

```
# Sarg-ı işe salan(Her bir şöbənin quraşdırmasını oxuyur və jurnalları generasiya edir).
35      18      *      *      *      root      /root/sarg.script.sh
```

```
# İş bitdikdən sonra access.log faylini təmizləyir ki, yeni gün üçün yeni
# vaxt möhürü və yeni jurnal sətirləri olan məlumatları oxuyaq.
37      1      *      *      *      root      /root/logtemizleyen
```

**/root/logs.script.sh** adlı skript yaradırıq və içində aşağıdakı sətirləri əlavə edirik. Skriptimiz oncə köhnə süzgəcdən keçirmiş şöbələr üçün faylları təmizləyir və ardınca yeni vaxt möhürlü yeni gün üçün olan jurnal sətirlərini süzgəcdən keçirib təyinatına uyğun olan fayllara yazar.

```
cat '/dev/null' > /usr/home/loglar/shobel.log
cat '/dev/null' > /usr/home/loglar/shobe2.log
# Şöbə1-in işçiləri
cat /var/squid/logs/access.log | grep 10.0.0.2 >> /usr/home/loglar/shobel.log
cat /var/squid/logs/access.log | grep 10.0.0.3 >> /usr/home/loglar/shobel.log
# Şöbə2-in işçiləri
cat /var/squid/logs/access.log | grep 10.0.0.10 >> /usr/home/loglar/shobe2.log
cat /var/squid/logs/access.log | grep 10.0.0.11 >> /usr/home/loglar/shobe2.log
```

**/root/sarg.script.sh** faylı yaradırıq və məzmununa aşağıdakı iki sətir əlavə edərək bildiririk ki, hər şöbə üçün öz quraşdırılmalarına əsasən jurnalları generasiya etsin.

```
/usr/local/bin/sarg -f /usr/local/etc/sarg/sarg.conf.shobel
/usr/local/bin/sarg -f /usr/local/etc/sarg/sarg.conf.shobe2
```

**/root/logtemizleyen** bu faylda isə Squid-in jurnal faylini təmizləyirik ki, yeni gün üçün yeni vaxt möhürü olan jurnal sətirləri yenidən yazılmışa başlasın.

```
cat '/dev/null' > /var/squid/logs/access.log
mkdir /usr/home/loglar/                                - Yaranacaq jurnal fayllar üçün qovluq yaradırıq.
```

**root** adlı istifadəçi üçün hər üç fayla tam yetki veririk:

```
chmod 700 /root/logs.script.sh
chmod 700 /root/sarg.script.sh
chmod 700 /root/logtemizleyen
```

Sonra test etmək üçün hazırladığımız istifadəçi maşınınında önce faylda qeyd etdiyimiz IP ünvanları növbə ilə təyin edib fərqli saytlara daxil olaq ki, jurnal sətirləri yiğilsin. Bu, bizə emal etdikdə hər bir istifadəçinin hansı saytlara daxil olmasını görmək üçün lazımdır.

```
tail -f /var/squid/logs/access.log
```

- Squid jurnal faylında onlayn rejimdə jurnallara baxa bilərik.

```
cd /usr/local/etc/sarg
```

- SARG-ın quraşdırma qovluğuna daxil oluruq.

```
ee shobel
```

- Faylda olan istifadəçilər(IP ünvanlar) WEB-də görünməyəcək. Yəni şöbəl üçün olan faylda özünə aid olan istifadəçilərdən başqa, digər bütün şöbələrin işçilərini yazırıq.

```
#shobel
```

```
#shobe2
```

```
10.0.0.10
```

```
10.0.0.11
```

```
ee shobe2
```

- Faylda olan istifadəçilər(IP ünvanlar) WEB-də görünməyəcək. Eynilə öz istifadəçilərindən başqa, bütün istifadəçiləri fayla əlavə edirik.

```
#shobel
```

```
10.0.0.2
```

```
10.0.0.3
```

```
#shobe2
```

```
ee user.txt
```

- Şəbəkəmizdə olan istifadəçilərin IP ünvanlarını ada çevirmək üçün istifadə edirik.

```
#Şöbə1
```

```
10.0.0.2 Bay Moh(10.0.0.2)
```

```
10.0.0.3 Claudia Fisher(10.0.0.3)
```

```
#Şöbə2
```

```
10.0.0.10 John Mayers(10.0.0.10)
```

```
10.0.0.11 Andrea Andreas(10.0.0.11)
```

```
ee sarg.conf.shobel
```

- SARG-ın şöbəl üçün oxuduğu quraşdırma faylinin məzmununa aşağıdakı satırları əlavə edirik.

```
access_log /usr/home/loglar/shobel.log
```

```
graphs yes
```

```
title "Test Ders statistikası"
```

```
output_dir /usr/local/www/apache24/data/shobel
resolve_ip yes
topuser_sort_field BYTES reverse
user_sort_field BYTES reverse
exclude_users /usr/local/etc/sarg/shobel
overwrite_report yes
usertab /usr/local/etc/sarg/user.txt
charset Koi8-r
show_sarg_logo no
```

**ee sarg.conf.shobe2** - Eynilə şöbə2 üçün SARG-ın oxuduğu quraşdırma faylinin məzmununu aşağıdakı kimi edirik.

```
access_log /usr/home/loglar/shobe2.log
graphs yes
title "Test Ders statistikasi"
output_dir /usr/local/www/apache24/data/shobe2
resolve_ip yes
topuser_sort_field BYTES reverse
user_sort_field BYTES reverse
exclude_users /usr/local/etc/sarg/shobe2
overwrite_report yes
usertab /usr/local/etc/sarg/user.txt
charset Koi8-r
show_sarg_logo no
```

**/root/logs.script.sh** - Skripti işə salırıq ki, jurnallarımızı şöbələrə görə ayırib test edə bilək.

**/root/sarg.script.sh** - Skripti işə salırıq ki, hər şöbənin jurnal sətirlərini analiz edib WEB serverimizə ötürək.

```
SARG: Records in file: 2316, reading: 100.00%
SARG: Successful report generated on
/usr/local/www/apache24/data/shobel/2015Apr03-2015Apr03
SARG: Records in file: 1053, reading: 100.00%
SARG: Successful report generated on
/usr/local/www/apache24/data/shobe2/2015Apr03-2015Apr03
```

Sonda serverimizin IP ünvanını WEB browserdə daxil edirik və nəticəyə baxırıq. Aşağıdakı şəklə uyğun olmalıdır:

The screenshot shows a web browser window with the URL [192.168.121.134/shobe2/2015Apr03-2015Apr03/index.html](http://192.168.121.134/shobe2/2015Apr03-2015Apr03/index.html). The page title is "Test Ders statistikası". It displays a table of top users with the following data:

NUM	USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT	ELAPSED TIME	MILLISEC	%TIME
1	Leyla Hacıyeva(10.0.0.10)	658	12.87M	69.56%	0.00%	100.00%	00:02:47	167.324 54.81%
2	Uzeyir Topchubashev(10.0.0.11)	395	5.63M	30.44%	0.02%	99.98%	00:02:17	137.948 45.19%
	TOTAL	1.05K	18.50M	0.01%	99.99%	00:05:05	305.272	
	AVERAGE	526	9.25M			00:02:32	152.636	

Generated by sarg-2.3.9 Sep-21-2014 on Apr/03/2015 14:22

### Squid File Descriptor problemi

Hər bir Squid istifadəçisi açdığı linklər üçün əməliyyat sistemində olan fayl deskriptorlardan istifadə edir. Bu say isə kernel-də susmaya görə məhdud olur. İstifadəçi sayı çoxaldıqda sizə jurnal faylda warning-lər çap edilə bilər. Bunun həlli yolu həmin deskriptorların sayını artırmaqdır.

Əgər Squid-in '**cache.log**' faylında '**WARNING! Your cache is running out of file descriptors**' səhvini görsəniz, demək ki, sizin kernel fayl deskriptorlarının istifadə sayına görə məhdudlaşdırılmışdır.

Bu problemin həlli üçün kernel-in lazımi parametrləri dəyişdirilməlidir.

**Qeyd:** squidclient səhvleri tapmaq üçün keş idarə edicisinə müraciət edir və o da susmaya görə olan **http\_port 3128**-ə müraciət edir. Unutmayın ki, həmin sətri aktiv etməsəniz, səhvleri görə bilməyəcəksiniz.

Squid-in özünün nə qədər File Deskriptoru istifadə etdiyini tapmaq üçün aşağıdakı əmrənən istifadə etmək olar:

```
squidclient mgr:info | grep 'file descri' - 'mgr:info' - Squid-in sistemdən istifadə etdiyi bütün resursları çap edirik.
```

Əgər audentifikasiya varsa, istifadəçi adı və şifrə ilə yoxlayırıq.

```
squidclient -p 3129 -h 127.0.0.1 -u USER -w 'parol' mgr:info | grep 'file descri'
```

CLI-dən limitləri artırırıq.

```
sysctl kern.maxfiles=8192
```

- Bir proses üçün maksimum açıla biləcək faylların sayı **8192**

```
sysctl kern.maxfilesperproc=65535 - Maksimum açıla biləcək faylların sayı 65535
```

```
sysctl net.inet.ip.portrange.last=65535 - Port aralıqlarını da artırırıq.
```

```
sysctl net.inet.ip.portrange.first=1024 - Port aralıqlarını da artırırıq.
```

Eyni ilə kernel StartUP faylinə əlavə edirik:

```
echo 'kern.maxfilesperproc=8192' >> /etc/sysctl.conf
```

```
echo 'kern.maxfiles=65535' >> /etc/sysctl.conf
```

```
/usr/local/etc/rc.d/squid restart - Sonda squidı yenidən işə salırıq.
```

```
squidclient mgr:info | grep 'file descri' - Və təkrar nəticəyə baxırıq.
```

```
Maximum number of file descriptors: 117477
```

```
Available number of file descriptors: 117465
```

```
Reserved number of file descriptors: 100
```

# IPFW PF Fail2Ban

Bu, bize çoxlu hostların müəyyən bir sayıda düzgün sayılmayan müraciətlərini blocklamağa kömək edəcək. Yəni bu program təminatı istənilən servisin jurnallarını oxumaq imkanına malikdir. Misal olaraq, bu dəfə SSH-a gələn müraciətlərin sayı 3-ü aşarsa, **Fail2Ban** edəcəyik. Bizim halda **IPFW** firewall-dan istifadə edəcəyik. Amma hər hal üçün kernel-i **PF** ilə də kompilyasiya edək ki, bize lazım ola bilər.

Öncə kernel-imizi kompilyasiya edirik və StartUP-a əlavə edirik.

**ee /sys/amd64/conf/GENERIC** - Faylıın sonuna aşağıdakı sətirləri əlavə edirik.

```
# ipfw
options      IPFIREWALL
options      IPFIREWALL_VERBOSE
options      IPFIREWALL_VERBOSE_LIMIT=3
options      DUMMYNET
options      IPFIREWALL_FORWARD
options      IPFIREWALL_NAT
options      LIBALIAS
```

```
# pf
options      HZ=1000
device       pf
device       pflog
device       pfsync
options      ALTQ
options      ALTQ_CBQ
options      ALTQ_RED
options      ALTQ_RIO
options      ALTQ_HFSC
options      ALTQ_CDNR
options      ALTQ_PRIQ
```

```
cd /usr/src/ ; make buildkernel
cd /usr/src/ ; make installkernel
```

- Kernel-i kompilyasiya edirik.
- Kompilyasiya edilmiş kernel-i yükleyirik.

```
portsnap fetch extract update
reboot
```

- Bütün portları yenileyirik.
- Sistemimize yenidənyüklənmə əmri veririk ki, yeni kernel-imiz yüklənsin.

```
ee /etc/rc.conf
firewall_logging="YES"
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/ipfw.conf"
pf_enable="YES"
pflog_enable="YES"
```

- StartUP faylımiza aşağıdakı sətirləri əlavə edirik.

Bu IPFW Firewall qaydaları faylında **100** rəqəmli cədvəldə **fail2ban**-dan gələn IP ünvanları saxlayacaq, ona görə ki, bu cədvəldə həmin IP ünvanları **block** edilir.

```
ee /etc/ipfw.conf
```

- IPFW StartUP skriptinin məzmunu aşağıdakı kimi olacaq.

```
ipfw add 10 deny ip from table'(100)' to any
ipfw add 65000 allow ip from any to any
```

Fail2Ban-i yükleyək.

```
cd /usr/ports/security/py-fail2ban
```

- Port ünvanına daxil oluruq.

```

make install - Yükləyirik.

echo 'fail2ban_enable="YES"' >> /etc/rc.conf - StartUP-a əlavə edirik.

StartUP jurnal rotasiyası faylimizə əlavə edirik. CLI-dən əmri daxil edirik.
echo "/var/log/fail2ban.log 600 7 200 * >> /etc/newsyslog.conf

ee /usr/local/etc/fail2ban/jail.local - Faylin işinə aşağıdakı məzmunu əlavə edirik.

[DEFAULT]
ignoreip = 127.0.0.1 10.0.0.1
# BAN ediləcək ünvanın saniyelərlə olan vaxtı
bantime = 600

# Yoxlanış müddəti, hansı ki, bu aralıqda hadisə təkrarlanı bilər.
# Belə hal olan kimi tutub bağlayacaq.
findtime = 900

# Maksimum neçə dəfə qanun pozuntusu ola bilər.
maxretry = 3

# Jurnalları araştırma metodikası
backend = auto

[ssh-ipfw]
enabled = true
# Filtrləri /usr/local/etc/fail2ban/filter.d/bsd-sshd.conf faylından oxuyub istifadə etmək.
filter = bsd-sshd
# Mənimsemə üçün /usr/local/etc/fail2ban/action.d/bsd-ipfw.conf faylından istifadə etmək.
# Kvadrat mötərizələrdə dəyişənin mənasını təyin edirik.
# Bizim halda göstəririk ki, tablearg22 olan 100 table-na əlavə et.
action = bsd-ipfw[table=100, tablearg=22]
logpath = /var/log/auth.log
maxretry = 3
# 7 sutka ban edəcək.
bantime = 604800

```

```
/usr/local/etc/fail2ban/filter.d/bsd-sshd.conf
```

faylı işə susmaya görə qalacaq.

Həmçinin '/usr/local/etc/fail2ban/action.d/bsd-ipfw.conf' faylında aşağıdakı 2 sətri uyğun formaya gətiririk.

```
actionban = ipfw table <table> add <ip> <tablearg>
actionunban = ipfw table <table> delete <ip>
```

```
/usr/local/etc/rc.d/fail2ban start      # İşə salırıq.
```

Test üçün həmin serverin IP ünvanına səhv şifrələrlə SSH sessiya açıb qoşulmaq istəsək, aşağıdakı jurnal sətirləri fail2ban-də göstərilməlidir.

```
tail -f /var/log/fail2ban.log
```

```
2013-12-06 17:19:53,407 fail2ban.server : INFO  Exiting Fail2ban
2013-12-06 17:19:53,850 fail2ban.server : INFO  Changed logging target to /var/
log/fail2ban.log for Fail2ban v0.8.11
```

```
2013-12-06 17:19:53,851 fail2ban.jail   : INFO  Creating new jail 'ssh-ipfw'
2013-12-06 17:19:53,851 fail2ban.jail   : INFO  Jail 'ssh-ipfw' uses poller
2013-12-06 17:19:53,860 fail2ban.jail   : INFO  Initiated 'polling' backend
2013-12-06 17:19:53,861 fail2ban.filter : INFO  Added logfile = /var/log/auth.log
2013-12-06 17:19:53,862 fail2ban.filter : INFO  Set maxRetry = 3
2013-12-06 17:19:53,863 fail2ban.filter : INFO  Set findtime = 900
2013-12-06 17:19:53,863 fail2ban.actions: INFO  Set banTime = 604800
2013-12-06 17:19:53,891 fail2ban.jail   : INFO  Jail 'ssh-ipfw' started
2013-12-06 17:21:06,949 fail2ban.actions: WARNING [ssh-ipfw] Ban 10.50.19.4
```

Əgər WEB serverləri qorumaq istəsək, '/usr/local/etc/fail2ban/jail.local' faylinə aşağıdakı sətirləri əlavə etməklə, PF ilə birgə işləyib bizim nGinx-i fail2ban etsin.

```
[nginx]
enabled = true
port = http,https
filter = apache-auth
action = pf[table=webhack]
#sendmail[name=WEBServ hacking,dest=admin@localhost, sender=fail2ban@localhost]
logpath = /var/log/nginx-access.log
maxretry = 6
```

```
[nginx-noscript]
enabled = true
port = http,https
filter = apache-noscript
action = pf[table=webhack]
logpath = /var/log/nginx-error.log
maxretry = 6

[nginx-overflows]
enabled = true
port = http,https
filter = apache-overflows
action = pf[table=webhack]
logpath = /var/log/nginx-error.log
maxretry = 2
```

Bu halda '`/etc/pf.conf`' faylı aşağıdakı kimi olacaq.

```
table <webhack> file "/etc/firewall/bruteforce_attackers" persist
block in quick inet from <webhack> to any
```

Həmçinin '`/usr/local/etc/fail2ban/action.d/pf.conf`' faylında aşağıdakı sətirlər mövcud olmalıdır.

```
actionban = /sbin/pfctl -t <tablename> -T add <ip>/32
actionunban = /sbin/pfctl -t <tablename> -T delete <ip>/32
```

# IPFilter firewall

IPF kimi tanınan IPfiltr həmçinin açıq kodlu firewall-dır və müxtəlif əməliyyat sistemlərində işləyə bilir. Bunlardan FreeBSD, NetBSD, OpenBSD və Solaris-in adını çəkmək olar. IPFiltr kernel-ə uyğunlaşmışdır və NAT mexanizmi istifadəçi proqramları tərəfindən idarə və monitoring edilə bilər. Firewall qaydaları **ipf** vasitəsilə, nat qaydaları isə **ipnat** vasitəsilə yazılı və silinə bilər. İşlek statistikalara **ipfstat** vasitəsilə baxıla bilər. **ipmon** vasitəsilə isə IPfilter-in gördüyü işləri sistem jurnal fayllarına yaza bilərik.

IPF əslində "**the last matching rule wins**" (son uyğun olan qayda qalib gəlir) prinsipi ilə yazılıb və yalnız stateless qaydalar istifadə edir. Sonradan IPF genişləndirilib və **quick**, **keep state** opsiyaları artırılmışdır. Detallı məlumat əldə etmək üçün isə <http://www.phildev.net/ipf/index.html> linkinə və <http://marc.info/?l=ipfilter> linkinə müraciət edə bilərsiniz.

IPFilter-i susmaya görə KERNEL-in içine əlavə etmək istəsəniz, aşağıdakı opsiyaları kernel fayliniza əlavə etməlisiniz:

**options IPFILTER**

**options IPFILTER\_LOG**

**options IPFILTER\_LOOKUP**

**options IPFILTER\_DEFAULT\_BLOCK**

- IPFiltr-i işə salırıq.

- IPF jurnallamani işə salırıq (hər bir log açar sözü istifadə edilən qaydalar üçün virtual alət olan ip paket jurnallamani istifadə edəcək).

- IP axtarışını sürətli eləmək üçün IP hovuzlarını işə salır.

- Susmaya görə olan iş prinsipi hamını blok edir. Yəni öz qaydaları ilə üst-üstə düşməyən istenilən paketi blok edəcək.

IPF-in sistem yenidənyüklənməsində avtomatik işə düşməsi üçün aşağıdakı sətirləri **/etc/rc.conf** StartUP faylinə əlavə etmək lazımdır. Bu sətirlər həmçinin jurnallamani işə salır və susmaya görə bütün trafikə izin verir. Susmaya görə olan siyaseti kernelsiz bağlı saxlamaq istəsəniz, qaydalar faylinin sonuna **block all** sətrini əlavə etməlisiniz.

<b>ipfilter_enable="YES"</b>	- ipf firewall-ı işə salırıq.
<b>ipfilter_rules="/etc/ipt.rules"</b>	- Qaydaları mətn fayldan oxuyuruq.
<b>ipmon_enable="YES"</b>	- IP monitor log-u işə salırıq.
<b>ipmon_flags="-Ds"</b>	- -D daemon rejimində işə salırıq. - -s Sysloga jurnallayıraq. - -v tcp pəncərəsini jurnallayıraq, ack, seq - -n IP və portları adlara xəritələyirik.

NAT funksionallığını işə salmaq üçün **/etc/rc.conf** faylinə aşağıdakı sətirləri əlavə etmək lazımdır:

<b>ipnat_enable="YES"</b>	- ipnat funksiyasını işə salırıq.
<b>ipnat_rules="/etc/iptnat.rules"</b>	- ipnat üçün təyin edilən qaydalar.

Qaydaları yüklemək üçün ipf-ə faylı göstərmək lazımdır. Aşağıdakı əmrlə mövcud işlək qaydalarla faylda olan yeni qaydalar əvəz ediləcək:

<b>ipf -Fa -f /etc/ipt.rules</b>	- <b>-Fa</b> hər şeyi sil. - <b>-f</b> bu fayldan oxu.
----------------------------------	---

Bu hissədə statusa baxan İPF qaydaları açıqlanır. Yadda saxlayın ki, qaydaları yazdıqdə quick açar sözündən istifadə edilməzsə, hər bir qayda ardıcılıqla oxunacaq və son uyğun gələni də mənimsədiləcək. Bu, o deməkdir ki, əgər paket ilk qaydada buraxılsa belə, sonrakı qaydada blok edilirsə, buraxılmayacaq. Qaydaların nüsxələrini **/usr/share/examples/iptfilter** qovluğundan tapa bilərsiniz. Faylda olan şərh simvolu '#'-dir. Sətrin əvvəli və sonunda ola bilər. Böyük hərflərlə yazılan hissələrdə tələb edilən resurslar(port, mənbə və ya mənsəb), kiçik hərflər isə resursu təyin edən dəyişənlərdir.

**ACTION DIRECTION OPTIONS proto PROTO\_TYPE from SRC\_ADDR SRC\_PORT to DST\_ADDR DST\_PORT TCP\_FLAG|ICMP\_TYPE keep state STATE**

Sıra ilə bütün açar sözləri və onların opsiyalarını açıqlayaq.

#### **ACTION**

Əgər paket bu qayda ilə üst-üstə düşərsə, açar söz görüləcək işin nə olacağını təyin edir. Aşağıdakı işlər mənimsədilə bilər:

**block:** Paketləri tutur, məhv edir.

**pass:** Paketə izin verir, buraxır.

**log:** Jurnal yazını generasiya edir.

**count:** Paketləri və baytları sayıb bize göstərir. Bunun sayesində qaydanın nə qədər tez-tez istifadə edildiyini təyin edirik.

**auth:** Paketi növbəyə salır, hansı ki, digər program tərəfindən sonra emal olunacaq.

**call:** IPF-də qurulmuş funksiyalara olan yetkini təmin edir, hansı ki, sayesində digər işlərə izin alırıq.

**decapsulate:** Paketin hissələrini səliqə ilə emal etmək üçün istənilən başlıqları silir.

## DIRECTION

Sonra hər bir qayda hərəkətin istiqamətini aşağıdakı açar sözlərinin biri ilə dəqiqliklə göstərməlidir:

**in:** Qayda daxil olan paketə əsaslanaraq mənimsədirilir.

**out:** Qayda çıxış edən paketə əsaslanaraq mənimsədirilir.

**all:** Qayda hər iki istiqamətə mənimsədirilir.

Əgər serverin bir neçə şəbəkə kartı mövcuddursa, şəbəkə kartı istiqamətin ardi ilə təyin edilə bilər.

## OPTIONS

Opsiyalar istəkdən asılıdır. Nə qədər çox opsiya təyin edilərsə, onlar aşağıdakı ardıcılılıqda təyin edilməlidir:

**log:** Təyin edilmiş ACTION yerinə yetirildikdə, paketlərin başlığının məzmunu ipl virtual paket log alətinə yazılaçaq. Log açar söz istifadə edildikdə, aşağıdakı ayırıcı səliqə ilə istifadə edilə bilər:

**first:** Əgər **log** açar sözü **keep state** opsiyası ilə birlikdə istifadə edilərsə, yalnız bu statful paketin loglanması anlamına gəlir, hamisinin deyil.

**quick:** Əgər paket bu qaydaya uyğun gələrsə, qaydaya uyğun olan iş yerinə yetirilir və bu paketə aid olan digər heç bir qaydaya fikir verilmir.

**on:** Şəbəkə kartının adının ardınca gəlməlidir, hansı ki, **ifconfig** əmrinin çıxışında görmək olar. Bu qayda yalnız o zaman uyğun olacaq ki, paket təyin edilmiş şəbəkə kartı və təyin edilmiş istiqamətdə gedir.

### **PROTO\_TYPE**

Protokol tipi istəyə bağlıdır. Buna baxmayaraq, o, mütləqdir, ona görə ki, qayda **SRC\_PORT**, ya da **DST\_PORT** protokol tipini təyin etmek üçün istifadə edilir. Protokol tipdən istifadə elədikdə, **proto** açar söz və ardınca ya protokolun rəqəmi, ya da **/etc/protocol** faylında olan adı istifadə edin. Misal üçün, nüsxə protokol adları UDP, TCP, ya da ICMP-dir. Əgər **PROTO\_TYPE** təyin edilibsə, ancaq **SRC\_PORT**, ya da **DST\_PORT**, təyin edilməyibsə, bu protokol üçün bütün port rəqəmləri uyğun gələcək.

### **SRC\_ADDR**

**from** açar sözü mütləqdir və mənbə, ya da paket təyin edən açar sözün ardınca gelir. Mənbə olaraq İP ünvanı, HOST adı, CIDR/MASK, ünvan aralığı, ya da **all** açar sözü ola bilər. İP və MASK aralığının hansısa bir digər təyinat üsulu yoxdur və siz subnet-i tam dəqiqlik hesablayıb təyin etməlisiniz.

### **SRC\_PORT**

Mənbənin port rəqəmi istəkdən asılıdır. Əgər o, istifadə edilsə, tələb edir ki, qaydada ilk olaraq **PROTO\_TYPE** təyin edilsin. Port rəqəmi də proto açar sözün yanında olmalıdır.

Fərqli şərt operatorları var və aşağıdakılardır:

= (bərabərdir),  
!= (bərabər deyil),  
< (kiçikdir), > (böyükdür),  
<= (kiçik ya da bərabərdir) və  
>= (böyükdür ya da bərabərdir)

Port aralığını təyin etmək üçün iki port rəqəmini <>(kiçikdir və böyükdür),><(böyükdür və kiçikdir), ya da: (böyük ya bərabər və kiçik ya bərabər) simvolları daxilində yazın.

### **DST\_ADDR**

**to** açar sözü mütləqdir və mənsəb paketi təyin edəcək açar sözün ardınca gelir. SRC\_ADDR-a oxşardır İP ünvanı, HOST adı, CIDR/MASK, ünvan aralığı, ya da **all** açar sözü ola bilər.

### **DST\_PORT**

SRC\_PORT-a oxşayır, mənsəbin port rəqəmi istəkdən asılıdır. Ancaq bu istifadə edilərsə, PROTO\_TYPE-in qaydada ilk olaraq təyin edilməsini istəyir. Port rəqəmi də **proto** açar sözlə yazılmalıdır.

### **TCP\_FLAG|ICMP\_TYPE**

Əgər PROTO\_TYPE olaraq TCP təyin edilibsə, flag-lar söz hərf kimi yazılı bilər. Hər bir hərf

mümkün ola biləcək TCP flag-dır, hansı ki, qoşulmanın statusunu təyin etmək üçündür. Mümkün mənalar: **S**(SYN), **A**(ACK), **P**(PSH), **F**(FIN), **U**(URG), **R**(RST), **C**(CWN) və **E**(ECN)

Əgər **PROTO\_TYPE** olaraq **İCMP** təyin edilibsə, **İCMP** uyğunluq tipi təyin edilə bilər.

## STATE

Əgər qaydada keep state varsa, İPF veriləni bu dinamik status cədvəlinə əlavə edəcək və qoşulmaya uyğun olan ardıcıl paketlərə izin verəcək. İPF statusa görə TCP, UDP və ICMP sessiyaları izləyə bilir. Hətta başqa protokola izin verildiyi halda, İPF təyin edə biləcək istənilən paket aktiv sessiyanın hissəsidir.

İPF-də PUBLİC şəbəkə kartından çıxacaq paketlər ilk olaraq dinamik status cədvəlində yoxlanılır. Əgər paket gözlənilən aktiv seans müzakirəsi paketinə uyğun gəlirsə, o, firewall-dan çıxır və sessiya müzakirəsi axının statusu dinamik status cədvəlində yenilənir. Aktiv sessiyaya aid olmayan paketlər yenidən çıxış qaydalarında yoxlanılır. PUBLİC Internet şəbəkə kartı üzərindən gələn paketlər yenidən dinamik qaydalar cədvəlində yoxlanılır. Əgər paket gözlənilən aktiv seans paketinə uyğun gəlirsə, o, firewall-dan çıxır və sessiya müzakirəsi axının statusu dinamik status cədvəlində yenilənir. Aktiv sessiyaya aid olmayan paketlər isə, yenidən giriş qaydalarında yoxlanılır.

## Qaydaların yazılması

FreeBSD öz daxilində LoopBACK kartından istifadə edir, hansı ki, **127.0.0.1** İP ünvanına malikdir. Məhz daxildə işləməsi üçün siz firewall-da qayda yazmalısınız ki, bu şəbəkə kartının yetkisi olsun.

```
pass in quick on lo0 all  
pass out quick on lo0 all
```

PUBLİC şəbəkə kartının giriş və çıkış seksiyalarında öncə əksər istifadə olunan, sonra isə qismən daha az istifadə olunan qaydalar yazılmalıdır.

Aşağıdakı qaydalarımızda PUBLİC şəbəkə kartı olaraq **em0** nəzərdə tutulur. Bu qaydalar daxili internet yetkisi olacaq sistemlər üçün xidmətləri təyin edib statusları saxlayır. Bütün qaydalar quick istifadə edir və mənsəb ünvanları və lazımı portları təşkil edir.

```
# Internet-ə baxan şəbəkə kartı (çıkış).  
# Internetə çıxməq istəyən və firewall arxasında olan sessiyalarla üst-üstə düşür.  
# Dünya DNS serverlərinə çıkışa izin veririk.
```

```

# x.x.x.x ünvanını /etc/resolv.conf faylında olan IP ünvanla dəyişdirin.
# Bütün DNS serverlər üçün təkrar edin.
pass out quick on em0 proto tcp from any to x.x.x.x port = 53 flags S keep state
pass out quick on em0 proto udp from any to x.x.x.x port = 53 keep state

# HTTP və HTTPS-ə izin veririk.
pass out quick on em0 proto tcp from any to any port = 80 flags S keep state
pass out quick on em0 proto tcp from any to any port = 443 flags S keep state

# Email-ə izin veririk.
pass out quick on em0 proto tcp from any to any port = 110 flags S keep state
pass out quick on em0 proto tcp from any to any port = 25 flags S keep state

# NTP-yə izin veririk.
pass out quick on em0 proto tcp from any to any port = 37 flags S keep state

# FTP-yə izin veririk.
pass out quick on em0 proto tcp from any to any port = 21 flags S keep state

# SSH-a izin veririk.
pass out quick on em0 proto tcp from any to any port = 22 flags S keep state

# Ping-ə izin veririk.
pass out quick on em0 proto icmp from any to any icmp-type 8 keep state

# Yerdə qalan hər şeyi bloklayırıq və loglayırıq.
block out log first quick on em0 all

```

Bu qaydalar nüsxəsi isə PUBLIC şəbəkə kartının giriş seksiyasındadır, hansı ki, bütün istənilməyən paketləri blok edir.

Bu son qayda ilə jurnallanan paketlərin sayını azaldır.

```

# Internet şəbəkə kartı(giriş).
# Bütün giriş trafikini rezerv olunmuş ünvanlardan bloklayırıq.
block in quick on em0 from 192.168.0.0/16 to any #RFC 1918 private IP
block in quick on em0 from 172.16.0.0/12 to any #RFC 1918 private IP
block in quick on em0 from 10.0.0.0/8 to any #RFC 1918 private IP

```

```
block in quick on em0 from 127.0.0.0/8 to any          #loopback
block in quick on em0 from 0.0.0.0/8 to any          #loopback
block in quick on em0 from 169.254.0.0/16 to any      # APIPA
block in quick on em0 from 192.0.2.0/24 to any        #reserved for docs
block in quick on em0 from 204.152.64.0/23 to any     #Sun cluster interconnect
block in quick on em0 from 224.0.0.0/3 to any         #Class D & E multicast
```

# Fragmentlər və çox qısa paketləri bağlayırıq.

```
block in quick on em0 all with frags
block in quick on em0 proto tcp all with short
```

# Mənbədən yönləndirilmiş paketləri blok edirik.

```
block in quick on em0 all with opt lsrr
block in quick on em0 all with opt ssrr
```

# OS fingerprint tapmaq cəhdlərini blok edirik və ilk cəhdi jurnallayıraq.

```
block in log first quick on em0 proto tcp from any to any flags FUP
```

# Spesifik opsiyalarla olan hər şeyi bağlayırıq.

```
block in quick on em0 all with ipopts
```

# Dünya pingləri və identi bağlayırıq.

```
block in quick on em0 proto icmp all icmp-type 8
block in quick on em0 proto tcp from any to any port = 113
```

# Girişə NetBIOS xidmətlərini bağlayırıq.

```
block in log first quick on em0 proto tcp/udp from any to any port = 137
block in log first quick on em0 proto tcp/udp from any to any port = 138
block in log first quick on em0 proto tcp/udp from any to any port = 139
block in log first quick on em0 proto tcp/udp from any to any port = 81
```

Log opsiyası ilə təyin edilmiş qaydalarda sayı görmək üçün **ipfstat -hio** əmrini daxil edin ki, dəqiq sayı görə biləsiniz. Say böyük olarsa, demək ki, serverimiz hücum altındadır. Giriş seksiyasında olan digər qaydalar isə internet tərəfdən hansı qoşulmalara izin verilməsini təyin edir. Son qayda isə öncəki qaydalarda izin verilənlərdən başqa hamisini bağlayır.

# Təyin edilmiş daxili web server üçün dünya qoşulmalarına izin veririk.

```
pass in quick on em0 proto tcp from any to x.x.x.x port = 80 flags S keep state  
  
# Yerdə qalan bütün trafiki bloklayın və ilk cəhdi jurnallayın.  
block in log first quick on em0 all
```

#### NAT-in quraşdırılması

NAT-i işə salmaq üçün aşağıdakı sətirləri **/etc/rc.conf** faylına əlavə etmək və nat qaydaları olacaq faylı təyin etmək lazımdır.

```
gateway_enable="YES"  
ipnat_enable="YES"  
ipnat_rules="/etc/ipnat.rules"
```

NAT qaydaları çox imkanlıdır və çoxlu biznes xarakterli və ev tələbi olan işləri görə bilər. Burada göstəriləcək sintaksis çox asanlaşdırılmışdır ki, ümumi istifadə göstərilsin. Tam sintaksisə tanış olmaq üçün man 5 ipnat oxuyun.

NAT üçün sadə sintaksis aşağıdakı kimidir. map sintaksisinin başlanğıcı, IF isə PUBLİC şəbəkə kartının adı ilə dəyişdirilməlidir:

**map IF LAN\_IP\_RANGE -> PUBLIC\_ADDRESS**

#### **LAN\_IP\_RANGE**

- Daxili istifadəçilər tərəfindən istifadə edilən IP ünvanları aralığıdır. Bu daxili IP aralığı siyahısına aid olan IP aralıqlarından biri olacaq.

#### **PUBLIC\_ADDRESS**

- Statik dünya IP ünvanı, ya da **0/32** açar sözü ola bilər, hansı ki, IF şəbəkə kartımıza təyin edilmiş IP ünvanı mənasını verir.

İPF-də LAN-dan gələn və dünyaya çıxməq istəyən paket önce çıxış qaydalarının üstünə düşür. Sonra paket NAT qaydaları üzərinə düşür və ilk uyğun olanı qalib gəlir. İPF hər bir NAT qaydasını şəbəkə kartının adına və mənbə paketin IP ünvanına əlaqələndirərək yoxlayır. Paketin şəbəkə kartı adı NAT qaydasına uyğun olarsa, LAN-da olan paketin mənbə IP ünvanının, doğrudan da, **LAN\_IP\_RANGE** aralığında olması yoxlanılır. Uyğun olduğu halda isə, paketin mənbə IP ünvanı PUBLIC\_ADDRESS-də təyin edilən IP ünvanla dəyişdirilir. İPF bundan sonra NAT cədvəline qeyd əlavə edir ki, bu paket Internet-dən qayıdanda onu yenidən daxildə olan IP ünvanına digər yoxlanma qaydaları başlanmadan önce ünvanlana bilsin. Çoxlu sistemləri və fərqli daxili şəbəkələri olan serverlərdə bu, resurs çatışmamazlığına gətirib çıxara bilər. Bu halların qarşısını almağın iki üsulu var.

İlk üsul mənbə portları üçün aralıq portların təyin edilməsidir:

```
map em0 192.168.1.0/24 -> 0/32 portmap tcp/udp 20000:60000
```

İkinci üsul auto açar sözdür, hansı ki, NAT-dan istifadə etmək üçün boş olan portların təyin edilməsini soruşur:

```
map em0 192.168.1.0/24 -> 0/32 portmap tcp/udp auto
```

İkinci metod PUBLİC ünvanların aralığı üçün yaxşıdır. Bu, çoxlu LAN olan yerdə və tək bir PUBLİC İP ilə onların hamisini dünyaya çıxarmaq istəyi olduqda işə yarayır. Ancaq PUBLİC İP ünvanlar siyahısı işə hovuz kimi istifadə ediləcək və NAT özü qərar verəcək ki, hansı dünyaya çıxsın.

Dünya İP ünvanları aralığı CİDR istifadə edərək təyin edilə bilər. Aşağıdakı iki qayda kimi:

```
map em0 192.168.1.0/24 -> 204.134.75.0/255.255.255.0
```

```
map em0 192.168.1.0/24 -> 204.134.75.0/24
```

Əksərən istifadə olunan topologiya daxili şəbəkə seqmentində olan MAIL, ya da WEB serverin dünyaya PUBLİC İP ünvan ilə yayılmışdır. Ancaq bu serverlərdən çıxan trafik yenə də NAT üzərindən çıxmalıdır. Misal üçün, daxildə işləyən 10.0.10.25 İP ünvanlı WEB server dünyaya çıxdıqda, 20.20.20.5 İP ünvanı istifadə etməlidir. Aşağıdakı qaydada bu edilir:

```
rdr em0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80
```

Əgər daxildə təkcə bircə ədəd WEB server olarsa, aşağıdakı qaydada işləyəcək. Bu qayda dünyadan gələn bütün HTTP müraciətləri 10.0.10.25 İP ünvanı üstünə yönləndirəcək:

```
rdr em0 0.0.0.0/0 port 80 -> 10.0.10.25 port 80
```

İPF-in daxilində FTP proxy imkanı var, hansı ki, NAT-la istifadə edilə bilər. O, bütün çıkış trafikini aktiv, ya da passiv FTP qoşulmaları müraciətləri üçün monitoring edir və dinamik olaraq müvəqqəti filtrlər qaydaları yaradır ki, tərkibində FTP data kanalı tərəfindən istifadə ediləcək port rəqəmi olur. Bu, FTP qoşulmaları üçün böyük aralıq portların açılmasının qarşısını alır.

Bu misalda ilk qayda daxili LAN-dan gələn çıkış FTP trafik üçün proxy-ni çağırır. İkinci qayda isə firewall-dan gələn FTP trafiki internetə buraxır və 3-cü qayda FTP-yə aid olmayan daxili LAN-dan gələn trafiki emal edir.

```
map em0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
```

```
map em0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
```

```
map em0 10.0.10.0/29 -> 0/32
```

FTP map qaydaları NAT qaydalardan önce gedir və paket FTP qaydaya uyğun olarsa, FTP proxy müvəqqəti filtr qaydası yaradır ki, FTP sessiya paketlərini buraxmaq və NAT-ı keçə bilmək imkanı olsun. LAN-dan gələn və FTP-yə aid olmayan digər paketlər isə FTP qaydaları ilə üst-üstə düşməyəcək, ancaq 3-cü qaydaya uyğun olarsa, NAT-ı keçəcək.

“FTP proxy”-siz aşağıdakı qaydalar tələb olunacaq. Nəzərə alın ki, proxy olmadan 1024-dən yuxarı olan bütün portları açmalısınız:

```
# LAN-da PC client-in FTP-sinin Internet-ə çıxışına izin veririk.
```

```
# Aktiv və Passiv rejimlər
```

```
pass out quick on em0 proto tcp from any to any port = 21 flags S keep state
```

```
# Passiv rejimin yuxarı port rəqəmləri üçün data kanalına izin veririk.
```

```
pass out quick on em0 proto tcp from any to any port > 1024 flags S keep state
```

```
# FTP serverdən data kanalın girişinə aktiv rejim üçün izin veririk.
```

```
pass in quick on em0 proto tcp from any to any port = 20 flags S keep state
```

Artıq bütün qaydalarımız hazırdır və NAT-ı işə sala bilərik.

```
ipnat -CF -f /etc/ipnat.rules
```

- **-CF** mövcud NAT qaydalarını silir və dinamik translyasiya cədvəlinin məzmununu sıfırlayır.
- **-f** NAT qaydaları yüklənəcək fayl ünvanını təyin edirik.

```
ipnat -s
```

- NAT statistikalarına baxırıq.

```
ipnat -l
```

- NAT cədvəlinin hazırkı xəritələnməsini çap edirik.

```
ipnat -v
```

- Bol məlumat rejimi (problem araşdırılonda lazım olur).

```
ipfstat
```

- Firewall-dan keçən qaydalara əsaslanan statistikanı çap edir.

```
ipf -Z
```

- Statistikani sıfırlayır.

```
ipfstat -ion
```

- **-i** giriş qaydalarına görə statistikalar.

- **o** çıxış qaydalarına görə statistikalar.

- **n** qaydaların rəqəmlərinə baxırıq.

- **h** qaydanın neçə dəfə üst-üstə düşməsini çap edir.

- **t** top əmri çıxışı formatına uyğun olaraq çap edir.

Hücum paketlərinə baxmaq üçün effektiv əmrdir.

## İPF Jurnallama

İPF jurnallama üçün ipmon imkanını verir, bunun sayəsində firewall jurnalama informasiyasını insan oxuya biləcək formatda yazır. Ancaq kernel-də **IPFILTER\_LOG** tələb edir. İstənilən yazılmış qaydada **log** açar sözü istifadə edilən kimi, həmin qaydaya uyğun olan paketlər haqda məlumatlar təyinata uyğun olaraq jurnallanmasıdır. Susmaya görə **ipmon -Ds** rejimində **local0** jurnallanması səviyyəsi istifadə edilir. Aşağıdakı jurnallanması səviyyələri jurnallanması bölgüləri aparmaq üçün istifadə edilə bilər:

- |                    |   |
|--------------------|---|
| <b>LOG_INFO</b>    | - Paketlər "log" açar söz istifadə edir, görüləcək işdə pass, ya da block olarsa. |
| <b>LOG_NOTICE</b>  | - Buraxılan paketlər jurnallanmasıdır.  |
| <b>LOG_WARNING</b> | - Blokunan paketlər jurnallanmasıdır.   |
| <b>LOG_ERR</b>     | - Tam bitməyən və ya yarımcıq başlıqlı olan paketlər jurnallanmasıdır.            |

Səliqə ilə İPF-in bütün dətaralarının jurnallanması üçün **/var/log/ipfilter.log** adlı fayl yaradırıq:  
**touch /var/log/ipfilter.log**

Sonra bütün jurnallanmış mesajların seçdiyimiz fayla yazılması üçün aşağıdakı sətri "**/etc/syslog.conf**" faylinə yazırıq:

**local0.\* /var/log/ipfilter.log**

Dəyişikliyimizin işə düşməsi üçün aşağıdakı əmri daxil edirik:

**service syslogd reload**

Jurnalların rotasiyası üçün lazımi quraşdırmasını **/etc/newsyslog.conf** faylinə əlavə etməyi unutmayın.



# BÖLÜM 10

## FAMP, DNS BIND

- / FreeBSD 10.1 x64 AMP (Apache MySQL PHP)
- / Berkeley Internet Domain (BIND) - DNS xidmətləri

Başlığımızda tam funksionallığı ilə işləyən WEB serveri qururuq. Web serverimizdə verilənlər bazası olaraq MySQL və əksər saytların işlədiyi PHP işlək vəziyyətə gətirilir. Həmçinin Web server təkcə IP ünvana görə deyil, həm də adla işləyə bilmək üçün quraşdırılır. DNS serverin quraşdırılması və başlangıç teoriyaları bu başlıqda açıqlanır. DNS server master-slave rejimdə işləyir. Məqsəd ondan ibarətdir ki, əsas DNS server öz funksionallığını itirdikdə, ikinci(slave) olan server əsas (master)-ə gələn müraciətləri öz üzərinə götürsün. Sonda sistemdə adların çevrilməsi ardıcılılığı alqoritmi izah olunur.

# **FreeBSD 10.1 x64 AMP (Apache MySQL PHP)**

PHP-də, ya da PERL-də yazılan WEB kodlarının işləməsi üçün müəyyən bir mühit lazımdır. Bu mühitə WEB server deyilir. Dünya statistikasında PHP-də yazılan saytlar üçün ən çox istifadə edilən **apache** və **nGinx**-dir. Ancaq bu başlığımızda apache-ni müzakirə edəcəyik. Başlığımızda **Apache24, Mysql56-Server** və **PHP56** yüklenib hazırlanacaq. Sonra isə apache WEB serverimizin funksionallığı açıqlanacaq. Lazım olacaq paketləri yükleyək və mühiti hazırlayaq:

## **Apache24 yüklenməsi və quraşdırılması**

```
cd /usr/ports/www/apache24  
make config
```

- Port ünvanına daxil oluruq.
- Lazımi modulları seçirik.

```
[x] EXT_FILTER External filter module
[x] FILE_CACHE File cache
[x] FILTER Smart Filtering
[x] HEADERS HTTP header control
[x] HEARTBEAT Generates Heartbeats
[x] HEARTMONITOR Collects Heartbeats
[x] IDENT RFC 1413 ident lookups
[x] IMAGEMAP Server-side imagemaps
[x] INCLUDE Server-side includes
[x] INFO Server information
[x] IPV4_MAPPED Allow IPv6 sockets to handle IPv4 connections
[x] LBETHOD_BYBUSYNESS Apache proxy Load balancing by busyness
[x] LBETHOD_BYREQUEST Apache proxy Load balancing by request counting
[x] LBETHOD_BYTRAFFIC Apache proxy Load balancing by traffic counting
[x] LBETHOD_HEARTBEAT Apache proxy Load balancing from Heartbeats
[x] LDAP LDAP caching and connection pooling services
[x] LOGIO Input and output logging
[x] LOG_DEBUG Configurable debug logging
[x] LOG_FORENSIC Forensic logging
[x] LUA Apache Lua Framework
[x] LUAJIT LuaJit Support
[x] MACRO Define and use macros in configuration files
[x] MIME Mapp file-ext. to MIME (recommended)
[x] MIME_MAGIC Automatically determining MIME type
[x] NEGOTIATION Content negotiation
[x] PROXY Build enabled PROXY modules
[x] RATELIMIT Output Bandwidth Limiting
[x] REFLECTOR Reflect request through the output filter stack
[x] REMOTEIP Translate header contents to an apparent client re
[x] REQTIMEOUT Limit time waiting for request from client
[x] REQUEST Request Body Filtering
[x] REWRITE Rule based URL manipulation
[x] SED Filter request and/or response bodies through sed
[x] SESSION Build enabled SESSION modules
[x] SETENVIF Modify ENV vars based on characteristics of the re
[x] SLOTMEM_PLAIN Slotmem provider that uses plain memory
[x] SLOTMEM_SHM Slotmem provider that uses shared memory
[x] SOCACHE_DBM dbm small object cache provider
[x] SOCACHE_DC distcache small object cache provider
[x] SOCACHE_MEMCACHE memcache small object cache provider
[x] SOCACHE_SHMCB shmcdb small object cache provider
[x] SPELING Correct common URL misspellings
[x] SSL SSL/TLS support (mod_ssl)
[x] STATUS Process/thread monitoring
[x] SUBSTITUTE Response content rewrite-like filtering
[x] SUEXEC Set uid and gid for spawned processes
[x] UNIQUE_ID Per-request unique ids
[x] USERDIR Mapping of requests to user-specific directories
[x] USERTRACK User-session tracking
[x] VERSION Determining httpd version in config files
[x] VHOST_ALIAS Mass virtual hosting
[x] WATCHDOG Watchdog module
[x] XMLZENC 118n support for markup filters
qqqqqqqqqqqqqqqqqqq Example and devel modules (do not use in prod) qqqqqqqqqqqqq
[x] BUCKETTEER (dev) buckets manipulation filter
[x] CASE_FILTER (dev) example uppercase conversion filter
[x] CASE_FILTER_IN (dev) example uppercase conversion input filter
[x] ECHO (dev) example echo server
[x] EXAMPLE_HOOKS (dev) example hook callback handler module
[x] EXAMPLE_IPC (dev) example of shared memory and mutex usage
[x] OPTIONAL_FN_EXPORT (dev) example optional function exporter
[x] OPTIONAL_FN_IMPORT (dev) example optional function importer
[x] OPTIONAL_HOOK_EXPO (dev) example optional hook exporter
[x] OPTIONAL_HOOK_IMP (dev) example optional hook importer
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Build enabled PROXY modules qqqqqqqqqqqqqqqqqqqqqqqqqqqqq
[x] PROXY_AJP AJP support module for mod_proxy
[x] PROXY_BALANCER mod_proxy extension for load balancing
[x] PROXY_CONNECT mod_proxy extension for CONNECT request handling
[x] PROXY_EXPRESS Dynamic mass reverse proxy extension for mod_proxy
[x] PROXY_FCGI FastCGI support module for mod_proxy
[x] PROXY_FDPASS fdpass external process support module for mod_pro
[x] PROXY_FTP FTP support module for mod_proxy
[x] PROXY_HTTP HTTP support module for mod_proxy
[x] PROXY_SCGI SCGI gateway module for mod_proxy
[x] PROXY_WSTUNNEL Websockets Tunnel module for mod_proxy
[x] PROXY_HTML Fix HTML Links in a Reverse Proxy
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Build enabled SESSION modules qqqqqqqqqqqqqqqqqqqqqqqqqqqqq
[x] SESSION_COOKIE Session cookie module
[x] SESSION_CRYPTO Session crypto module
[x] SESSION_DBD Session dbd module
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqq The default MPM module qqqqqqqqqqqqqqqqqqqqqqqqqqqqq
(*) MPM_PREFORK non-threaded, pre-forking web server
( ) MPM_WORKER hybrid multi-threaded multi-process web server
( ) MPM_EVENT MPM worker variants with the goal of consuming thre
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Build all MPMs as shared Module qqqqqqqqqqqqqqqqqqqqqqqqqqqqq
(*) MP_SHARED All MPMs as loadable module
```

**make install** - Yükleyirik.

**echo 'apache24\_enable="YES"' >> /etc/rc.conf** - Apache24-ü StartUP faylımiza əlavə edirik ki, sistem yenidənyüklənməsindən sonra avtomatik işə düşsün.

Daemonu işə salmaq üçün işə **/etc/hosts** faylına öz serverimizin host adını əlavə edirik(X-ların yerinə serverin IP ünvanı yazılımalıdır):

**echo "X.X.X.X ps.sarg.az ps" >> /etc/hosts**

Öncədən **/usr/local/etc/apache24/httpd.conf** faylında **DirectoryIndex** sətrini tapırıq və sonuna **index.php** əlavə edirik. index.php-nin mənası odur ki, WEB server üçün susmaya görə olan index fayllardan ikincisi budur. Yəni saytımıza ad və ya IP ünvanla müraciət edilərsə, öncə index.html və sonra **index.php** faylına müraciət olunacaq.

**DirectoryIndex index.html index.php**

PHP-ni yükleməzdən öncə işləməsi üçün tələb edilən sətirləri WEB server quraşdırımızı əlavə edirik. Bunun üçün

**/usr/local/etc/apache24/Includes/php-application.conf** adlı fayl yaradırıq və içində aşağıdakı sətirləri əlavə edirik.

**AddType application/x-httpd-php .php**

**AddType application/x-httpd-php-source .phpsXsource**

**/usr/local/etc/rc.d/apache24 start** - Apache24 serverimizi işə salırıq.

**/usr/local/etc/apache24** - Əsas quraşdırma faylı yerləşən qovluqdur.  
**/var/log/httpd-error.log** - Susmaya görə səhvər yığılan jurnal faylıdır.  
**/var/log/httpd-access.log** - Susmaya görə müraciətlər yığılan jurnal faylıdır.  
**/usr/local/www/apache24/data** - Susmaya görə olan PUBLIC\_HTML qovluğudur.  
**/usr/local/www/cgi-bin** - Susmaya görə perl-də yazılın proqramlar bu ünvanda yerləşir.  
**apachectl -S** - Quraşdirmaları və vHost quraşdirmaları çap edir.  
**httpd -t** - Quraşdırma fayllarında sintaksisin düzgünlüyünü yoxlayırıq.

**/usr/local/etc/apache24/httpd.conf** faylında olan bəzi direktivləri açıqlayaq:

İstənilən vHost üçün PUBLIC\_HTML qovluğunda AuthConfig bölməsində "**AllowOverride All**" olduqda htpasswd istifadə etmək imkanı yaranır.

**Listen** - Qulaq asdıgı port.

### **ScriptAlias**

- Alias edir ve adı esl ünvana yöneldir.  
Məsələn: **ScriptAlias /post /usr/local/www/postfixadmin**  
Web browserdə **http://server\_ip/post** yazsaq, kifayətdir.

**/usr/local/etc/apache24/httpd.conf** faylında aşağıdakı sətirlərin qarşısından şərhi silirik:

**LoadModule rewrite\_module libexec/apache24/mod\_rewrite.so**

**LoadModule ssl\_module libexec/apache24/mod\_ssl.so**

Yeni vHostlar üçün quraşdırma sətrimizi httpd.conf faylına əlavə edirik:

```
echo "Include/usr/local/domen/*" >> /usr/local/etc/apache24/httpd.conf
```

Təhlükəsizlik üçün aşağıdakı sətirləri **/usr/local/etc/apache24/httpd.conf** faylına əlavə edirik:

# Müraciət həcmi 1MB edirik.

**LimitRequestBody 1048576**

#xml tərkibini 1 milyon bayt edirik (1mb).

**LimitXMLRequestBody 10485760**

# Müraciətin bitmə vaxtı

**Timeout 45**

**MaxKeepAliveRequests 100**

**KeepAliveTimeout 10**

# Susmaya görə olan unikod UTF-8

**AddDefaultCharset UTF-8**

# Apache-mizin versiyasını gizlədirik.

**ServerSignature Off**

**ServerTokens Prod**

# Apache-nin Httpd Trace və Track metodunu bloklayırıq.

**RewriteEngine On**

**RewriteCond %{REQUEST\_METHOD} ^TRACE**

**RewriteRule .\* - [F]**

**RewriteCond %{REQUEST\_METHOD} ^TRACK**

**RewriteRule .\* - [F]**

## Virtual Hosts

İmkanları:

1. IP ilə qoşulan - Bir site-a 1 IP ünvan
2. Adla yazılıan - Çoxlu site-a 1 IP ünvan

Şəbəkə kartınıza bir neçə ikinci dərəcəli IP ünvan təyin edin ki, virtualhost-u test edə biləsiniz.

İşimiz:

1. IP bazalı seçilmiş IP ünvanla virtual host yaradaq.

```
<VirtualHost 192.168.121.136>
    ServerAdmin webmaster@cavid.az
    ServerName cavid.az
    DocumentRoot /usr/local/www/cavid
    <Directory /usr/local/www/cavid>
        Options none
        AllowOverride Limit
        Order Deny,Allow
        Require all granted
    </Directory>
</VirtualHost>
```

**Qeyd:** Options bölümünə xüsusi diqqət yetirin, çünki burada olan bütün opsiyaların qalmasına ehtiyacınız yoxdur. Həm də bütün opsiyaların olması da həddən artıq təhlükəlidir. Bu səbəbdən yalnız ən çıxılmaz hallarda və sərt tələb yarandığı zaman hansıa opsiyanın artırılması haqqında düşünün.

```
mkdir /usr/local/www/cavid - PUBLİC_HTML qovluğunuzu yaradırıq.
```

Test üçün **index.html** faylı yaradaq və İP ünvana web browser vasitəsilə müraciət edib yoxlayaqq.

```
echo "<html><h1><center>Cavid-in saytidır</center></h1></html>" >> /usr/local/www/cavid/index.html
```

Nəticə aşağıdakı kimi olacaq:

192.168.121.136



**Cavid-in saytidır**

Növbəti misalımızda IP bazalı virtualhost-da IP:port birləşməsi təyin etmişik və jurnal fayllarımızı fərqli fayllara yazmışıq.

```
<VirtualHost 192.168.121.137:80>
```

```
    ServerAdmin webmaster@elcin.az
```

```
    ServerName elcin.az
```

```
    CustomLog "/var/log/elcin_access.log" common - Girişlərin jurnallaşmasını  
    ayırmış faylda edirik.
```

```
    ErrorLog /var/log/elcin_error.log - Səhvlerin jurnallaşmasını  
    ayırmış faylda edirik.
```

```
    DocumentRoot /usr/local/www/elcin
```

```
    <Directory /usr/local/www/elcin>
```

```
        Options none
```

```
        AllowOverride Limit
```

```
        Order Deny,Allow
```

```
        Require all granted
```

```
    </Directory>
```

```
</VirtualHost>
```

```
mkdir /usr/local/www/elcin
```

- **elcin.az** saytı üçün PUBLIC\_HTML qovluğununu yaradırıq.

```
touch /var/log/elcin_access.log
```

- Sayta daxil olma jurnallarını ayrıca faylda qeyd edirik.

```
touch /var/log/elcin_error.log
```

- Saytin səhvlerinin jurnallarını ayrıca faylda qeyd edirik.

Test üçün **index.html** faylı yaradaq və IP ünvana web browser vasitəsilə müraciət edib yoxlayaqq.

```
echo "<html><h1><center>Elcin-in saytidir</center></h1></html>" >> /usr/local/www/  
elcin/index.html
```

2. Ad bazalı Virtual Hostlar:

**Qeyd:** Ad bazalı Virtual Host qurulanda **ServerName** direktivi kritikdir.

**Qeyd:** Ad bazlı Virtual host yaradılanda **httpd.conf** faylında "NameVirtualHost \*"  
yazılırsa, Virtual hostda **IP:port** yazmaq olmaz, mütləq "**VirtualHost \***" olmalıdır.

Məsələn: **behruz.az** saytını quraşdırırıq.

**Qeyd:** Ancaq sərt olaraq **IP:Port** birləşməsini bağlamağı unutmayın. Çünkü siz artıq  
**VirtualHost** prinsipi ilə bir IP ünvanla işləyirsiniz.

**/usr/local/domen/behruz.az** faylına aşağıdakı sətirləri əlavə edirik:

```
<VirtualHost *>
    ServerName behruz.az
    ServerAlias www.behruz.az
    DocumentRoot "/usr/local/www/behruz"
<Directory "/usr/local/www/behruz">
    Options All
    AllowOverride AuthConfig
    Order Deny,Allow
    Require all granted
</Directory>
</VirtualHost>
```

- Bütün opsiyalar dəstəklənsin.  
- htpasswd istifadə etmək olacaq  
- Giriş ardıcılılığı (əvvəl icazə, sonra qadağa).  
- Girişə hamiya izin veririk.

Test üçün **index.html** fayı yaradıq və web browser vasitəsi ilə **http://behruz.az** linkinə müraciət edib yoxlayaq. Ancaq sizdə bu ad ya DNS-də tanınmalıdır, ya da **C:\Windows\System32\drivers\etc\hosts** faylinin sonuna "**192.168.121.134 behruz.az**" sətri əlavə edilməlidir.  
**echo "<html><h1><center>Behruz-un saytidir</center></h1></html>" >> /usr/local/www/behruz/index.html**

Nəticə şəkildəki kimi olacaq:



**Behruz-un saytidir**

behruz.az sayt üçün giriş istifadəçi adı və şifrə ilə təyin etmək istəyiriksə, saytin PUBLIC\_HTML qovluğunda ".htaccess" adlı fayl yaradıb məzmununa tələb edilən sintaksisi əlavə etməliyik.

```

cd /usr/local/www/behruz
ee .htaccess
AuthUserFile /usr/local/www/behruz/.htpasswd
AuthName "Soft Admin"
AuthType Basic
Require valid-user

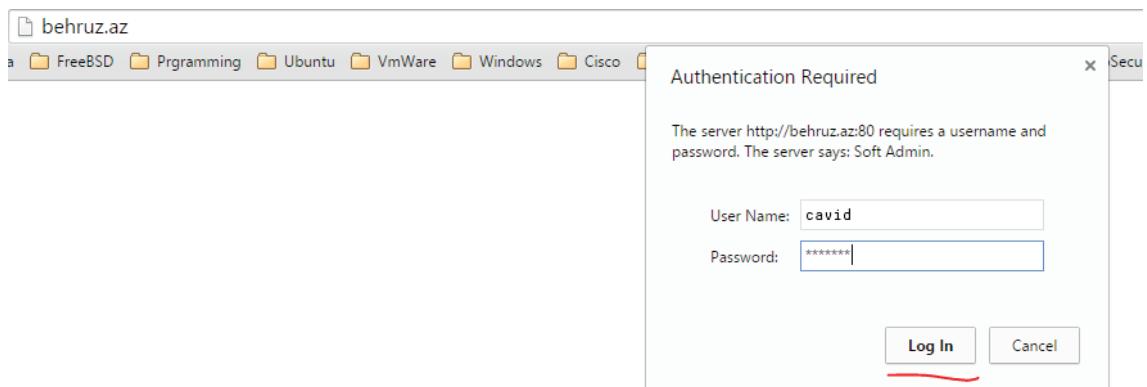
```

Istifadəçi adı ilə şifrəni yaradırıq.

```
htpasswd -bc .htpasswd cavid freebsd
```

- .htpasswd faylına **cavid** istifadəçi adını **freebsd** şifrəsi ilə yazın.
- **b** - Command line-dan istifadəçi adı və şifrəni götürün.
- **c** - Göstərilən faylı yaradın və ona daxil edin (əgər varsa, silib yenidən yazacaq).

Sonra **http://behruz.az** linkinə yenidən daxil olsanız, şəkildəki nəticəni əldə etmiş olacaqsınız:



### Apache Log Strukturu

1. Serverin Log control sətrini tapmaq üçün **/usr/local/etc/apache24/http.conf** faylında **LogLevel: Control** sətrini tapmaq lazımdır. Misal üçün, журнallanma səviyyəsini qaldırı bilərik - '**LogLevel emerg**'

**Qeyd:** HTTP səhvlerin tipləri.

- a. **20x** – problem yoxdur
- b. **30x** – yönləndirmə
- c. **40x** – istifadəçi səhvleri
- d. **50x** – server səhvleri

Apache dəyişənlər:

- '%h' - Hosta qoşulur.
- '%l' - Ident. yoxlayın, adətən çatılmazdır.
- '%u' - İstifadəçi qoşulması - adətən çatılmazdır, çünki əksər HTTP qoşulmalar anonim olur.
- '%t' - Vaxt möhürü.
- '%r' - Müraciət üsulu: 'GET', 'POST'
- '%>s' - Status mesajı: 20x - 50x mesaj
- '%b' - Müştəri və server arasında transfer olan bayt həcmi.
- '%{Referer}' - Siz necə götürdüñünüz? - Qeyd: '' bu, o deməkdir ki, istifadəçi birbaşa qoşulmuşdur.
- '%{User-Agent}' - HTTP istifadəçi qoşulur.

**Qeyd:** Mesajlar jurnalı '**favicon.ico**' yoxdursa, həmişə link tutur.

**apachectl configtest**

- Quraşdırma faylında olan sintaksisin doğruluğunu yoxlayır.

**apachectl -M**

- İstifadə etdiyi modulları çap edir.

**apachectl -l**

- Modullar, daxili kompilyasiya olmuşları çap edir.

**/usr/local/etc/apache24/** qovluğundakı faylları açıqlayaq:

**mime.types**

- Standart fayl tipləri və identifikasiya xarakteristikalarının siyahısını təşkil edir. Web server istifadəçiyə faylı ötürəndə həmin faylin identifikasiyasını elə etməlidir ki, istifadəçi fayl tipini anlaşın (Biz bu faylda heç vaxt dəyişiklik etmirik).

**magic**

- **mime.types** faylı ötürülən bütün fayl tiplərini tanımır. O, faylların tanıdılması üçün "**apache**"-nin "**mime\_magic**" modulundan istifadə edir. Həmin modul da öz növbəsində "**magic**" faylına müraciət edir.

## **httpd.conf**

- Bu fayl əsas qlobal fayldır. Bütün host quraşdırımları bu faylda yerinə yetirilir.

## **MySQL yüklenməsi və quraşdırılması**

MySQL – OpenSource verilənlər bazasıdır. PHP-də yazılmış əksər WEB serverlər bu verilənlər bazasından istifadə edir. Klaster funksionallığına sahibdir və kiçik bazalar üçün çox sürətlidir.

**cd /usr/ports/databases/mysql56-server** - Port ünvanına daxil oluruq.

**make install** - Yükləyirik.

Quraşdırma faylini yaradırıq. **/var/db/mysql/my.cnf** faylına aşağıdakı sətirləri əlavə edirik:

```
[mysqld]
bind-address = 127.0.0.1
character-set-server=utf8
init-connect="SET NAMES utf8"
query_cache_size=64M
long_query_time=5
slow_query_log=1
slow_query_log_file=/var/db/mysql/slow.log
```

**chown mysql:mysql /var/db/mysql/my.cnf** - Quraşdırma faylına MySQL sahibliyi təyin edirik.

**chmod 0660 /var/db/mysql/my.cnf** - Oxuma, yazma yetkisi yalnız mysql üçündür.

**touch /var/db/mysql/slow.log** - Jurnal faylini yaradırıq.

**chown mysql:mysql /var/db/mysql/slow.log** - Jurnal faylı üçün MySQL daemon-a izin veririk.

**chown -R mysql:mysql /var/db/mysql** - Baza qovluğu hüquqlarını MySQL-ə təyin edirik.

**echo 'mysql\_enable="YES"' >> /etc/rc.conf** - MySQL daemon-u StartUP faylımiza əlavə edirik.

**service mysql-server start** - MySQL daemonu işə salırıq.

```
/usr/local/bin/mysql_secure_installation
- Skript işə salırıq ki, MySQL daemon'a aid olan
  ilkin quraşdırılmalarımızı edək. Bura root
  istifadəçisinin şifrəsi, uzaqdan idarə etmənin
  söndürülməsi və test bazasının silinməsi daxildir.

Enter current password for root (enter for none): Şifrə yoxdur, ENTER sıxırıq
Change the root password? [Y/n] ENTER
New password: yeni_şifrə
Re-enter new password: təkrar_yeni_şifrə

Remove anonymous users? [Y/n] Y
... Success!

Disallow root login remotely? [Y/n] ENTER
... Success!

Remove test database and access to it? [Y/n] ENTER
... Success!

Reload privilege tables now? [Y/n] ENTER
... Success!

Nəticəni test etmək üçün consol-a daxil oluruq və çıxırıq:
mysql -uroot -p
        - MySQL consol-una daxil oluruq.
Enter password: şifrə
mysql> \q
        - Və consol-dan çıxırıq.

Yükləndikdən sonra, şifrəni əvvəlki qayda ilə dəyişməsək, mysqladmin əmri ilə də dəyişə bilərik.
mysqladmin -u root password freebsd
        - root istifadəçi şifrəsini freebsd təyin edirik.

mysqladmin -u root -p'kohne_parol' password yeni_parol
        - root istifadəçininin şifrəsini dəyişirik (istənilən
          istifadəçini etmək olar).

mysqldump -uroot -pfreebsd --all-databases > all.sql
        - Bütün verilənlər bazalarını all.sql adlı fayla
          rezerv nüsxə edirik.
```

```
mysqldump -uroot -pfreebsd --databases mysql > mysql.sql
```

- Yalnız **mysql** adlı verilənlər bazasının rezerv nüsxəsini **mysql.sql** adlı fayla yazırıq.

```
FLUSH PRIVILEGES;
```

- Hər bir qlobal dəyişiklikdən sonra bu əmri mütləq işə salın ki, yetkilər yenidən oxudulsun.

```
CREATE DATABASE websayt;
```

- websayt adlı verilənlər bazası yaradırıq.

```
use websayt;
```

- websayt adlı verilənlər bazasını seçirik.

websayt adlı verilənlər bazasının içində **userler** adlı cədvəl və onun içində fname, lname, email adlı sütunlar yaradırıq:

```
CREATE TABLE userler (fname char(20), lname char(20), email char(20) );
```

User-lər cədvəlinin sütunlarına ardıcıl məlumat daxil edirik:

```
INSERT INTO userler (fname, lname, email) VALUES ('Cavid', 'Bayramov', 'cavid.b@gmail.com');
```

```
show databases;
```

- Bütün verilənlər bazalarının siyahısını çap edirik.

```
show tables;
```

- Mövcud yerləşdiyimiz bazanın içində olan cədvəlləri çap edirik.

```
DROP TABLE userler;
```

- user-lər adlı cədvəli silirik.

```
select * from user;
```

- user cədvəlində olan bütün sütunları çap et.

```
describe websayt.userler;
```

- websayt bazasında user-lər cədvəlinin strukturunu (sütunlarını və tiplərini) göstər.

websayt adlı bazaya **cavid** istifadəçi adı və **192.168.1.115** IP ünvandan freebsd şifrəsi ilə qoşulmaq olar.

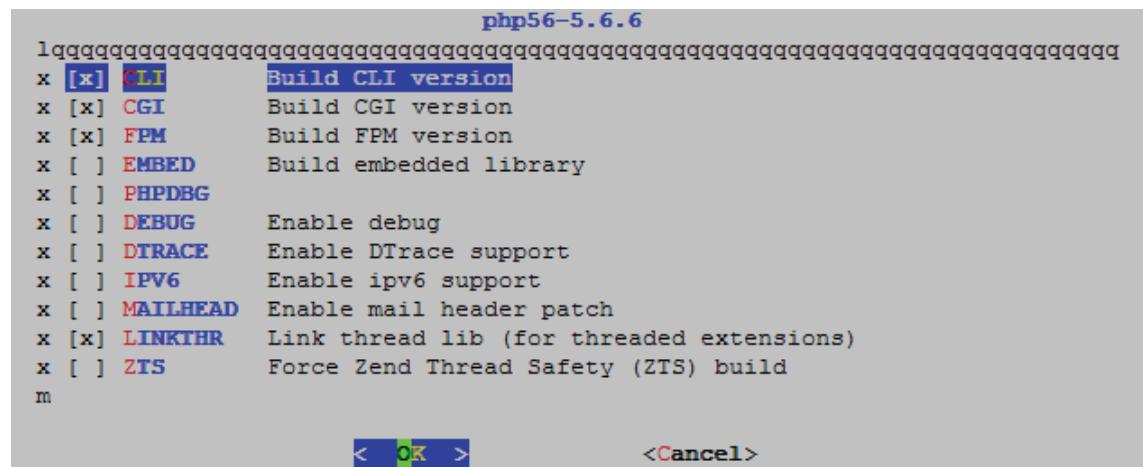
```
GRANT ALL PRIVILEGES ON websayt.* TO 'cavid'@'192.168.1.115' IDENTIFIED BY 'freebsd' WITH GRANT OPTION;
```

## PHP56 yüklenməsi və quraşdırılması

PHP-ni yükleyirik. PHP programlaşdırma dilidir və əksər WEB saytların sürətli işləməsi üçün bu dildə yazılır. Ancaq nəzərə alın ki, PHP həddən artıq hücumlara məruz qalan bir mühitdir və həqiqətən də, kifayət qədər deşiklərə sahibdir. Lakin bizim müzakirə mövzumuz təhlükəsizlik deyil. Ona görə də PHP mühiti yükleyək və işlek vəziyyətə gətirək:

```
cd /usr/ports/lang/php56  
make config
```

- PHP56-nın port ünvanına daxil oluruq.
- Lazımı modullarını seçirik.



```
make all install
```

- Yükleyirik.

PHP üçün program yazdıraqda yazılın kodun tələblərinə uyğun olaraq genişlənmələr yüklenməlidir. Misal üçün, yazılın kod MySQL bazasına qoşulmalı olarsa, lazımı genişlənmə olmadığıda qoşula bilməyəcək. Buna görə də lazımı genişlənmələri yükleyirik.

```
cd /usr/ports/lang/php56-extentions  
make config
```

- Port ünvanına daxil oluruq.
- Lazımı genişlənmələri seçirik.

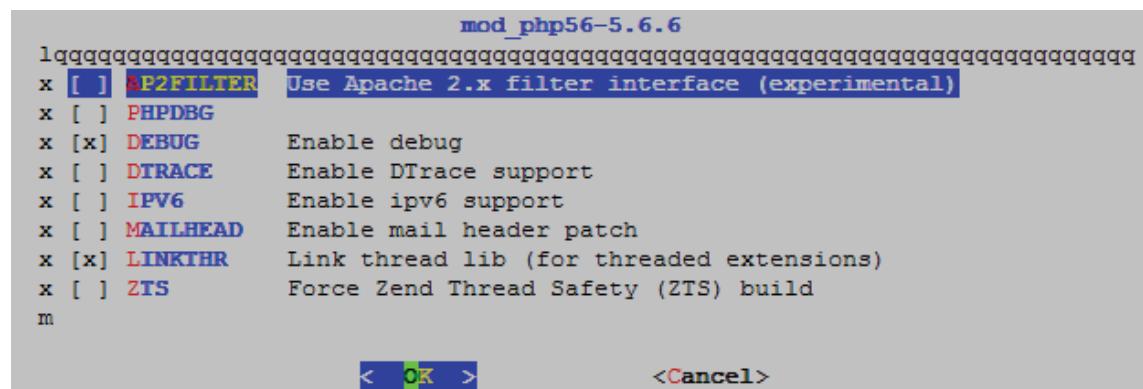


```

make install                                - Yükleyirik.
cd /usr/ports/www/mod_php56                - Apache24 için PHP modulunu portlardan
                                              yükleyirik.

make config                                  - Lazımi modulları seçirik.

```



```

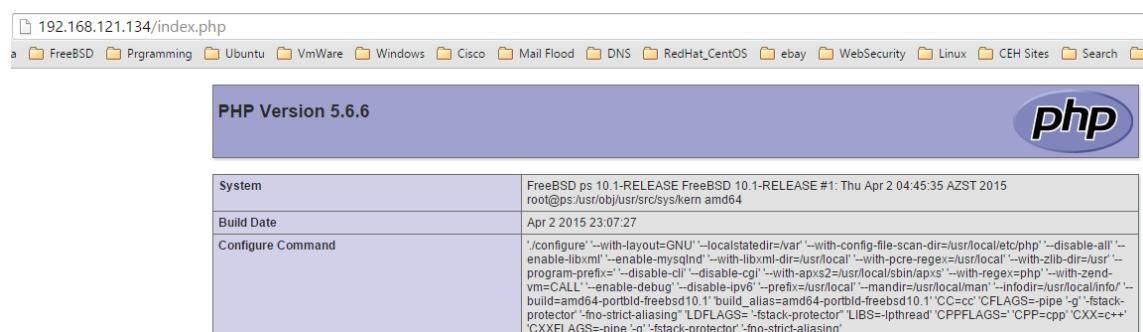
make install                                - Yükleyirik.
/usr/local/www/apache24/data/index.php      faylinə aşağıdakı sətirləri əlavə edirik ki, PHP-nin
isleməsini test edək.
<?
phpinfo();
?>

```

```

/usr/local/etc/rc.d/apache24 restart      - WEB serveri yenidən işə salırıq ki, dəyişikliklər
isə düşsün.

```



[http://server\\_IP\\_address/index.php](http://server_IP_address/index.php) linkini WEB browser-imizdə açırıq.  
Əgər aşağıdakı şəkil sizdə də eyni şəkildə açılsrsa, demək ki, hər şey qaydasındadır.

PHP üçün quraşdırma faylı hazırlayıraq və lazımı yetkini veririk:

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini  
chmod u+w /usr/local/etc/php.ini
```

- |  |  |
|--|--|
| <b>php -m</b>  | - Bütün kompilyasiya edilmiş modullarını çap edəcək. |
| <b>php -v</b>  | - Versiyasını çap edəcək.                            |
| <b>php -ini</b>                                      | - Quraşdırma fayllarının adlarını çap edəcək.        |
| <b>php -l /usr/local/www/apache24/data/index.php</b> | - Faylın sintaksis problemlərini yoxlayırıq.         |
| <b>php -i</b>  | - PHP haqda məlumatları çap edir.                    |
| <b>php -h</b>  | - Köməkçini çap edəcək.                              |

# Berkeley Internet Domain (BIND) - DNS xidmətləri

DNS işini görmək üçün FreeBSD serverin əvvəlki versiyalarında susmaya görə **named** adlı servis işləyirdi və bütün quraşdırma faylları sistemin **/etc/namedb** qovluğunda yerləşirdi. Onu işə salmaq üçün isə sadəcə **/etc/rc.conf** StartUP faylinə lazımi sətirlərin əlavə edilməsi zəruri idi. Ancaq FreeBSD10.1-də artıq bind fərqli versiyalarla portlarda yerləşir və onu önce yüklemək, sonra quraşdırmaq lazımdır.

İmkanlar:

1. Adı IP ünvana çevirmək - Forward
  - a. **www.dns.az** -> **188.72.129.10**
2. IP ünvanı ada çevirmək - Reverse
  - a. **188.72.129.10** -> **dns.az**

**/usr/local/etc/namedb/**

- Susmaya görə olan quraşdırma və zone fayllarının qovluğu.

**/usr/local/etc/namedb/named.conf**  
**/usr/local/sbin/named**

- Əsas bind9 quraşdırma faylı.  
- BIND9 server binar faylı.

**Qeyd:** Göstərilən config/zone faylı quruluşu **dns.az** DNS-i tərəfindən istifadə olunur:

1. **\*.db** - **Zone** quraşdırma fayllarıdır.
2. **named.\*** - 'named'-in quraşdırma fayllarıdır.
3. **Bind9** – Susmaya görə FreeBSD10.1-dən kiçik versiyalarda yüklənmiş olur.
4. Göstərilən **/usr/local/etc/namedb/master** ünvanı araşdırıraq.
  - a. **localhost-forward.db** - Localhost üçün **zone** faylıdır.
  - b. **localhost-reverse.db** - **Zone** faylı localhost üçün **revers** faylıdır.
  - c. **empty.db** - Reverse **zone** fayllar üçün şablondur.
5. **/usr/local/etc/namedb** qovluğunda isə root ad serverləri və qlobal quraşdırma üçün tələb edilən faylları yerləşdiririn.
  - d. **named.conf** - Əsas quraşdırma faylı.
  - e. **named.root** - Root serverlərin siyahısı.

**Qeyd:** Eyni zamanda həm "**primary server**", həm də "**secondary server**" ola bilər.

**Qeyd:** Çalışaq ki, daxili şəbəkəmizdə xarici adlardan istifadə etməyək.

**Qeyd:** "**bind9**"-un daxilində olan "**sample zones**"-ları istifadə etmək daha yaxşı olar.

Başlanğıc üçün tələb edilən yazı tiplərini və zone faylinin sintaksisini açıqlayaq:

- ;** - şərh
  - @** - Zonanın içində domain adı.
  - IN** - Internet Record.
  - A** - Lazımı adın IP ünvanını təyin edir.
  - MX** - (Mail Exchange) Mail serverlərin təyin edilməsi üçün istifadə edilir. Balansı təyin etmək üçün rəqəmlərlə prioritet yazmaq olur. Daha kiçik rəqəm daha üstün deməkdir.
- Məsələn: **IN MX 10 mx.dns.az.**

**root.dns.az.** - Bu domain üçün email ünvanı (DNS zonada email ayıricısı "@" simvolu deyil, adı '.'(nöqtə simvoludur)).

**HINFO** - DNS yerləşən serverin avadanlıq statusu haqqında məlumat. Test etmək üçün əmr "**nslookup -ty=hinfo dns.az**", ya da '**dig hinfo dns1.az**'

**TXT** - İstənilən tip mətni təyin etmək olur. Məsələn: **TXT "salam alekum"** Test etmək üçün əmr: "**nslookup -ty=txt dns.az**", ya da "**dig txt dns.az**"

- PTR** - IP ünvanını ada çevirmek üçün istifadə edilir.
- \$GENERATE** - Əgər biz müəyyən şəbəkə aralığına rəqəm ardıcılılığı ilə PTR təyin etmək istəsək, bu yazidan istifadə edirik.

**Qeyd:** BIND serverin **zone** fayllarında həmişə **SOA** (**S**tart of **A**uthority **R**ecord) yazıları olur. **SOA** domain haqqında bütün informasiya resurslarının intervalini özündə cəmləşdirir. Zone faylinin başlangıcıdır.

**Qeyd:** Zona faylinin əvvəlində yazılın SOA yazısının sintaksisində

```
@ IN SOA ns1.dns.az. admin.dns.az. {
```

'@' simvolu zona faylından kök domain adını özünə mənimsedir. Yəni **dns.az** zona faylini özünə götürür. "@"-nin əvəzinə.

```
"dns.az. IN SOA ns1.dns.az. admin.dns.az. {"
```

yazsaq, eyni işi görəcək. Ancaq burada "**dns.az.**" nöqtə ilə bitdiyinə görə, biz DNS serverimizə deyirik ki, yazılın DOMAIN adı sub deyil və kök domain-dir.

Əgər biz '**dns.az**'-in sonunda nöqtə yazmasaq, onda deyirik ki, bu, '**dns.az**' domain üçün '**dns.az**' alt (sub) domain-dir (yəni '**dns.az.dns.az**'). Bu domain adı ad təyin edilməyən "**Internet Record**"-dan IP ünvanını özünə mənimsedir.

Bu tip yazidan: "IN A 10.0.0.20"

**Qeyd:** İstənilən zona faylinin sonunda boş sətir saxlayın. Şəxsi təcrübəmdir, əks halda jurnal faylında görəcəksiniz ki, "**zone file does not end with newline**".

#### Forward zone üçün BIND data faylı

**\$TTL 3600**

- 3600 saniyədir, yəni 1saat, **60m**(dəqiqə), **1h**(saat), **1d**(gün), **1w**(həftə) yazıla bilər (Simvolsuz rəqəm yazılsa, saniyə kimi qəbul ediləcək). **TTL** - Düzgün DNS cavablarının CACHE-lənməsinin vaxtını təyin edir.

```
@ IN SOA ns1.dns.az. admin.dns.az. {
```

- '@' simvolu kök domain adına bərabərdir (yəni **dns.az**). Həmçinin dns.az. yazıla bilər, sondakı nöqtə bildirir ki, mən subdomain deyiləm. Mən kök domain-əm.

- '**ns1.dns.az**' isə **SOA** yazısının "**dns.az**" domain adının ilk NS serverini elan edir.

- '**admin.dns.az**' isə domain inzibatçısının email ünvanını təyin edir. "**admin@dns.az**", burada '@' simvolunu nöqtə əvəz edir.

**- 2010092317 ; Serial**

Bu serial hər dəfə dəyişdikdən sonra sayca **1** rəqəm artırılmalıdır. Bununla slave server özünü müqayisə edir. Dəyişiklik varsa, öz yazılarını master ilə sinxronizasiya edəcək.

**- 86400 ; Refresh**

Bu vaxt intervalından sonra **Slave** (asılı) serverlər **Primary** (əsas) serverə müraciət edib zonalarını yeniləməlidirlər.

**- 7200 ; Retry**

Əgər Slave server Primary serverə müraciət edə bilmədisə, saniyelərlə olan bu müddətdən sonra o, yenidən müraciəti təkrarlayacaq.

**- 604800 ; Expire**

Əgər bu müddət ərzində Slave server zonasını Primary serverdən yeniləyə bilmədisə, onda o, həmin zonaya xidməti dayandırmalıdır.

**- 172800 ; Minimum TTL )**

Səhv olan DNS cavablarının Cache-ləmə vaxtı. (Hansı ki, adı IP ünvana çevirmək mümkün olmadı).

; DNS serverlər

IN	NS	<b>ns1.dns.az.</b>	- DNS zonasının control-leri.
IN	NS	<b>ns2.dns.az.</b>	- DNS zonasının control-leri.

; MX Yazıları

IN	MX 10	mx.dns.az.	- 10 prioritətə sahib olan <b>mx.dns.az</b> adlı MX yazı əlavə edirik.
IN	MX 20	mail.dns.az.	- 20 prioritətə sahib olan <b>mail.dns.az</b> adlı MX yazı əlavə edirik.

<b>dns.az.</b>	IN	A	10.0.0.20
	IN	A	10.0.0.30

- Daha öncə qeyd etdiyim kimi, burada domain adı nöqtə ilə bitirirəm. Yəni elan edirəm ki, bu subdomain deyil və **10.0.0.30** IP ünvanını ona mənimşədirəm.

; Avadanlıq haqda məlumat

HINFO	"Intel Core i5" "FreeBSD x64"
TXT	"salam aleykum"

; Machine Names

<b>localhost</b>	IN	A	127.0.0.1
<b>ns</b>	IN	A	10.0.0.100
<b>ns1</b>	IN	A	10.0.0.2
<b>ns2</b>	IN	A	10.0.0.3
<b>mx</b>	IN	A	10.0.0.5
<b>mail</b>	IN	A	10.0.0.5

; Aliases

<b>www</b>	IN	CNAME	@
------------	----	-------	---

İstənilən əlavə edilmiş yazını yoxlamaq üçün '**dig**' əmrindən istifadə edə bilərsiniz:

<b>dig MX dns.az</b>	- Mail Exchange yazısını yoxlayırıq.
<b>dig A dns.az</b>	- A yazısını yoxlayırıq.
<b>dig SOA dns.az</b>	- SOA yazısını yoxlayırıq.
<b>dig NS dns.az</b>	- NS yazısını yoxlayırıq.
<b>dig HINFO dns.az</b>	- HINFO yazısını yoxlayırıq.

#### Reverse (zone) Server quraşdırılması

İmkanları:

1. Zona üçün avtoritardır.
2. IP ünvanını ada çevirir.

BIND reverse data faylinin məzmunu

```
0.0.10.in-addr.arpa.    IN SOA ns1.dns.az. admin.dns.az. {
                          2010092211      ; Serial
                          86400          ; Refresh
                          7200           ; Retry
                          604800         ; Expire
                          172800 )       ; Minimum

; NSs
IN    NS     ns1.dns.az.          - dns.az saytinin reverse zonasinin kontrolleri.
IN    NS     ns2.dns.az.          - dns.az saytinin reverse zonasinin kontrolleri.

; PTRs
2    IN    PTR    ns1.dns.az.
3    IN    PTR    ns2.dns.az.
5    IN    PTR    mail.dns.az.
```

**\$GENERATE 6-80 \$0.0.10.in-addr.arpa. PTR host-10-0-0-\$dns.az.**

- **6-80** IP ünvanı aralığını generasiya edib, növbə ilə '\$' dəyişəninə mənimsədəcək. Yəni **10.0.0.\$-a 6-80** aralığında **nslookup** eləsək, hər IP ünvan üçün PTR cavab '**host-10-0-0-\$dns.az**' aralığında olacaq. Yəni burada '\$' simvolu **6-80** IP aralığından müraciət gələn kimi həmin IP ünvanı dolların əvəzinə yerləşdirib PTR-a cavab qaytarır.

**\$GENERATE 81-253 \$0.0.10.in-addr.arpa. PTR host-10-0-0-\$edns.az.**

- Burada da '\$' dəyişəni **81-253** aralığı üçün '**host-10-0-0-\$edns.az**' cavabını qaytaracaq.

**Qeyd:** IPv4 reverse zonanı göstərilən şəkildə təyin edək. Əgər şəbəkəmiz **10.0.0.0/24**-dürse, onda şəbəkə aralığının sonunadək istifadə edirik. **10.0.0** və **reverse** budur: **0.0.10.in-addr.arpa.**

**Qeyd:** Slave serverlər susmaya görə "Zone Transfer" üçün açıq olur. Onu mütləq bağlamağı unutmayın. **"/usr/local/etc/namedb/named.conf"** faylında ilk "options" bölümündə **"allow-transfer {"none";};"** sətri əlavə etmək kifayətdir.

Yuxarıda sadalanan sətirlərini **"/usr/local/etc/namedb/master/10.0.0.rev** faylında yerləşdirildikdən sonra isə, **"/usr/local/etc/namedb/named.conf** faylında, uyğun saydığınız hissədə aşağıdakı sətirləri əlavə edin və faylı yadda saxlayın.

```
zone "0.0.10.in-addr.arpa" {
    type master;
    file "/usr/local/etc/namedb/master/10.0.0.rev";
};
```

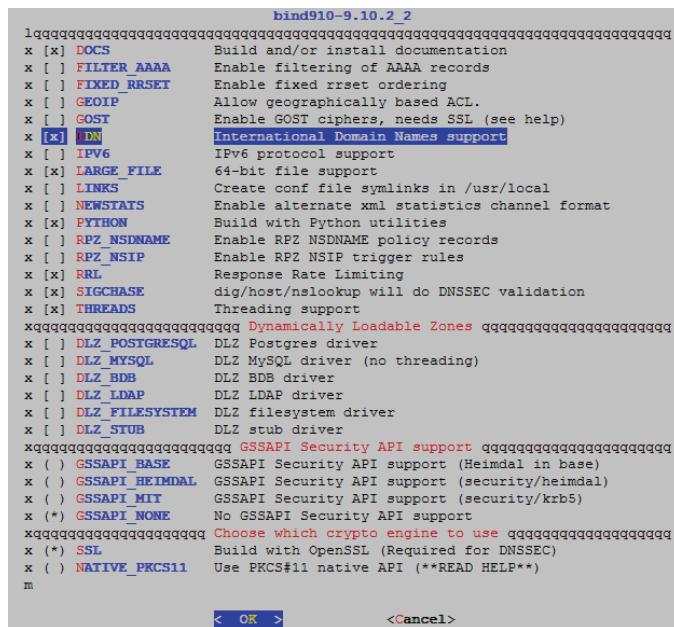
Reverse ünvanları FreeBSD serverindən yoxlamaq üçün dig əmrindən aşağıdakı kimi istifadə edirik:

```
dig -x 10.0.0.2
dig ptr 10.0.0.2
```

#### BİN910 portlardan yüklenməsi

```
cd /usr/ports/dns/bind910
make config
```

- Port ünvanına daxil oluruq.  
- Lazımı modulları seçirik.



<b>make install</b>	- Yüklenir.
<b>echo 'named_enable="YES"' &gt;&gt; /etc/rc.conf</b>	- StartUp faylımızın əlavə edirik ki, sistem yenidənyüklənməsindən sonra avtomatik işə düşsün.
<b>/usr/local/etc/rc.d/named start</b>	- İşə salırıq.

## Master/Slave Zonalar

Istənilən BIND dns serverin "Master/Slave server" funksionallığı var. Bu alt başlıqda Master/Slave DNS serveri quracaq. İki ədəd ardıcıl server qurulacaq. Öncə Master, ardınca Slave. Master server öz zona faylında etdiyi dəyişiklikləri ötürür Slave serverə. Beləliklə, istənilən istifadəçi ilk nameserver olaraq Master-i, ikinci olaraq Slave-i təyin edərsə, tələb edilən dns adı həmişə işlək vəziyyətdə olacaq. Nəzərdə tutulur ki, artıq hər iki server üçün **bind910** portlardan yüklənmişdir.

### Master DNS serverin qurulması

**/etc/rc.conf** yenidənyüklənmə faylımızın aşağıdakı sətirləri əlavə edirik:

```
hostname="master.dns.az"
sshd_enable="YES"
ifconfig_em0="inet 192.168.121.139/24"
defaultrouter="192.168.121.2"
named_enable="YES"
named_flags="-u bind -c /usr/local/etc/namedb/named.conf"
```

**/etc/resolv.conf** istifadəçi faylımızın öncə öz DNS İP ünvanımızı, sonra isə slave serverimizin İP ünvanını əlavə edirik:

```
domain dns.az
nameserver      192.168.121.139
nameserver      192.168.121.139
```

**/etc/syslog.conf** faylinin sonuna aşağıdakı sətirləri əlavə edirik ki, DNS-imizin jurnallarını ayrıca fayla yönləndirək:

```
!named
*:*
                                /var/log/named.log
```

<b>touch /var/log/named.log</b>	- DNS üçün jurnal faylı yaradırıq.
<b>/etc/rc.d/syslogd restart</b>	- Syslog-u yenidən işə salırıq.

```
/usr/local/etc/rc.d/named restart ; tail -f /var/log/named.log
```

- DNS-i yenidən işə salırıq və ardınca onlayn rejimdə jurnallarına baxırıq.

**Qeyd:** Əgər named start olunmasa və jurnallarda heç bir səhv tapa bilməsək, onda named-i CLI-dan **debug** rejimdə işə sala bilərik.

```
named -t /var/named -u bind -c /usr/local/etc/namedb/named.conf -g -d9
```

'-t' - CLI arqumentləri işə düşən kimi, göstərilən qovluğa CHROOT edin. Yəni root qovluğa qayydın.

'-u' - 'bind' istifadəçi adından yerinə yetirin.

'-c' - '/usr/local/etc/named/named.conf' faylini oxuyaraq,

'-g' - named-i ekran rejimində işə salın, yəni bütün səhvləri ekrana stderr rejimində çap edin.

'-d9' - debug səviyyəsi. (Man-da debug səviyyələri haqda çox az danışılır.)

**/usr/local/etc/namedb/named.conf** qlobal quraşdırma faylinin options bölümünü aşağıdakı kimi edirik:

```
options {  
    directory      "/usr/local/etc/namedb/working";  
    pid-file       "/var/run/named/pid";  
    dump-file      "/var/dump/named_dump.db";  
    statistics-file "/var/stats/named.stats";  
    listen-on      { 192.168.121.139; }; - Hansı IP-lərdə DNS qulaq asır?  
    allow-query     { any; };           - Müraciət hər kəsdən qəbul edilir.  
};
```

Qlobal quraşdılmalarımıza rekursiyani da əlavə edə bilərsiniz. Lakin hal-hazırda rekursiyani, testlərimizi edə bilməmiz üçün bağlamaq bize lazımdır. Sintaksis isə aşağıdakı kimi olacaq:

```
allow-recursion { 192.168.0.0/24; }; - Rekursiyaya göstərilən şəbəkə üçün izin veririk,  
                                         yəni istifadəçi bizim DNS IP-mizi istifadə edərək  
                                         'nslookup test.az' edəndə test.az zonası  
                                         haqda informasiya yalnız bizim serverdə  
                                         axtarılacaq və root DNS serverlə gedişə qadağa  
                                         qoyulacaq.  
recursive-clients 30000; - Adından məlum olduğu kimi, maksimal rekursiv  
                           istifadəçilərin sayı 30000-dir.
```

**Qeyd:** FreeBSD 10.1-dən aşağı versiyalar üçün BIND-in daxilində jurnallamanı aktivləşdirəndə daxili BUG var. Ona görə də onu aktivləşdirmədən önce bilmək lazımdır ki, jurnal üçün nəzərdə tutulmuş fayllar '`/var/log/bind/named.log`' və '`/var/log/bind/query.log`' əslində sistemin '`/var/log/bind`' ünvanından deyil, BIND üçün nəzərdə tutulmuş CHROOT qovluğundan oxunur. Və CHROOT etmisinizsə, həmin ünvanı '`/etc/rc.conf`' faylında göstərmişik. '`/var/named`' ünvanı elə məhz həmin CHROOT qovluqdur. Buna görə də 'BIND' jurnal fayllara oxumaq və ona yazmaq üçün '`/var/named/var/log/bind`' qovluğunun altında işləməyə çalışacaq. Bu səbəbdən də aşağıdakı ardıcılılığı edirik ki, onu qane edə bilək.

```
mkdir /var/named/var/log/bind           - Jurnal qovluğunu yaradırıq.  
touch /var/named/var/log/named.log /var/named/var/log/query.log  
                                         - Jurnal fayllarımızı yaradırıq.
```

```
chown -R bind:bind /var/named/var/log/bind/ - Jurnal qovluğunu 'bind'-in üzvü edirik ki, ora yetkisi olsun.
```

#### logging - jurnallama

```
logging { category lame-servers { null; }; }; - 'logging' named-ə gələn istənilən müraciəti müxtəlif jurnal səviyyəsində jurnallaya bilər.  
Həddən artıq böyük imkanları var.
```

Hal-hazırda jurnallaşma işini FreeBSD 10.1 üçün edirik. Aşağıdakı sətirləri `/usr/local/etc/namedb/named.conf` faylında qlobal `options`-dan sonra əlavə edirik:

```
logging {  
    channel default-log { file "/var/log/bind/named.log"; severity debug; print-  
    severity yes; }; - Susmaya görə jurnallar yığılacaq ünvan  
    category default { default-log; };  
    channel querylog { file "/var/log/bind/query.log"; print-time yes; }; - Müraciət jurnalları yığılacaq faylin ünvanı  
    category queries { querylog; };  
};
```

Təyin etdiyimiz jurnal faylları üçün qovluğu və faylları yaradıb lazımı hüquqları təyin edirik:

```
mkdir /var/log/bind/           - Jurnal qovluğunu yaradırıq.  
touch /var/log/bind/named.log /var/log/bind/query.log  
                                         - Jurnal fayllarını yaradırıq  
chown bind:bind /var/log/bind/named.log /var/log/bind/query.log
```

**Qeyd:** Query jurnalları yoxlamaq üçün aşağıdakı ardıcılılığı etsəniz, iş prinsipi tam aydın olacaq.

`tail -f /var/log/bind/query.log`

- UNIX DNS serverimizdə bu əmri daxil edib jurnalları onlayn analiz edirik.

Və hansısa bir Windows, ya da UNIX maşından bizim DNS serverimizə DNS müraciətlərini yollayaq. Məsələn, aşağıdakı şəkildəki kimi:

```
Name: mail.atl.az
Address: 85.132.57.61

C:\Users\Qabriel>nslookup
Default Server: google-public-dns-a.google.com
Address: 8.8.8.8

> server 192.168.121.139
Default Server: [192.168.121.139]
Address: 192.168.121.139

> mail.ru
Server: [192.168.121.139]
Address: 192.168.121.139

Non-authoritative answer:
Name: mail.ru
Addresses: 217.69.139.200
          217.69.139.202
          94.100.180.200
          94.100.180.202

>
```

`tail -f /var/log/bind/query.log`

- Eyni anda müraciət log faylımızı analiz edirik.

07-Apr-2015 05:10:29.417 client 192.168.121.1#54012 (**mail.ru**): query: mail.ru IN A + (192.168.121.139)

07-Apr-2015 05:10:29.775 client 192.168.121.1#54016 (**mail.ru**): query: mail.ru IN AAAA + (192.168.121.139)

Slave serverimizin dəyişiklikləri master serverdən yetkiyə əsasən ala bilməsi üçün **Access Control List** yaratmaq lazımdır. Bunun məqsədi ondan ibarətdir ki, bütün DNS serverlər bizdən zonaları özünə transfer edə bilər. Transfer edən tərəf artıq bizə aid olan bütün daxili resurslardan xəbərdar olacaq. Slave serverimizin IP ünvanı **192.168.121.140**-dir. "**/usr/local/etc/namedb/named.conf**" qlobal quraşdırma faylımızda logging-dən sonra aşağıdakı sətri əlavə edirik ki, yalnız özümüz və Slave serverimizin **trusted-dns** adlı ACL-in üzvü olsun.

```
acltrusted-dns { 127.0.0.1; 192.168.121.140; }; - 'trusted-list' adında acl list yaradırıq və  
üzv olaraq yalnız localhost və 192.168.121.140  
IP ünvanını əlavə edirik.
```

Artıq dns.az adı üçün master DNS zona və master PTR zonasını quraşdırıq. "/usr/local/etc/namedb/named.conf" faylına yazdığımız ACL sətrindən sonra aşağıdakı sətirləri yerləşdiririk:

```
zone "dns.az" {  
    type master;  
    also-notify { 192.168.121.139; }; - 192.168.121.139 yollayır.  
/*allow-transfer { 192.168.121.140; }*/ - 192.168.121.140 IP ünvanına zonanı transfer  
etməyə izin veririk. Ancaq biz İP əvəzinə ACL-dən  
istifadə edirik. Ona görə də bu sətir şəhər  
daxilindədir. Növbəti sətir də eyni işi görür.  
    allow-transfer { trusted-dns; }; - Zonanı transfer etmək üçün yalnız 'trusted-dns'  
    allow-update { 192.168.121.140; }; - 192.168.121.140 IP ünvanı yenilənməni qəbul  
    file "/usr/local/etc/namedb/master/dns.az.zone"; - Zona faylı təyin edilən ünvandan oxunulur.  
};  
zone "121.168.192.in-addr.arpa" {  
    type master; - Pointer zonası yaradırıq.  
    file "/usr/local/etc/namedb/master/192.168.121.rev"; - Zona faylı göstərilən ünvanda oxuyur.  
};
```

Qeyd etdiyimiz zona fayllarını yaradırıq:

```
touch /usr/local/etc/namedb/master/dns.az.zone  
touch /usr/local/etc/namedb/master/192.168.121.rev
```

/usr/local/etc/namedb/master/dns.az.zone zona faylimizin məzmunu aşağıdakı kimi  
olacaq:

```
$TTL 3600      ; 1 hour  
dns.az.      IN      SOA      ns1.dns.az. root.dns.az. {  
2015040701      ; Serial      - Unutmayın, hər dəfə Slave serverə dəyişikliyin  
                                -              gönderilməsi üçün serial dəyişməlidir.
```

```

                86400      ; Refresh
                7200       ; Retry
                604800     ; Expire
                172800     ; Minimum TTL
}

; DNS serverlər
    IN      NS      ns1.dns.az.      - nameserver1 ns1.dns.az-dir.
    IN      NS      ns2.dns.az.      - nameserver2 ns2.dns.az-dir.

; MX yazıları
    IN      MX 10   mx.dns.az.      - mx.dns.az 1-ci prioritədə durur.
    IN      MX 20   mail.dns.az.    - mail.dns.az mail serverinə 20-ci, yəni
                                    ikinci prioritət verilib.

; A yazıları
    IN      A      192.168.121.134  - Saytimiz özü 192.168.121.134 IP
                                    ünvanında yerləşir.
    ns1    IN      A      192.168.121.139 - ns1 192.168.121.139 IP ünvanındadır.
    ns2    IN      A      192.168.121.140 - ns2 192.168.121.140 IP ünvanındadır.
    mx     IN      A      192.168.121.130 - 10 prioritəli MX yazısının A yazısı
                                    192.168.121.130 IP ünvanındadır.
    mail   IN      A      192.168.121.131 - 20 prioritəli mail MX yazısının A yazısı
                                    192.168.121.131 IP ünvanındadır.

; Aliaslar
    www   IN      CNAME   @          - Sayt www ilə də istifadə edilə bilər.


```

`/usr/local/etc/namedb/master/192.168.121.rev` PTR zona faylımızın məzmununa aşağıdakı sətirləri əlavə edirik:

```

$TTL 3600
121.168.192.in-addr.arpa.    IN SOA ns1.dns.az. root.dns.az. (
                                2015040701      ; Serial  - Unutmayın, hər dəfə Slave serverə
                                                dəyişikliyin göndərilməsi üçün serial
                                                dəyişməlidir.
                                86400      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                172800     ; Minimum

```

	<b>IN</b>	<b>NS</b>	<b>ns1.dns.az.</b>	- nameserver1 ns1.dns.az-dır.
	<b>IN</b>	<b>NS</b>	<b>ns2.dns.az.</b>	- nameserver2 ns2.dns.az-dır.
<b>139</b>	<b>IN</b>	<b>PTR</b>	<b>ns1.dns.az.</b>	- ns1.dns.az IP ünvanı <b>192.168.121.139</b> ,
<b>140</b>	<b>IN</b>	<b>PTR</b>	<b>ns2.dns.az.</b>	- ns2.dns.az IP ünvanı <b>192.168.121.140</b> ,
<b>131</b>	<b>IN</b>	<b>PTR</b>	<b>mail.dns.az.</b>	- mail.dns.az-a <b>192.168.121.135</b> IP ünvanı cavab verəcək.

Və sonda DNS BIND-i yenidənyüklənmə edib jurnalları analiz edirik.

```
/usr/local/etc/rc.d/named restart ; tail -f /var/log/named.log
```

#### Slave DNS server

Slave DNS serverin vəzifəsi ondan ibarətdir ki, masterdən gələn məlumatları özündə yoxlasın və əgər yazı tiplərində fərq olarsa, masterdə oolanla özündəkini eyniləşdirsin.

**Qeyd:** Unutmayın ki, Master serverlə Slave arasında olan tarix və vaxt eyni olmalıdır. Əks halda, sinxronizasiya baş tutmayıcaq. (Hər iki serverdə **ntpdate 0.asia.pool.ntp.org** əmrini CLI-da daxil etmək lazımdır)

Slave serverimizin **/etc/rc.conf** StartUP faylı aşağıdakı kimi olacaq:

```
hostname="slave.dns.az"
sshd_enable="YES"
ifconfig_em0="inet 192.168.121.140/24"
defaultrouter="192.168.121.2"
named_enable="YES"
named_flags="-u bind -c /usr/local/etc/namedb/named.conf"
```

Slave serverimizin DNS istifadəçilərini **/etc/resolv.conf** faylında təyin edirik:

```
domain dns.az
nameserver 192.168.121.140
nameserver 192.168.121.139
```

Eynilə sistem jurnal quraşdırması faylında **/etc/syslog.conf** dns jurnallarımızı ayrı fayla təyin edirik:

```
!named
*.*                                     /var/log/named.log
```

```

touch /var/log/named.log - Ayırıcıımız jurnal faylı yaradırıq.
/etc/rc.d/syslogd restart - Syslog servisi yenidən işə salırıq.
/usr/local/etc/rc.d/named restart ; tail -f /var/log/named.log
- DNS serveri yenidən işə salaraq, onlayn
jurnallara baxırıq.

```

**/usr/local/etc/namedb/named.conf** quraşdırma faylımızda Master serverə uyğun olaraq lazımı işləri görürük(Jurnalları Masterdə olduğu kimi etmək lazımdır):

```

options { - Əsas global quraşdırımlar.
    directory      "/usr/local/etc/namedb/working";
    pid-file      "/var/run/named/pid";
    dump-file     "/var/dump/named_dump.db";
    statistics-file "/var/stats/named.stats";
    listen-on     { 192.168.121.140; }; - Hansı IP ünvanda DNS qulaq asır?
    allow-query    { any; }; - Müraciət hər kəsdən qəbul edilir.
    allow-transfer { "none"; }; - Unutmayın, qlobal quraşdırma faylında bu sətir
                                mütləq olmalıdır. Əks halda, susmaya görə
                                bütün zona faylimizi host utilit-lə transfer etmək
                                olacaq. Transferi bu əmrlə 'host -l dns.az
192.168.121.140' yoxlaya bilərsiniz.

};
zone "dns.az" { - dns.az saytı üçün zona yaradırıq.
    type slave; - Tipi Slave olur.
    file "/usr/local/etc/namedb/slave/dns.az.zone"; - Faylı göstərilən ünvana təyin edirik.
masters { - Master-dən oxuyur.
    192.168.121.139; - Master-in IP ünvanı 192.168.121.139-dur.
};
};
zone "121.168.192.in-addr.arpa" { - Şəbəkə üçün PTR yaradırıq.
    type slave; - Tipi Slave-dir.
    file "/usr/local/etc/namedb/slave/192.168.121.in-addr.arpa"; - PTR-i göstərilən fayldan oxuyur.
masters { - Master-dən oxuyur.
    192.168.121.139; - Master-in IP ünvanı 192.168.121.139-dur.
};
};

```

```
/usr/local/etc/rc.d/named restart ; tail -f /var/log/named.log
```

- DNS-i yenidən işə salıb jurnallara baxırıq, görək ki, bizə master serverdən yazılar gelir, ya yox.

Və sonda ad və PTR üçün zonalarımızın fayllarını açıb yoxlamaq istəsək, faylların fərqli sintaksisdə olduğunu görəcəyik.(Fayllar BIND9.9-dan başlayaraq raw formatda yazılır.) Bu fayllar hələ ki raw formatda olacaq, cünki bind susmaya görə Slave DNS fayllarını RAW formatda yazır. Ancaq bu, problem deyil və siz bu faylları adı mətn formatına və geriye problemsiz konvertasiya edə bilərsiniz. Aşağıdakı əmrlərdən istifadə edə bilərsiniz. RAW formatdan mətnə konvertasiya:

```
#[command]      (format options)  (output file) (zone origin) (input file)
named-compilezone -f raw -F text -o dns.az.text dns.az dns.az.raw
```

Həmçinin mətn formatdan raw formata:

```
#[command]      (format options) (output file) (zone origin) (input file)
named-compilezone -f text -F raw -o dns.az.raw dns.az dns.az.text
```

**Qeyd:** Əgər biz BIND-i **rndc** utilit tərəfindən idarə etmək istəsək, aşağıdakı qaydaları etmək lazımdır(Şəxsi praktikamda, demək olar ki, istifadə etmirəm). RNDC utilit named-i CLI-dan tam idarə etmək üçün istifadə edilir.

#### **rndc-confgen**

- CLI-dan əmri daxil etdiğdə, utilit avtomatik fayl generasiya edir və açıq şəkildə göstərir ki, hansı sətirləri '**/usr/local/etc/namedb**' ünvanında "**rndc.conf**" faylinə əlavə edəcəyik. Və hansı sətirləri '**named.conf**' faylinin sonuna əlavə edəcəyik. Generasiyadan alınan sətirlər aşağıdakı kimi olur.

```
/usr/local/etc/namedb/rndc.conf
```

```
key "rndc-key" {
    algorithm hmac-md5; - hmac-md5 alqoritmi ilə pre-sharedkey şifrələnir.
    secret "bDUIgLPUU7a6FSk4Up8Ydg=="; - Paylaşılmış açar.
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1; - Localhost.
```

```
    default-port 953;           - 953-cü porta qoşulur.  
};
```

**/usr/local/etc/namedb/named.conf** faylinın sonuna aşağıdakı sətirləri əlavə edirik:

```
key "rndc-key" {  
    algorithm hmac-md5;      - hmac-md5 alqoritmi ilə pre-sharedkey şifrələnir.  
    secret "bDUIgLPuU7a6FSk4Up8Ydg=="; - Paylaşılmış açar.  
};  
controls {  
    inet 127.0.0.1 port 953   - Localhost 953-cü portda idarə etməyə  
        allow { 127.0.0.1; } keys { "rndc-key"; }; izin veririk,  
        127.0.0.1-i rndc-key düzgün olan halda.  
};
```

**rndc reload** - CLI-dan bu əmr daxil edildikdə name serveri asanlıqla yenidənyüklənmə edir. '**/usr/local/etc/rc.d/named restart**'-in əvəzinə.

**rndc stop** - BIND-i tamamilə dayandırır. Qeyd: Əgər bu əmrden sonra '**rndc start**' etsəniz, **rndc** localhost 953-cü porta qoşula bilməyəcək. Çünkü Bind artıq **stop** edilib. ☺

**rndc status** - BIND-in statusunu tam çap edəcək.  
**rndc flush** - Bütün named-in keşini flush edir.

DNS-lerdə axtarış etmək üçün çox gözəl bir utilit mövcuddur. Onun adı **DNSwalk**-dir. Onu yükleyək və kiçik test edək.

```
cd /usr/ports/dns/dnswalk  
make install clean  
rehash
```

- Port ünvanına daxil oluruq.  
- Yükleyirik.  
- HASH-i yeniləyirik ki, əmr işləsin.

**dnswalk az. > azlist &** - Əmrələ saatlar keçməlidir ki, az domain-in altında olan bütün altdomain-ləri təpaq.

**Qeyd:** Unutmayın, root domain yazılanda mütləq sonda nöqtə ("az.") yazın, əks halda işləməyəcək.

DNS müraciətlərinə onlayn rejimdə baxmaq istəsək, DNSTop adlı paketdən istifadə edə bilərik.

<b>cd /usr/ports/dns/dnstop</b>	- Port ünvanına daxil oluruq.
<b>make install</b>	- Yükləyirik.
<b>dnstop -Q em0</b>	- Onlayn rejimdə 'em0' şəbəkə kartında '-Q' müraciətlərin sayına baxırıq.

## NSSWITCH

Ad xidmətləri arasında keçid üçün istifadə edilən keçid quraşdırma faylıdır. Əməliyyat sistemi adın IP ünvanına çevriləməsi üçün ilk olaraq **/etc/hosts** faylinə müraciət edir. Əgər bu faylda lazımi ada cavab qayıtmazsa, DNS-ə müraciət edilir. Ancaq biz bu növbələşməni özümüz idarə edə bilərik. Yəni **nsswitch.conf** faylında dəyişiklik edərək qeyd edə bilərik ki, öncə DNS-ə müraciət olmalıdır, sonra **/etc/hosts** faylinə. Ancaq nəzərə alın ki, **nsswitch.conf** faylı təkçə IP-nin ada və adın IP ünvana çevriləməsi üçün istifadə edilmir. O, **/etc/services** faylında olan port rəqəmlərin adlara çevriləməsində, **/etc/protocols** faylında olan protokol adlarının rəqəmlərə çevriləməsində, həmçinin UİD,GİD-in istifadəçi adlarına çevriləməsində iş ardıcılığını təyin edir.

<b>/etc/nsswitch.conf</b>	- Faylda sintaksis aşağıdakı kimidir.
<b>hosts: files dns</b>	

NSSWITCH-in idarəciliyində olan fayllar aşağıdakılardır:

groups Group membership checks (/etc/group)  
hosts Hostname and IP checks (/etc/hosts,DNS)  
networks Network entries (/etc/networks)  
passwd Password entries (/etc/passwd)  
shells Checks for valid shells (/etc/shells)  
services TCP and UDP services (/etc/services)  
rpc Remote procedure calls (/etc/rpc)  
proto TCP/IP network protocols (/etc/protocols)



# BÖLÜM 11

## Samba, AD ilə integrasiyası, badsect-lar, CLRI, NullFS, clonehdd

- / FreeBSD Samba
- / Samba server Active Directory Authentication
- / badsect – badblock (korlanmış disk blokları) faylların köçürülməsi üçün program
- / CLRI - clear an inode, NullFS
- / Sıradan çıxmış fayl sistemin geri qaytarılması, clonehdd - sərt diskin hissəsinin digərinə nüsxələnməsi

İstənilən Unix/Linux və Windows inzibatçıya informasiyanın fərqli əməliyyat sistemləri üzərində paylaşımı hansısa bir zamanda adı istək yox, mütləq tələbə çevrilə bilər. Bu başlığımızda UNIX/Linux əməliyyat sistemlərində olan Samba program təminatının qurulması və imkanları açıqlanır (eynilə onun WEB browser vasitəsilə idarə edilməsi). Həmçinin UNIX/Linux əməliyyat sistemlərində olan qovluqların Windows əməliyyat sistemlərinə və əksinə yayılması barədə məlumat verilir. Sözsüz ki, UNIX/Linux serverin Windows domain controller-ə üzv olunması da açıqlanacaq. Serverinizin diskində korlanmış disk bloklarının bərpa edilməsi metodikası, indeks deskriptorlarının təmizlənməsi, Null fayl sistem, korlanmış fayl sistemin bərpa edilməsi və diskin bütövlükdə klon edilməsi haqda danışılır.

# FreeBSD Samba

Samba - **SMB/CIFS** protokolları vasitəsilə fərqli əməliyyat sistemlərində olan şəbəkə disklərinə və printerlərə müraciət etmək imkanı yaradır. Server və client hissələrdən ibarətdir. GPL lisenziyası altında işləyən pulsuz program təminatıdır.

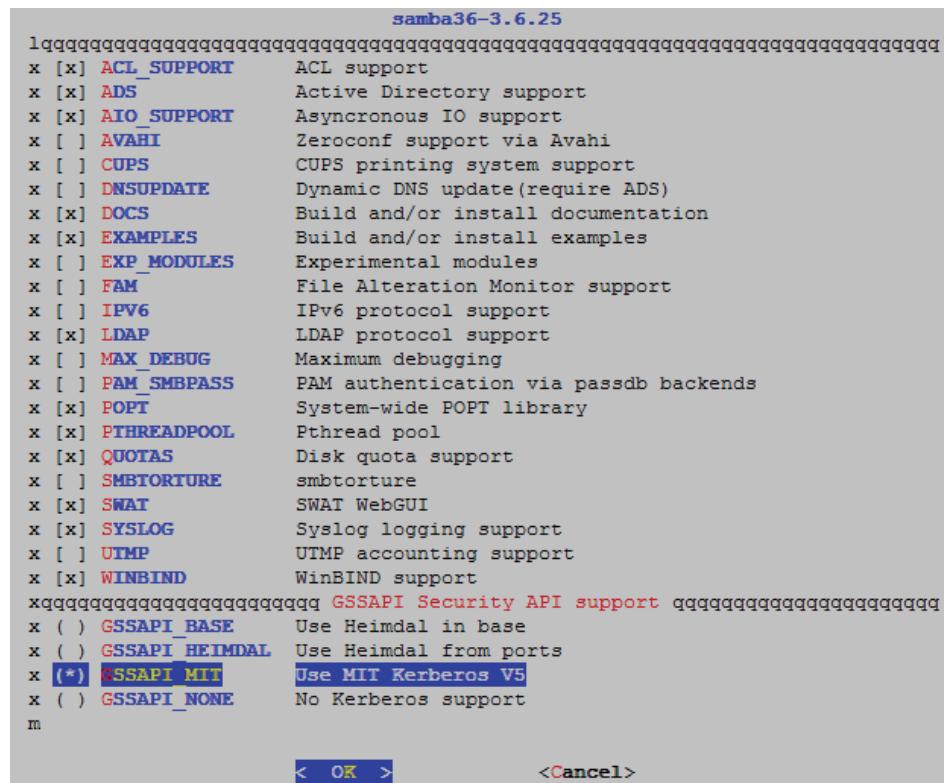
Məqsədimiz hansısa Windows maşinindən paylaşılmış qovluğu UNIX maşınına mount etmək və əksinə UNIX maşinindən paylaşılmış qovluğu Windows maşinində istifadə etməkdir. Yükleyək və quraşdırma işlərimizə başlayaq.

```
cd /usr/ports/net/samba36
```

- Port ünvanına daxil olurq.

**make config**

- Lazımı modulları seçirik.



```
make install clean
```

- Yükləyirik.

FreeBSD maşnimizda StartUP quraşdırma faylımız **/etc/rc.conf**-un məzmunu aşağıdakı kimi olacaq.

```
samba_enable="YES"  
inetd_enable="YES"  
ifconfig_em0="inet 192.168.121.139 netmask 255.255.255.0"  
defaultrouter="192.168.121.1"  
hostname="samba.freebsd.lan"
```

```
cp /usr/local/etc/smb.conf.sample /usr/local/etc/smb.conf
```

- Sambanı nüsxə faylından digər fayla köçürürik ki, daemon-u işə sala bilək.

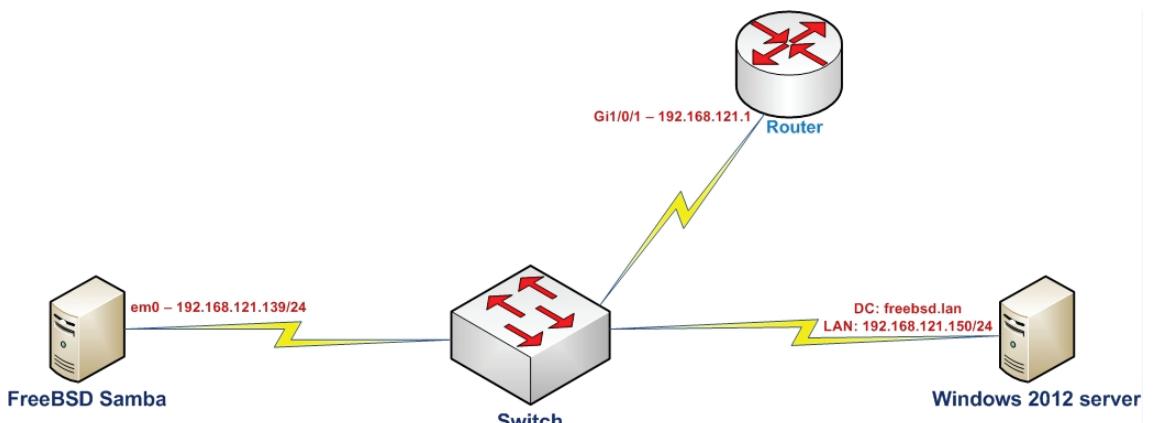
```
/usr/local/etc/rc.d/samba start
```

- Samba serverimizi işə salırıq.

Windows olaraq server 2012 istifadə edilmişdir. Domain Controller qaldırılmış vəziyyətdədir. Domain adı **freebsd.lan**-dır. İP ünvanı isə **192.168.121.150**-dir.

**Qeyd:** Windows maşında server adını **w2k12** adına dəyişməyi unutmayın.

Windows maşnimizla FreeBSD maşnimiz aşağıdakı şəkildəki quruluşda bir-birlərini görürələr:



SMBD daemon haqda bəzi məlumatları açıqlayaq.

<b>netbios</b>	- Maksimal <b>16</b> simvol uzunlığunda simvollar dəstəkləyir.
<b>samba</b>	- İki port ilə işləyir <b>TCP(139/445)</b> .

- nmbd** - Samba netbios name-i registr için istifadə edir **UDP (137,138)**.  
**winbind** - Windows və Unix user və qrupları arasında əlaqə yaradır.  
**smbstatus** - Hal-hazırkı Samba qoşulmalarının statusunu çap edir.

**Qeyd:** Əmri işə salıb yoxlamazdan əvvəl lazımi quraşdırımlar(paylaşılmış qovluq və istifadəçi adı) öncədən edilməlidir. Hansısa windows maşından ən azı bir qoşulma etməlisiniz ki, nəticə ala biləsiniz.

Samba version 3.6.25

PID	Username	Group	Machine
7775	root	wheel	daemon (192.168.121.1)

Service	pid	machine	Connected at
IPC\$	7775	daemon	Fri Apr 10 08:42:49 2015
root	7775	daemon	Fri Apr 10 08:42:49 2015

Locked files:

Pid	Uid	DenyModeAccess	R/W	Oplock	SharePath	Name	Time
							-----7775 0
DENY_NONE	0x100081	RDONLY	NONE	/root .			Fri Apr 10 08:42:51 2015

- smbtree** - Samba paylaşımlarının hamısını gösterir (netbios ada windows vasitəsilə müraciət edə bilirik).

WORKGROUP

MYGROUP

\MASTER	Samba Server
\MASTER\root	Home Directories
\MASTER\IPC\$	IPC Service (Samba Server)

- smbtree -v** - Şəbəkədə gördüyü bütün Netbios adları çap edəcək.

WORKGROUP

\ZBOOK
\USER-HP
\SEYMUR

```

\\SALMAN-PC
\\PC
\\NZ-PC
\\GEORGIYS2
\\ELMAN-PC
\\DAEMON
\\ARPADARAIS-PC

MYGROUP
    \\MASTER           Samba Server
        \\MASTER\IPC$          IPC Service (Samba Server)

```

**smbtree -d** - Bütün domain-lerin siyahısını çap edirik.

WORKGROUP

MYGROUP

## Samba Client

**Qeyd:** Öncədən bildirmək istərdim ki, samba istənilən windows maşın tərəfindən paylaşılmış resursa yetki vermək və almaq imkanına malikdir. Siz bu imkandan istifadə etmək üçün **smbcacls** əmrindən istifadə edə bilərsiniz.

**smbcacls //192.168.121.150/papka pp -U cavid** - Windows serverımızdə **papka** adı ilə paylaşılmış qovluğun içinde **pp** adlı qovluğun yetkilərinə **cavid** adlı istifadəçi adından baxırıq.

Enter cavid's password:

REVISION:1

CONTROL:0x8404

OWNER:BUILTIN\Administrators

GROUP:WIN-9JFOLDODA91\None

ACL:NT AUTHORITY\SYSTEM:ALLOWED/OI|CI|I/FULL

ACL:WIN-9JFOLDODA91\cavid:ALLOWED/OI|CI|I/FULL

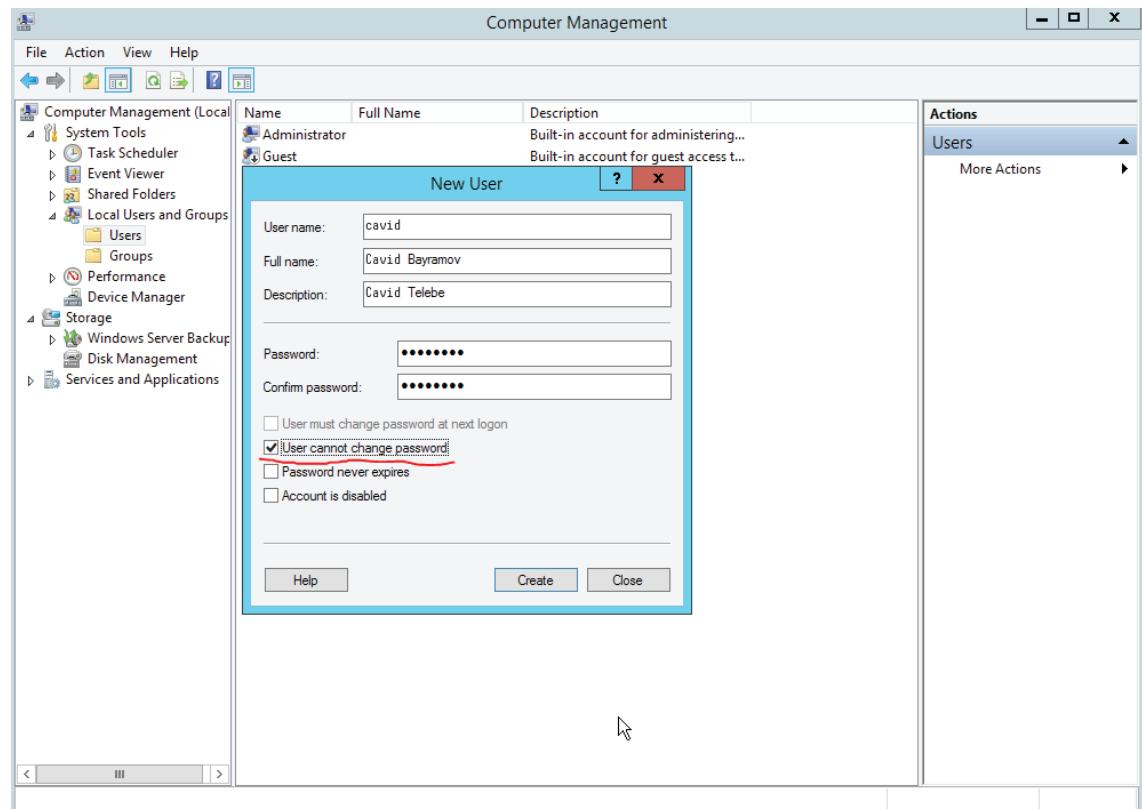
ACL:WIN-9JFOLDODA91\Administrator:ALLOWED/OI|CI|I/FULL

ACL:BUILTIN\Administrators:ALLOWED/OI|CI|I/FULL

**mget** - Windows paylaşılmış qovluqdan lazımi informasiyanı götürür.

**mput** - Windows paylaşılmış qovluğa lazımi informasiyanı ötürür.

Aşağıdakı şəkillərdə göstərildiyi kimi, Windows 2012 serverimizdə **cavid** adlı istifadəçi yaradırıq və **papka** adlı qovluq yaradıb cavid istifadəçisinin həmin qovluqdan istifadəsi üçün yetki veririk.



**smbclient //192.168.121.150/papka -U cavid** - FreeBSD serverimizdən Windows 2012 serverin papka adlı paylaşılmış qovluğuna cavid istifadəçi adı ilə qoşulurq. Açılmış konsol-da **help** əmrini daxil etsəniz, bütün imkanlar çap ediləcək.

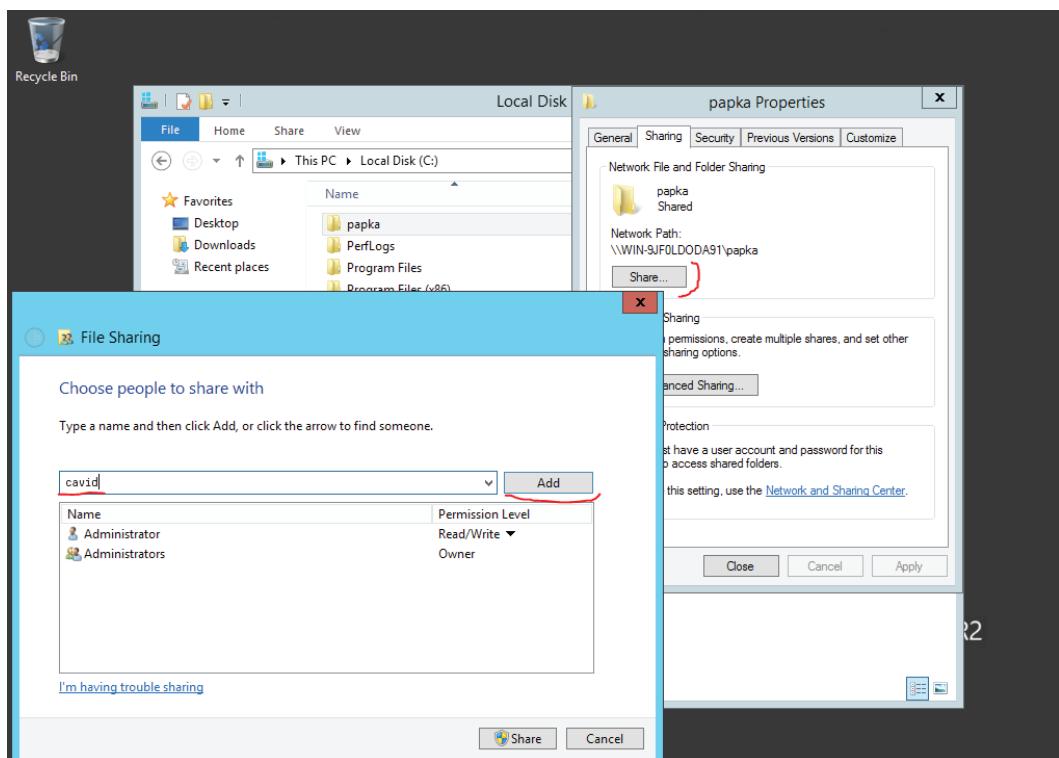
Enter cavid's password:**cavidinshifresi**

Domain=[WIN-9JFOLDODA91] OS=[Windows Server 2012 R2 Standard 9600] Server=[Windows Server 2012 R2 Standard 6.3]

smb: \>

**smbclient -U root -L 192.168.121.139**

-Samba serverimizin özünə root istifadəçi adı ilə qoşulurq.



**Qeyd:** Ancaq öncədən sistemdə samba istifadəçi bazasında '**root**' istifadəçisi üçün şifrə təyin etməyi unutmayın. Əks halda, qoşula bilməyəcəksiniz.

**smbpasswd -a root**

- root istifadəçisi üçün şifrə təyin edirik ki, önceki əmrde həmin şifrəni istifadə edək.

**smbclient -U Administrator -L 192.168.121.150**

- Administrator istifadəçi adı ilə **192.168.121.150** IP ünvanlı windows serverin bütün paylaşımlarının siyahısını çap edəcək.

**smbclient -A /root/winpass //192.168.121.150/papka - 192.168.121.150** IP ünvanlı windows serverində papka adlı paylaşılmış ünvana **/root/winpass** adlı fayldan istifadəçi adı və şifrəni oxuyaraq qeydiyyatdan keçirik.

```
cat /root/winpass  
username = cavid  
password = N123456n
```

- İstifadəçi faylin məzmunu isə aşağıdakı kimidir.

## Samba TAR

Smbtar əmri vasitəsilə windows maşında olan paylaşılmış qovluqlara və içindəki fayllara baxmadan onları birbaşa FreeBSD maşınımiza köçürə bilərik.

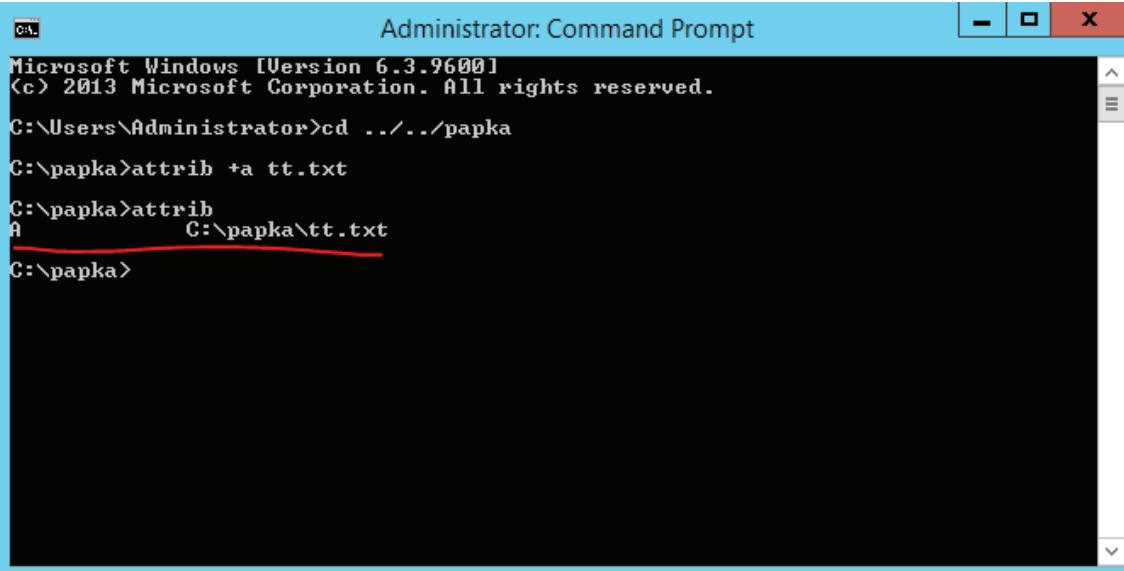
Aşağıda göstərilən ilk misalımızda **192.168.121.150** IP ünvanlı windows serverə **cavid** istifadəçi adı ilə və **A123456** şifrəsi ilə qoşulub, papka adlı paylaşılmış qovluqda olan bütün məlumatları **tar** edib **windows.tar** adlı fayla yazırıq.

```
smbtar -s 192.168.121.150 -u cavid -t windows.tar -p A123456a -v -x papka ->  
      -s -Server,  
      -u -İstifadəçi adı,  
      -t -Adı fayl və ya yazılı bilən tape(kaset) alət,  
      -p -İstifadəçi şifrəsi,  
      -v -Ətraflı məlumat rejimi,  
      -x -Paylaşılmış qovluğun adını gözləyir.
```

```
server    is 192.168.121.150  
share     is papka\\  
tar args  is  
tape      is windows.tar  
blocksize is  
added interface em0 ip=192.168.121.139 bcast=192.168.121.255 netmask=255.255.255.0  
Domain=[WIN-9JFOLDODA91] OS=[Windows Server 2012 R2 Standard 9600] Server=[Windows  
Server 2012 R2 Standard 6.3]  
tarmode is now full, system, hidden, noreset, verbose  
directory \pp\  
      36 ( 35.2 kb/s) \pp\yenifayl.txt  
      18 ( 17.6 kb/s) \tt.txt  
tar: dumped 3 files and directories  
Total bytes written: 1024
```

Bu halda "-a" opsiyası windows serverdə arxiv olunmuş faylların hamısını **reset** edir və arxiv olma haqda heç bir sübut saxlamır.

**Qeyd:** Yeni windows serverdə olan paylaşılmış fayl və ya qovluğa FreeBSD server tar edib faylları götürəndə heç bir **flag** (işarə) yerləşdirmir. Siz flag-lara Windowsda paylaşılmış qovluğun daxilində windows-un CLI-indən '**attrib**' əmri yığaraq görə bilərsiniz.



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command history is as follows:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd ../../papka
C:\papka>attrib +a tt.txt
A          C:\papka\tt.txt
C:\papka>
```

The line "A C:\papka\tt.txt" is highlighted with a red underline.

```
smbtar -s 192.168.121.150 -u cavid -t windows1.tar -p A123456a -v -a -x papka
```

### Samba Mount

Windows paylaşımının FreeBSD serverimizdə mount olunması üçün kernel-i aşağıda göstərilən opsiyalarla kompilyasiya etmək lazımdır.

```
options      NETSMB      #SMB/CIFS requester
options      LIBMCHAIN   #mbuf management library
options      LIBICONV
options      SMBFS
```

**Qeyd:** Windows paylaşımı **UNIX/Linux** əməliyyat sisteminə mount etdikdə nəzərə alın ki, Unix/Linux serverdə iki və daha çox şəbəkə kartı varsa, samba susmaya görə onun ilk şəbəkə kartından çıkış etməyə çalışacaq. Bunun qarşısını almaq və ya hər iki şəbəkə kartında samba çıkışını aktivləşdirmək istəsək, aşağıdakı dəyişiklikləri '/usr/local/etc/smb.conf' faylında yazmaq lazımdır.

```
/usr/local/etc/smb.conf faylında aşağıdaki dəyişiklikləri edirik:  

interfaces = em1 - SAMBA-ya deyirik ki, ikinci şəbəkə kartımızda  
dinlə.  

;interfaces = 192.168.105.10/24 172.16.1.1/24 - Ya da hər iki şəbəkə kartımızın IP ünvanını təyin  
edə bilərik.  

hosts allow = 192.168. 127. 172.16. - Daxil olmaq üçün izin verdiyimiz şəbəkə  
aralıqları.  

wins server = 172.16.1.3 - Sambaya Wins serverin IP-sini göstəririk.  
(Windows Server)
```

Windows maşınında olan qovluğu FreeBSD maşınınə mount etmək sintaksis şablonu aşağıdakılardır:

```
mount_smbfs -I winserver_IP -W domain_name_of_windows.com //username@computername_  
of_windows/windows_folder$ /winserverdisk
```

Aşağıda göstərilən ilk misalımızda açıqlanmanı ardıcıl olaraq göstərək.

```
mkdir /winserver - FreeBSD serverimizdə öncədən mount  
edəcəyimiz qovluğu yaradırıq.
```

```
mount_smbfs -I 192.168.121.150 -W freebsd.lan //Administrator@w2k12/papka /winserver  

-I - Windows Domain Controller-in IP ünvanını  
parametr kimi tələb edir.  

-W - Windows Serverdə yaratdığımız Full  
Qualified Domain Name-i tələb edir, yəni  
DC-nin adı. (Bizim halda o, freebsd.lan-dir.)  

Administrator - Windows istifadəçinin adı.  

w2k12 - Windows maşının computer name-idir.  

winserver - FreeBSD maşınınızda yaratdığımız  
qovluqdur, hansı ki, windows 'papka' adı ilə  
paylaşılmış qovluğu bura mount edəciyik.  

df -h | grep winserver - Nəticəyə baxırıq.  

//ADMINISTRATOR@W2K12/PAPKA      59    8    51    14%    /winserver
```

Öncəki sətirlə eyni işi görür, ancaq burada qovluğa yazacağımız, oxuyacağımız informasiyaya kodlaşdırma təyin edirik və disk olaraq Windows C:-ni tam götürürük. (Ən yaxşısı **utf8**-dir.)

```
mount_smbfs -I 192.168.121.150 -E koi8-r:cp866 -W freebsd.lan  

//Administrator@w2k12/c$ /winserver
```

Biz həmçinin aşağıda göstərilən sintaksisə Windows diskləri **/etc/fstab** faylında StartUP-a əlavə edə bilərik ki, sistem yenidənyüklənməsindən sonra disklər avtomatik mount edilsin.

**/etc/fstab** faylının sonuna aşağıdakı sətirləri əlavə edirik:

```
//Administrator@W2K12/papka    /winserver smbfs  rw      0      0  
//Administrator@W2K12/C$        /mnt      smbfs  rw      0      0
```

**Qeyd:** Ancaq sistem yenidənyüklənməsindən sonra paylaşılmış qovluğun şifrəsinin soruşulmasını istəməsək, **/etc/nsmb.conf** faylında lazımi quraşdırmaları etməliyik.

**smbutil crypt A123456789a**

- 'smbutil' əmri ilə konsoldan '**A123456789a**' adlı şifrəni **crypt** (şifrləyirik) edirik.

**\$\$1551907717b695f18**

- Alınan nəticəni '**/etc/nsmb.conf**' faylında istifadə edəcəyik.

**/etc/nsmb.conf** Domain Controller-lə autentifikasiya faylımızla lazımi sətirləri əlavə edirik ki, qeydiyyatımızı avtomatlaşdırıq:

**[default]**

**workgroup=FREEBSD.LAN**

- Windows-da yaratdığımız Domain Controller-in FQDN adı. Mütləq böyük hərflərlə olmalıdır.

**[W2K12]**

**addr=192.168.121.150**

- Windows Domain Controller maşının Computer Name-i. Mütləq böyük hərflərlə olmalıdır.  
- Windows Domain Controller maşının IP ünvanıdır.

**[W2K12:ADMINISTRATOR]**

**password=\$\$1551907717b695f18**

- Windows Domain Controller maşının Computer Name-i və **Administrator** istifadəçi adı. Mütləq böyük hərflərlə olmalıdır.

- Windows Domain Controller maşının **crypt** edilmiş **ADMINISTRATOR** istifadəçi adının şifrəsi.

Sistemə yenidənyüklənmə əmri daxil edirik ki, diskin avtomatik mount edilməsini yoxlayaq.

**df -h | grep -i administrator**

- Disklərin olmasını yoxlaysınq.

**//ADMINISTRATOR@W2K12/PAPKA**

**60G 8.5G 51G 14% /winserver**

**//ADMINISTRATOR@W2K12/C\$**

**60G 8.5G 51G 14% /mnt**

## Samba istifadəçilərin yaradılması

Samba avtorizasiyanı iki yerdə keçir:

1. **/etc/passwd**
  2. Samba özü yaratdığı bazadan **smbpasswd -a username**
- Yaradılan istifadəçi adı mütləq faylda olmalıdır.
- Samba öz bazasında həmin istifadəçi üçün şifrə təyin edir. (Windows maşından UNIX istifadəçi adı və samba şifrəsi ilə samba serverə qoşuluruz.)

**Qeyd:** Əgər bir istifadəçi adı ilə FreeBSD tərəfindən paylaşılmış ünvana Windows-dan daxil olsanız, unutmayın ki, o, istifadəçi adını və şifrəni öz Cache-ində saxlayır. Yeni istifadəçi adını yoxlamaq üçün köhnəni ya silin, ya da yenidənyüklənmə edin ki, digər istifadəçi adı ilə daxil ola biləsiniz.

**/usr/local/etc/smb.conf** faylında WORKGROUP adını, server adını və NETBIOS-a alias təyin edə bilərik.

**workgroup = MYGROUP**

- Windows axtarış verdikdə MYGROUP adla tapacaq.

**server string = PFSERVER**

- Samba serverımız windows-da PFSERVER adı ilə görünəcək.

**netbios aliases = UNIX\_FreeBSD**

- Windows maşın samba paylaşım axtardıqda samba server UNIX\_FreeBSD adı ilə netbios aliasi edilir. (Yəni şəbəkədə iki müxtəlif adlı eyni server görünəcək)

## **Paylaşma təhlükəsizlik səviyyələri**

**/usr/local/etc/smb.conf** faylımızda paylaşım yaratmaq üçün aşağıdakı sətirləri uyğun olaraq quraşdırırıq:

**[global]**

**workgroup = MYGROUP**

- Samba serverimizə NetBIOS ad veririk.

**server string = Samba Server**

- Serverimizin Comment adı.

**security = user**

- Bu başlıq çox önemlidir. Bunun sayəsində Samba istifadəçinin necə qeydiyyatdan keçməsini təyin edir. Əgər siz SAMBA-ya

'**security = user**' seçsəniz, o, 'Windows 98' və 'Windows NT' ilə danışmaq üçün istifadə edilir. Əgər sizin PC-ləriniz həm UNIX-də olan və həm də olmayan istifadəçi adlarından istifadə edirsə, onda '**security = user**' etməyiniz məsləhətdir.

```

hosts allow = 192.168. 127. 172.16.
load printers = no
log file = /var/log/samba/log.%m

max log size = 50
interfaces = em0 em1
wins server = 172.16.1.3

dns proxy = no

[homes]
comment = Home Directories
browseable = no
writable = yes

[public]
comment = Public Stuff
path = /home/samba
    valid users = cavid root@account
public = yes
writable = yes
printable = no

[cavid]
comment = Camal's Share
valid users = cavid
read only = no

create mask = 0765
path = /home/cavid

veto files = /*.mp3/*.wav/*jpg

```

- Sambaya giriş izni olan şəbəkələr.
- Printerlərin yüklənməsinin qarşısını alırıq.
- Jurnallar yığılan ünvanı təyin edirik.
- '%m' - Hər qoşulan Host-un IP ünvanı üçün ayrıca fayl generasiya ediləcək.
- Maksimum jurnal həcmi 50KBayt təyin edirik.
- Hansı interfeyslərdə Samba işləyir?
- WINS server Domain Controller-imizin IP ünvanını göstəririk.

- Yalnız **cavid**, **root** istifadəçi adının və **account** adlı qrupun üzvlərinin **/home/samba** qovluğununa yetkisi olacaq.

- Cavid adlı yeni paylaşılmış qovluq yaradırıq.
- Uyğun olan şərh yazırıq.
- Cavid istifadəçi adı ilə qoşulmaq izni təyin edirik.
- Ancaq "oxumaq olsun" yetkisinə yox deyirik ki, yazmaq da olsun.
- Qovluğa '**chmod 765**' yetkisi veririk.
- İstifadəçi hansı UNIX maşınının hansı qovluğunda olacaq?
- '**mp3**', '**wav**', '**jpg**' genişlənməli faylların **/home/cavid** adlı paylaşılmış qovluğuna yazılıması qadağandır.

```
public = yes  
writable = yes  
browsable = yes
```

- Public-də olsun.
- Yazmaq mümkün olsun.
- Gizli olmasın.

Printer quraşdırmaq üçün FreeBSD maşinimizə "cd /usr/ports/print/cups" yüklenməlidir, sonra smb.conf-da "/etc/printcap" ünvani təpilir və içində lazımi printer üçün driver quraşdırılır.

**Qeyd:** Samba paylaşılmış qovluğa anonim giriş təmin etmək üçün öncə "/usr/local/etc/smb.conf" faylında qlobal quraşdirmalarda **guestaccount = nobody** sətrini əlavə etmək və sonra nəzərdə tutduğumuz qovluğun quraşdirmalarında **guest ok = yes** sətrini əlavə etmək lazımdır.

## Samba SWAT

Əgər siz samba serverinizi WEB browser vasitəsilə idarə etmək istəyirsinizsə, SWAT-dan istifadə edə bilərsiniz. Ancaq samba-nı kompilyasiya etdiğdə, mütləq SWAT seçmək lazımdır. "/etc/rc.conf" StartUp faylinə **inetd\_enable="YES"** sətri əlavə edib, inetd-ni işə salmağı unutmayın. /etc/inetd.conf faylında aşağıdakı sətrin qarşısından şərhini silirik:

```
swat      stream  tcp      nowait/400      root      /usr/local/sbin/swat      swat
```

Bu sətri aktiv edirik və browser vasitəsilə [http://server\\_ip:901-ci](http://server_ip:901-ci) port ilə qoşuluruq. Sonra istənilən quraşdırmanın web vasitəsilə etmək mümkün olacaq.

The screenshot shows the SWAT web interface for managing a Samba server. At the top, there's a navigation bar with links like 'HOME', 'GLOBALS', 'SHARES', 'PRINTERS', 'WIZARD', 'STATUS', 'VIEW', and 'PASSWORD'. Below the navigation bar, the title 'Welcome to SWAT!' is displayed, followed by a note: 'Please choose a configuration action using one of the above buttons.' A sidebar on the left titled 'Samba Documentation' lists various tools and utilities:

- Daemons**:
  - `bind` - the SMB daemon
  - `mbd` - the NetBIOS nameserver
  - `nmblookup` - the NetBIOS broadcast daemon
- Configuration File**:
  - `smb.conf` - the main Samba configuration file
  - `smbd` - the SMB daemon
  - `nmbd` - the NetBIOS daemon
  - `smbsess` - SMB password file
- Administration**:
  - `subcmd` - send control messages to Samba daemons
  - `net` - tool for administration of Samba and remote CIFS servers
  - `smbtorture` - SMB stress test tool
  - `tdb2dump` - Tool for backing up TDB databases
- Clear Text Authentication**:
  - `rpcclient` - command line MS-RPC client
  - `smbclient` - command line SMB client
  - `smbdump` - helper utility for mounting SMB filystems on Linux hosts
  - `smbmount` - helper utility for mounting SMB filystems under Linux
  - `smbmigrate` - user space tool for mounting SMB filystems under Linux
  - `smbcrypt` - get or set quotas on NTFS 5 shares
  - `smbcryptfs` - SMB file system for Linux with FUSE
  - `smbdump` - Text-based SMB network browsing
- Diagnostic Tools**:
  - `smbdump` - monitoring Samba
  - `testparm` - validating your config file
  - `smbtorture` - SMB stress test tool
  - `smbfinger` - Tool for getting windows information
- Misc Utilities**:
  - `profiles` - migrating profiles from one domain to another
  - `smbdump` - dumping profiles from samba log files
- Books**:
  - `Samba 3rd` - by Ivo T. Robert Eckstein, and David Collier-Brown
  - `The Official Samba HOWTO-DSN and Reference Guide`
  - `Samba In Practice`
  - `The Samba Developers Guide`

At the bottom of the page, there's a 'Feedback' section and a note: 'Please join the [samba](#) mailing list if you want to discuss issues with this release of SWAT.'

**192.168.121.150** IP ünvanlı serverdən istifadəçi adı cavid və A123456a şifrəsi ilə paylaşılmış test qovluğundan **tt.txt** faylini öz serverimizə nüsxələyirik.

```
smbget -u cavid -p A123456a smb://192.168.121.150/papka/tt.txt
```

```
nmblookup -B root mpd
```

- Samba serverdən root istifadəçi adı ilə "**mpd**" samba adının IP ünvana çevrilməsinə müraciət edir. Bu müraciət broadcast vasitəsilə edilir (**nmblookup** kimi).

Məs: "nmblookup -B Administrator W2K12" querying W2K12 on 192.168.121.255  
192.168.121.150 W2K12<00>

#### Windows serverdə istifadə ediləcək əmrlər

```
net view
```

```
net share
```

```
net use \\192.168.121.150\public
```

- Windows maşından bütün workgroup-lara baxırıq.

- Windows maşından paylaşılmış bütün qovluqları çap edir.

- **192.168.121.150** IP ünvanlı və **public** adlı paylaşımı olan FreeBSD Samba server, windows öz keşinə istifadə üçün əlavə edir.

```
net use \\192.168.121.150\public /delete
```

- Windows öz keşindən samba serverin ünvanını silir.

```
net use \\10.0.0.10\share /user:cavid
```

- Windows maşınımız **192.168.121.150** IP ünvanlı samba serverə "**cavid**" istifadəçi adı ilə paylaşılmış qovluğa daxil olacaq.

```
net use z: /delete
```

- Windows maşınımızda olan "z" adlı şəbəkə diskini silirik.

```
start\\192.168.121.150\share
```

- Windows serverimiz əgər **192.168.121.150** IP ünvanlı samba serverdə qeydiyyatdan keçibse, share adlı qovluğu açacaq.

```
nbtstat -RR
```

- Serverdə olan netbios adın statusunu yeniləyir.

# Samba server Active Directory Authentication

Aşağıdakı əməliyyatlar FreeBSD 10.1 x64 üzərində test edilmişdir.

StartUP quraşdırma tam şəkildə sondakı nəticəni nümayiş etdirir. Ancaq StartUP faylinizi öncədən də bu formaya gətirsəniz, heç nəyə mane olmayıacaq. Sadəcə IP ünvanları özünüzə uyğun olaraq dəyişməyi unutmayın.

**DC: FREEBSD.LAN**

**DC IP: 192.168.121.150**

(UNIX maşın və bütün istifadəçilər üçün DNS servers-də təyin ediləcək IP ünvan)

**DC Computer Name: FREEBSD**

**SAMBA NETBIOS: unixnetbiosname** (İstifadə etməsəniz də olar.)

**SAMBA WorkGroup NAME: FREEBSD** (Bu, mütləq addır, çünki biz SAMBA-ya deyirik ki, Windows workgroup-un üzvü olsun.)

**Domain Admin User: Administrator**

FreeBSD serverimizin **/etc/resolv.conf** faylinə Windows IP ünvanını əlavə edirik.

**domain freebsd.lan**

**nameserver 192.168.121.150**

- DNS server olaraq Active Directory-nin DNS-ni istifadə edirik.

**ntpdate 192.168.121.150**

- FreeBSD sistem vaxtını Windows maşınla sinxronizasiya edirik.

**Qeyd:** Əgər bunu etməyi unutsanız, FreeBSD maşını Domain Controller-ə qoşula bilməyəcək və vaxt fərqi haqda səhvi çap edəcək.

```
/etc/rc.conf StartUP faylimiz aşağıdakı kimi olacaq:  
ifconfig_em0="inet 192.168.121.139 netmask 255.255.255.0"  
defaultrouter="192.168.121.2"  
hostname="samba.freebsd.lan"  
sshd_enable="YES"  
  
#### SendMail-i tam söndürürük.  
sendmail_enable="NO"  
sendmail_submit_enable="NO"  
sendmail_outbound_enable="NO"  
sendmail_msp_queue_enable="NO"  
sendmail_rebuild_aliases="NO"  
  
# Security Rules # SYN, FIN hücumlarının qarşısını alırıq.  
tcp_drop_synfin="YES"  
  
# Samba Portlardan yüklandıkdən sonra aktivləşdirilir.  
samba_enable="YES"  
winbindd_enable="YES"  
  
# Kerberos susmaya görə sistemdə olur. Sadəcə onu aktivləşdiririk.  
kdc_enable="YES"  
inetd_enable="YES"  
  
# Syslog-u yalnız serverin daxilində işləməyə məcbur edirik.  
syslogd_enable="YES"  
syslogd_program="/usr/sbin/syslogd"  
syslogd_flags="-ss"  
  
/etc/nsswitch.conf - Aşağıdakı sətirləri adın çevriləməsi faylinə yazırıq.  
group: files winbind  
passwd: files winbind  
shadow: files winbind  
group_compat: nis
```

```
hosts: files dns
networks: files
passwd_compat: nis
shells: files
services: compat
services_compat: nis
protocols: files
rpc: files
```

```
/etc/sysctl.conf
kern.maxfiles=25600
kern.maxfilesperproc=16384
net.inet.tcp.sendspace=65536
net.inet.tcp.recvspace=65536
```

- System Controla lazımı dəyişənləri əlavə edirik.

FreeBSD serverimizə Sambanın yüklenməsini **FreeBSD Samba** başlığından oxuyub yüklemək lazımdır. Bu səbəbdən yüklenməsi ardıcılığı bu başlıqda yazılmamışdır.

**/etc/krb5.conf** faylında KERBEROS quraşdırırıq ki, serverimizi Domain Controller-də qeydiyyatdan keçirək ki, istifadəçi yoxlanışını orada edək.

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log
[libdefaults]
default_realm = FREEBSD.LAN
    ticket_lifetime = 24h
    forwardable = yes
```

- Jurnalamanı aktivləşdiririk.
- Kitabxana jurnal faylini təyin edirik.
- KDC jurnalları faylini təyin edirik.
- Kadmin jurnal faylini təyin edirik.
- Domain Controller-imizin adı. Mütləq böyük hərflərlə olmalıdır.
- Biletin istifadə müddəti
- Yönləndirmə imkanı olsun.

```
[appdefaults]
pam = {
debug = false
    ticket_lifetime = 36000
    renew_lifetime = 36000
forwardable = true
    krb4_convert = false
}
```

Samba üçün `/usr/local/etc/smb.conf` quraşdırma faylinı aşağıdakı kimi edirik:

```
[global]
workgroup      = FreeBSD           - SAMBA serverimizin qrup adı.
server string   = Samba Server Version %v - SAMBA serverimizin açıqlanması.
security= ADS    - Şifrə siyaseti Active Directory serverdə olacaq.
realm= FREEBSD.LAN - KERBEROS adı, Domain Controller-in adı.
password server = 192.168.121.150 - Domain Controller-imizin IP ünvanını yazırıq.
netbios name = unixnetbiosname - SAMBA serverimizin NetBIOS adı  
(Yazmasanız da olar).

domain master = no - SMBD-yə deyir ki, WAN tərəfdən girişin  
siyahısını tutma.

local master = no - NMDB təyin edir ki, lokal master browser deyil  
ve ona qoşulan müraciətləri sindirib, DC-yə  
yollayır.

preferred master = no - Bu parametr təyin edir ki, NMDB öz WorkGroup-u  
üçün əsasdırımı, ya yox. (Detallar 'man smb.conf')

interfaces = em0 - Yalnız em0 şəbəkə kartında qulaq asın.  
(em0 em1 - ardıcılılığı ilə bir neçə şəbəkə kartı  
yaza bilərsiniz)

bind interfaces only = yes - Yalnız təyin etdiyimiz şəbəkə kartları qulaq  
asacaq.

idmap config * : range = 10000 - 40000 - Serverimizin DC ID aralığı.
idmap config * : backend = tdb - Burada SAMBA-ya deyirik ki, AD ilə əlaqə  
saxlayanda hansı alqoritmik xəritələnmə  
quruluşunu istifadə etsin. tdb NT Security  
identifiers-lə danışın uid/gid quruluşunu istifadə  
edir.

winbind enum groups = yes - Windows qruplarını rəqəmləyirik.
winbind enum users = yes - Windows istifadəçilərini rəqəmləyirik.
winbind use default domain = yes - Susmaya görə bu domain controller istifadə  
edilsin, istifadəçiləri domain-siz yazsaq, girəcək.

winbind refresh tickets = yes - Bu parametr Winbind-in verdiyi bilet üçün  
yoxlanışı yeniləyir.

template homedir = /home/%D/%U - Şablon shell kimi 'sh' istifadə edirik.
template shell = /bin/sh - Burada SAMBA təyin edir ki, həqiqətən,  
'smbclient' autentifikasiya üçün keçici
```

```

client ntlmv2 auth = yes

encrypt passwords = yes
restrict anonymous = 2

log level = 10
log file = /var/log/samba/%m.%U.log

max log size = 50000

[data]
comment = Windows-Share
path = /shares/data
read only = No
valid users = @"FREEBSD+Domain Users"

force group = "Domain Users"

directory mode = 0770

force directory mode = 0770

create mode = 0660

force create mode = 0660

access based share enum = yes

hide unreadable = yes

```

- serverdən istifadə edir.
- Burada SAMBA təyin edir ki, '**smbclient**' autentifikasiya üçün yönənləmiş serverdən **NTLMv2** şifrələnmiş autentifikasiya istifadə edir.
  - Parolları şifrələyək.
  - Anonim qoşulanlar üçün istifadəçi və qrup siyahısı məlumatının görünmə izni.
  - Jurnal səviyyəsi.
  - Qoşulan hər bir istifadəçiye görə jurnal faylı yaranacaq.
  - Maksimum jurnal faylı həcmi **50000KB** olacaq.
  - '**data**' adlı paylaşım yaradırıq.
  - Paylaşım şəhri yazırıq.
  - Paylaşılmış qovluğunun ünvani.
  - Yalnız oxuma hüququnu bağlayırıq.
  - İcazəsi olan istifadəçilər FreeBSD server və '**Domain Users**' qrupunun üzvləridir.
  - Sətirlə deyilir ki, servisimizdən istifadə edən hər kəs susmaya görə '**Domain-Users**' qrupunun üzvü olmalıdır.
  - DOS fayl sistemdən UNIX fayl sistemə qovluqlar üçün 10-luq say sistemində olan yetkilərin konvert olunmasında istifadə edilir. **770**-i UNIX-ə və Windows-a başa salır.
  - Samba serverdə yaradılan hər bir qovluğa susmaya görə sərt olaraq '**0770**' təyin edir.
  - DOS fayl sistemdən UNIX fayl sistemə fayllar üçün 10-luq say sistemində olan yetkilərin konvert olunmasında istifadə edilir.
  - UNIX fayl sistemdə yaradılacaq faylların sərt olaraq yetkisi maksimum **660** ola bilər.
  - Bu parametr təyin edir ki, ümumiyyətlə, servisimizə yalnız oxuma və yazma yetkisi olan istifadəçilər paylaşımı görə bilər.
  - Windows istifadəçilərinin oxuya bilməyəcəkləri faylları onlara göstərməyəcək.

```
mkdir-p /shares/data
```

- Paylaşım için lazım olan qovluğu yaradırıq.

```
/usr/local/etc/rc.d/samba start
```

- Samba serverimizi işe salırıq.

```
kinit Administrator
```

- Administrator istifadəçi adı ilə susmaya görə olan '**FREEBSD.LAN**' DC-mizə daxil oluruq.

```
Administrator@FREEBSD.LAN's Password:Administrator_Shifresi
```

```
klist
```

- Bu əmrlə '**FREEBSD.LAN**' domain-indən aldığımiz biletə və bitmə vaxtına baxırıq.

```
Credentials cache: FILE:/tmp/krb5cc_0
```

```
Principal: Administrator@FREEBSD.LAN
```

Issued	Expires	Principal
Apr 12 01:27:28 2015	Apr 12 11:27:28 2015	krbtgt/FREEBSD.LAN@FREEBSD.LAN

```
net join -U Administrator
```

- '**Administrator**' istifadəçi adı ilə DC-mizə üzv oluruq. Nəticə aşağıdakı sətirlərlə olmalıdır.

```
Enter Administrator's password:Administrator_shifresi
```

```
Using short domain name -- FREEBSD
```

```
Joined 'UNIXNETBIOSNAME' to dns domain 'freebsd.lan'
```

- Təyin etdiyimiz '**UNIXNETBIOSNAME**' NetBIOS adı ilə daxil oluruq.

```
net ads join -U Administrator
```

- Öncəki əmrlə eyni işigörür.

```
net ads testjoin
```

- Active Directory-ə qoşulmayı test edirik.

```
Join is OK
```

```
getent passwd
```

- FreeBSD istifadəçi siyahısı çap ediləcək və sonda Domain Controller istifadəçi siyahısı **10000+** aralığında siyahiya alınmalıdır.

```
getent group
```

- FreeBSD qrup siyahısı çap ediləcək və sonda Domain Controller qrupları **10000+** aralığında çap ediləcək.

**id Administrator** - DC-nin Administrator adlı istifadəçisinin **UID** və **GID**-nə baxırıq.

```
uid=10000(administrator) gid=10000(domain users) groups=10000(domain users),  
10012(denied rodc password replication group),10006(enterprise admins),  
10005(schema admins),10008(domain admins),10009(group policy creator owners)
```

**wbinfo -t** - Winbind-in işləməsini test edirik.  
checking the trust secret for domain FREEBSD via RPC calls succeeded

**wbinfo -g** - Domain qruplarına baxırıq.

```
winrmremotewmiusers__  
domain computers  
domain controllers  
schema admins  
enterprise admins  
cert publishers  
domain admins  
domain users  
domain guests  
group policy creator owners  
ras and ias servers  
allowed rodc password replication group  
denied rodc password replication group  
read-only domain controllers  
enterprise read-only domain controllers  
cloneable domain controllers  
protected users  
dnsadmins  
dnssupdateproxy
```

**wbinfo -u** - Domain istifadəçilərinə baxırıq.

```
UNIXNETBIOSNAME\root  
administrator  
guest  
cavid  
krbtgt
```

**kdestroy**

- Bütün alınan biletleri silir. (Silinərsə, yenidən **kinit -p administrator** əmrinə ehtiyac olacaq).

**chown -R Administrator:"Domain Users" /shares/data**

- SHARE qovluğumuzu Administrator istifadəçisi və 'Domain Users' qrupunun üzvü edirik ki, domain istifadəçiləri daxil ola bilsinlər.

**chmod 0770 /shares/data**

- Paylaşılmış qovluğumuza susmaya görə olan təyin etdiyimiz yetkiləri veririk.

**testparm**

- Əmrlə samba quraşdırılmalarımızı yoxlanış edirik. Bu əmr smb.conf faylında olan sintaksis səhvərini və yanlışlığı sizə göstərəcək. (Çox yararlı əmrdir.)

Samba serverə windows maşınınından paylaşılmış qovluğu istifadəyə vermək üçün aşağıdakı quraşdırımları etməliyik.

**/etc/nsmb.conf** faylına domain controller hüquqları quraşdırımlarını edirik.

**[default]**

**workgroup=FREEBSD.LAN**

- DC-mizi təyin edirik. Mütləq böyük hərflərə olmalıdır.

**[W2K12]**

- DC Computer Name. Mütləq böyük hərflərə olmalıdır.

**addr=192.168.121.150**

- DC sevrerin IP ünvani.

**[W2K12:ADMINISTRATOR]**

- DC AdminAccount: Admin İstifadəçi

**password=\$\$156404f27272d5844b0**

- DC ADMINISTRATOR Şifrəsi. Şifrəni **smbutil encrypt** əmri ilə şifrələyib buraya qeyd etmək lazımdır.

**/etc/fstab** StartUP faylımiza Windows serverdən paylaşılmış ünvanını əlavə edirik. **Administrator** istifadəçi adı ilə 'W2K12' serverin '**papka**' qovluğununu və 'C' diskini FreeBSD serverdə olan '**winserver**', '**mnt**' adlı qovluqlara mount edirik.

**//Administrator@w2k12/papka /winserver smbfs rw 0 0**

**//Administrator@w2k12/C\$ /mnt smbfs rw 0 0**

Sonda sistemə yenidənyüklənmə əmri daxil edib yoxlayırıq ki, disklerin avtomatik mount edilməsinin uğurlu nəticəsini görə bilək.

# **badsect - badblock (korlanmış disk blokları)-in fayllara köçürülməsi üçün program.**

**Badsect**-programdır, hansı ki, defect sektorlu fayl yaradır(daha doğru desək, o, deffektiv sektorların hər birini ayrıca fayla yerləşdirir). Bu, o halda lazımlı olur ki, əgər disk dağılırsa, onu həmin anda da dəyişdirmək mümkün deyil və hələ ki işləməlidir. Ümumi olan haldır.

Demək, programın cəmi iki parametri var - qovluq (harada ki, bad-blocklu fayllar yerləşdirilməlidir) və dağılmış sektorun rəqəmi.

```
root@:~ # df -h
Filesystem      Size   Used  Avail Capacity Mounted on
/dev/da0s1a     963M   572M   313M    65%   /
devfs          1.0k   1.0k    0B   100%   /dev
/dev/da0s1f     10G    3.7G   5.6G    40%   /usr
/dev/da0s1d     3.7G   1.1M   3.4G    0%    /var
/dev/dal        19G    577M   17G     3%    /home
/dev/da0s1e     944M   24k   868M    0%    /tmp
```

```
root@:~ # mkdir /tmp/bad-sect
root@: # badsect /tmp/bad-sect 104100
Don't forget to run ``fsck /dev/da0sle''
root@: # badsect /tmp/bad-sect 104101
Don't forget to run ``fsck /dev/da0sle''
root@: # badsect /tmp/bad-sect 104102
Don't forget to run ``fsck /dev/da0sle''
root@: # badsect /tmp/bad-sect 104103
Don't forget to run ``fsck /dev/da0sle''
root@: # badsect /tmp/bad-sect 104104
Don't forget to run ``fsck /dev/da0sle''
root@: # badsect /tmp/bad-sect 104105
Don't forget to run ``fsck /dev/da0sle''
root@:/ # ls -lah /tmp/bad-sect/
ls: 104100: Bad file descriptor
ls: 104101: Bad file descriptor
ls: 104102: Bad file descriptor
ls: 104103: Bad file descriptor
ls: 104104: Bad file descriptor
ls: 104105: Bad file descriptor
total 4
drwxr-xr-x  2 root  wheel   512B Feb 25 22:25 .
drwxrwxrwt 11 root  wheel   512B Feb 25 22:24 ..
root@:/ # umount -f /tmp/
root@:/ # fsck /dev/da0s1f
** /dev/da0sle
** Last Mounted on /tmp
** Phase 1 - Check Blocks and Sizes

HOLD BAD BLOCK? [yn] y

INCORRECT BLOCK COUNT I=94208 (0 should be 4)
CORRECT? [yn] y

HOLD BAD BLOCK? [yn] y
```

26025 DUP I=94211  
UNEXPECTED SOFT UPDATE INCONSISTENCY

INCORRECT BLOCK COUNT I=94211 (0 should be 4)  
CORRECT? [yn] y

HOLD BAD BLOCK? [yn] y

26025 DUP I=94212  
UNEXPECTED SOFT UPDATE INCONSISTENCY

INCORRECT BLOCK COUNT I=94212 (0 should be 4)  
CORRECT? [yn] y

HOLD BAD BLOCK? [yn] y

26025 DUP I=94213  
UNEXPECTED SOFT UPDATE INCONSISTENCY

INCORRECT BLOCK COUNT I=94213 (0 should be 4)  
CORRECT? [yn] y

HOLD BAD BLOCK? [yn] y

INCORRECT BLOCK COUNT I=94214 (0 should be 4)  
CORRECT? [yn] y

HOLD BAD BLOCK? [yn] y

26026 DUP I=94215  
UNEXPECTED SOFT UPDATE INCONSISTENCY

```
INCORRECT BLOCK COUNT I=94215 (0 should be 4)
CORRECT? [yn] y
```

```
INTERNAL ERROR: dups with -p
UNEXPECTED SOFT UPDATE INCONSISTENCY
** Phase 1b - Rescan For More DUPS
26025 DUP I=94208
UNEXPECTED SOFT UPDATE INCONSISTENCY
```

```
26026 DUP I=94214
UNEXPECTED SOFT UPDATE INCONSISTENCY
```

```
** Phase 2 - Check Pathnames
DUP/BAD I=94208 OWNER=root MODE=100600
SIZE=2048 MTIME=Feb 25 22:33 2014
FILE=/bad-sect/104100
```

```
UNEXPECTED SOFT UPDATE INCONSISTENCY
```

```
REMOVE? [yn] n
```

```
DUP/BAD I=94211 OWNER=root MODE=100600
SIZE=2048 MTIME=Feb 25 22:33 2014
FILE=/bad-sect/104101
```

```
UNEXPECTED SOFT UPDATE INCONSISTENCY
```

```
REMOVE? [yn] n
```

```
DUP/BAD I=94212 OWNER=root MODE=100600
SIZE=2048 MTIME=Feb 25 22:33 2014
FILE=/bad-sect/104102
```

```
UNEXPECTED SOFT UPDATE INCONSISTENCY
```

```
REMOVE? [yn] n
```

```
DUP/BAD I=94213 OWNER=root MODE=100600  
SIZE=2048 MTIME=Feb 25 22:33 2014  
FILE=/bad-sect/104103
```

UNEXPECTED SOFT UPDATE INCONSISTENCY

REMOVE? [yn] n

```
DUP/BAD I=94214 OWNER=root MODE=100600  
SIZE=2048 MTIME=Feb 25 22:33 2014  
FILE=/bad-sect/104104
```

UNEXPECTED SOFT UPDATE INCONSISTENCY

REMOVE? [yn] n

```
DUP/BAD I=94215 OWNER=root MODE=100600  
SIZE=2048 MTIME=Feb 25 22:33 2014  
FILE=/bad-sect/104105
```

UNEXPECTED SOFT UPDATE INCONSISTENCY

REMOVE? [yn] n

```
** Phase 3 - Check Connectivity  
** Phase 4 - Check Reference Counts  
** Phase 5 - Check Cyl groups  
FREE BLK COUNT(S) WRONG IN SUPERBLK  
SALVAGE? [yn] y
```

SUMMARY INFORMATION BAD

SALVAGE? [yn] y

```
ALLOCATED FRAGS 26025-26026 MARKED FREE  
BLK(S) MISSING IN BIT MAPS  
SALVAGE? [yn] y
```

```
16 files, 12 used, 483435 free (51 frags, 60423 blocks, 0.0% fragmentation)

***** FILE SYSTEM MARKED DIRTY *****

***** FILE SYSTEM WAS MODIFIED *****

***** PLEASE RERUN FSCK *****

root@:~ # mount /tmp
root@:~ # ls -lah /tmp/bad-sect/
total 16
drwxr-xr-x  2 root  wheel   512B Feb 25 22:33 .
drwxrwxrwt 11 root  wheel   512B Feb 25 22:32 ..
-rw-----  1 root  wheel    2.0k Feb 25 22:33 104100
-rw-----  1 root  wheel    2.0k Feb 25 22:33 104101
-rw-----  1 root  wheel    2.0k Feb 25 22:33 104102
-rw-----  1 root  wheel    2.0k Feb 25 22:33 104103
-rw-----  1 root  wheel    2.0k Feb 25 22:33 104104
-rw-----  1 root  wheel    2.0k Feb 25 22:33 104105
```

Bununla da bitdi.

Yeri gəlmışkən - testlərinizi real mاشında etməyin, ya da test etmək üçün block-ların rəqəmini ehtiyatla seçin - **block 104449**, **/var** slice-ında idi. Ən yaxşısı test etməzdən önce sistemi dump edin.

# CLRI - clear an inode, NullFS

## CLRI

**clri** - "görünmeyen" faylların silinmesi için utilit-dir(görünməmək - bu, fayl sistemin səhvinin nəticəsidir). O, fayl deskriptorunun ilk **64** baytını null-larla doldurur - sonda isə faylin qalan bütün blockları itmiş kimi təyin edilir, ona görə ki, **clri**-dan sonra **fsck**-ni isə salmaq lazımdır.

Bu testi etmək üçün **/dev/dal** diskini **/mnt** qovluğuna mount etmişdim.

```
root@:/ # mount /dev/dal /mnt/
root@:/ # cd /mnt/
root@:/mnt # mkdir clri
root@:/mnt # cd clri
root@:/mnt/clri # touch {1,2,3}.txt
root@:/mnt/clri # ls -i
2166788 1.txt  2166789 2.txt  2166790 3.txt
root@:/mnt/clri # clri /dev/dal 2166788
clri: /dev/dal: Operation not permitted
root@:/mnt/clri # cd /
root@:/ # umount /mnt/
root@:/ # clri /dev/dal 2166788
```

```
clearing 2166788
root@:/ # clri /dev/dal 2166789
clearing 2166789
root@:/ # clri /dev/dal 2166790
clearing 2166790
root@:/ # fsck /dev/dal
** /dev/dal
** Last Mounted on /mnt
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
UNALLOCATED I=2166788 OWNER=root MODE=0
SIZE=0 MTIME=Jan 1 00:00 1970
NAME=/clri/1.txt
```

REMOVE? [yn] **y**

```
UNALLOCATED I=2166789 OWNER=root MODE=0
SIZE=0 MTIME=Jan 1 00:00 1970
NAME=/clri/2.txt
```

REMOVE? [yn] **y**

```
UNALLOCATED I=2166790 OWNER=root MODE=0
SIZE=0 MTIME=Jan 1 00:00 1970
NAME=/clri/3.txt
```

REMOVE? [yn] **y**

```
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
FREE BLK COUNT(S) WRONG IN SUPERBLK
SALVAGE? [yn] y
```

SUMMARY INFORMATION BAD

SALVAGE? [yn] **y**

```

BLK(S) MISSING IN BIT MAPS
SALVAGE? [yn] y

14 files, 18 used, 10154181 free (29 frags, 1269269 blocks, 0.0% fragmentation)

***** FILE SYSTEM IS CLEAN *****

***** FILE SYSTEM WAS MODIFIED *****

root@:/ # mount /dev/dal /mnt/
root@:/ # cd /mnt/clri/
root@:/mnt/clri # ls

```

Açıqlamaq lazımdır ki, **1.txt**, **2.txt**, **3.txt** - test üçündür, guya "silinmirlər". Inode rəqəminə baxmaq üçün isə **ls -i** əmrindən istifadə edin. Ya da üsul olaraq **man fsdb**

## NullFS

**mount\_nullfs** - bir qovluğun digər qovluğa mount edilməsi

**mount\_nullfs**-utilit-i **null** üzlük yaradır, hansı ki, mövcud olan fayl sistemin ağac quruluşunu digər adla fayl sistemdə yayımlayır. Bu, fayl və qovluqlara müxtəlif yollarla çatma imkanı yaradır. Xülasə, ☺ mənası odur ki, eyni maşın üzərində bir direktoriyanı digərinə mount edirik. Maraqlı üstünlükdür, eyni program portu və mənbə kodlarını maşınınımızda olan bir neçə **jail** üçün istifadə etməyimizə şərait yaradır. ([man 8 jail](#)).

Misal:

```

root@:/ # foreach i [1 2 3 4 5 6 7 8 9 0]
foreach? mkdir /mnt/$i
foreach? mount_nullfs /usr/ports /mnt/$i
foreach? end
root@:/ # df -h
Filesystem      Size   Used   Avail Capacity Mounted on
/dev/da0s1a     963M   572M   313M    65%     /
devfs          1.0k   1.0k     0B   100%     /dev
/dev/da0s1e     944M    24k   868M     0%     /tmp
/dev/da0s1f     10G    3.7G   5.6G    40%     /usr

```

/dev/da0s1d	3.7G	1.1M	3.4G	0%	/var
/dev/dal	19G	577M	17G	3%	/home
/usr/ports	10G	3.7G	5.6G	40%	/mnt/1
/usr/ports	10G	3.7G	5.6G	40%	/mnt/2
/usr/ports	10G	3.7G	5.6G	40%	/mnt/3
/usr/ports	10G	3.7G	5.6G	40%	/mnt/4
/usr/ports	10G	3.7G	5.6G	40%	/mnt/5
/usr/ports	10G	3.7G	5.6G	40%	/mnt/6
/usr/ports	10G	3.7G	5.6G	40%	/mnt/7
/usr/ports	10G	3.7G	5.6G	40%	/mnt/8
/usr/ports	10G	3.7G	5.6G	40%	/mnt/9
/usr/ports	10G	3.7G	5.6G	40%	/mnt/0

Ancaq nəzərə alın ki, NFS ilə nullfs birgə işləmir. 😊

# Sıradan çıxmış fayl sistemin geri qaytarılması

Elə hallar olur ki, server donmağa başlayır. Donma serveri tam qeyri-işlek vəziyyətə gətirib çıxara bilir. Məcburiyyət inzibatçını nəyinsə düzəlməsi üçün serveri **reset** etməyə gətirib çıxarır. Hətta səhv **INSERT SYSTEM HARD DISK AND PRESS ENTER** konsola çıxır. İndi isə həmin sərt diskin geri qaytarılmasına baxaq.

```
testbed# mount /dev/ad1s1a /mnt/
mount: /dev/ad1s1a : Operation not permitted
testbed# fsck /dev/ad1s1a
fsck: Could not determine filesystem type
```

Necə edək?!

Əslində serverdə slice və partitionların olması artıq işi rahatlaşdırır. Özümüzü sakitləşdirmək üçün **fdisk** və **bslabel** əmrlərinin nəticələrinə baxmaq olar:

```

testbed# fdisk /dev/ad1
***** Working on device /dev/ad1 *****
parameters extracted from in-core disklabel are:
cylinders=77542 heads=16 sectors/track=63 (1008 blks/cyl)
Figures below won't work with BIOS for partitions not in cyl 1
parameters to be used for BIOS calculations are:
cylinders=77542 heads=16 sectors/track=63 (1008 blks/cyl)

Media sector size is 512
Warning: BIOS sector numbering starts with sector 1
Information from DOS bootblock is:
The datafor partition 1 is:
sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
start63, size36852417 (17994 Meg), flag 80 (active)
    beg: cyl 256/ head 0/ sector 0;
end: cyl 1023/ head 7/ sector 7
The datafor partition 2 is:
sysid 7 (0x07),(NTFS, OS/2 HPFS, QNX-2 (16 bit) or Advanced UNIX)
start36853110, size41287050 (20159 Meg), flag 0
    beg: cyl 1023/ head 255/ sector 63;
end: cyl 1023/ head 15/ sector 63
The datafor partition 3 is:
<UNUSED>
The datafor partition 4 is:
<UNUSED>
testbed# bslabel /dev/ad1sl
# /dev/ad1sl:
8 partitions:
#      size      offset      fstype   [fsizze bsize bps/cpg]
a: 104857604.2BSD     2048163848
b: 41263521048576          swap
c: 368524170      unused      00# "raw" \
\ part, don't edit
d: 415948851749284.2BSD     20481638428552
e: 104857693344164.2BSD     2048163848
f: 26469425103829924.2BSD     20481638428552

```

Öncə disk hissəsinin işlək və rezerv nüsxəsini çıxarmaq lazımdır.

Fayl sistem partitionda nə isə qalmasından əmin olmadığımiza görə, gələcək işlərimiz üçün işlək nüsxə yaradaq.

```
testbed# dd if=/dev/adlsla of=/usr/adlsla.img
testbed# cp /usr/adlsla.img /usr/adlsla.img.orig
testbed# file /usr/adlsla.img
/usr/adlsla.img: Unix Fast File system [v2] (little-endian) last mounted on /,
last written at Tue May 3117:37:512011, clean flag 0, readonly flag 0,
number of blocks 262144, number of data blocks 253815, number of cylinder
groups 4, block size16384, fragment size2048, average filesize16384,
averagenumber of files in dir 64, pending blocks to free 0, pending inodes
to free 0, system-wide uuid 0, minimum percentage of free blocks 8, TIME
optimization
testbed# mdconfig -a -t vnode -f /usr/adlsla.img
md0
testbed# mount /dev/md0 /mnt/
mount: /dev/md0 : Operation not permitted
123456789_123456789_123456789_123456789-123456789-12465789-123456789-
123456789-
```

Gördüyüümüz kimi, disk hissəsi tam ölmüş vəziyyətdə deyil, onda həqiqətən də, UFS2(aka FFS) var, ancaq nə üçünsə mount olmaq istəmir. Əgər sizdə vəziyyət daha pisdirsə, ola bilər ki, fayl sistemin diskində qalan nə varsa çıxaracaqsınız. Bu halda size **/usr/ports/sysutils/scan\_ffs** kömək edə bilər.

Əsas mövcud olan səbəblərdən biri (təcrübədə hərdən rastlaşmaq olur) superblock-un zədələnməsidir. (**UFS2-nin 160-ci bloku**). Bu hallar üçün superblock-un rezerv nüsxəsi planlaşdırılıb, fayl sistemin yaradılması anında ünvanı çap edilməli və etibarlı yerdə saxlanılmalıdır. Əgər siz superblock-un nə olduğunu və fayl sistemin necə yükləndiyini bilmirsinizsə, onda bu haqda məlumatı internetdən təpib çox ciddi oxumaq məsləhətdir. Qısaca, superblock fayl sistemin bəzi quraşdırılmalarını və disk həndəsəsini açıqlayan, fayl sistemi inisializasiya edən sehrli rəqəm və digər vacib rəqəmlərdən ibarətdir. Ya da **man newfs(8)**

```
testbed# man newfs
NEWFS(8)          FreeBSD System Manager's Manual      NEWFS(8)

NAME
newfs -- construct a new UFS1/UFS2 file system

...
-N      Cause the file system parameters to be printed out without really
creating the file system.
^C
testbed# newfs -N /dev/md0
/dev/md0: 512.0MB (1048576 sectors) block size16384, fragment size2048
using 4 cylinder groups of 128.02MB, 8193 blks, 16448 inodes.
super-block backups [for fsck -b #] at:
160, 262336, 524512, 786688
```

Gördüğümüz kimi, hissə hər birində superblock-un nüsxəsi olan 4 "qrup silindrə" ayrılmışdır.

Alternativ superblock-u istifadə edərək diskı yoxlamağa çalışaq.

```
testbed# man fsck_ffs
FSCK_FFS(8)          FreeBSD System Manager's Manual      FSCK_FFS(8)

NAME
fsck_ffs, fsck_ufs -- file system consistency check and interactive
repair

SYNOPSIS
fsck_ffs [-BFprfny] [-b block] [-c level] [-m mode] filesystem ...

...
-b      Use the block specified immediately after the flag as the super
blockfor the file system. An alternate super block is usually
```

```
located at block 32for UFS1, and block 160for UFS2.  
^C  
testbed# fsck_ffs -b 262336 /dev/md0  
Alternate super block location: 262336  
** /dev/md0  
** Last Mounted on  
** Phase 1 - Check Blocks and Sizes  
** Phase 2 - Check Pathnames  
** Phase 3 - Check Connectivity  
** Phase 4 - Check Reference Counts  
** Phase 5 - Check Cyl groups  
SUMMARY BLK COUNT(S) WRONG IN SUPERBLK  
SALVAGE? [yn] y  
  
3430 files, 117564 used, 136251 free (771 frags, 16935 blocks,  
0.3% fragmentation)  
  
UPDATE STANDARD SUPERBLOCK? [yn] y  
  
***** FILE SYSTEM IS CLEAN *****  
  
***** FILE SYSTEM WAS MODIFIED *****  
testbed# mount /dev/md0 /mnt/
```

Əla! Düzəldi!☺

Artıq bizdə fayl sistemin işlək nüsxəsi mövcuddur - **/usr/adls1a.img**

Əgər tələb edilərsə, analoji işləri digər hissələr üçün də edirik və disklərdə olan səhvləri diqqətlə analiz edirik. Tələb olarsa, diskı dəyişirik və onu elə böyük ki, hər bir yeni böldüyüümüz disk hissəsində köhnələr yerləşə bilsin, bundan sonra köhnə hissələrdə olan verilənləri yeni yaratdığımıza köçürürük. Bunu **tar**, **pax**, ya da **dump/restore** ilə edə bilərik.

## clonehdd - sərt diskin hissəsinin digərinə nüsxələnməsi

Sərt disk hissələrinin digərinə nüsxələnməsi üçün alət. FreeBSD “partition” kimi göstərilən hissələr haqqında öz verilənlərini saxlayır. **Partition Magic** və ya **Acronis** tipli programlar ancaq **FreeBSD** slice-ni təyin edir. (Lazım olan isə slice-da olan hissələrin real həcmidir.) **CloneHDD** isə sistemdə olan utilit-lərin ön tərəfində işləmək üçün istifadə edilən alətdir (dəqiq desək, **dump/restore** üçün).

Ona görə də bu alət fayl sistemi səviyyəsində yox, **hissə (slice)** səviyyəsində işləyir və uyğun olaraq böyük imkan yaradır ki, HDD-nizi rahat köçürü və rezerv nüsxə edə biləsiniz. Yəni kiçik həcmli olan HDD-dən böyüyə və böyük həcmli olan HDD-dən kiçiyə informasiyanın köçürülməsi imkanı mövcuddur. Sadəcə həmin diskdə köçürüləcək ümumi informasiya həcmi qədər yer olmalıdır. Program çox sadədir, yükləyək:

```
root@:/ # cd `whereis clonehdd | awk '{ print $2}'`  
root@:/usr/ports/sysutils/clonehdd # make install
```

Sonra **da0** diskimizi **da1**-ə nüsxələyək.

```
clonehdd -src=da0 -dst=da1 -swap=512
```

Susmaya görə tam **yes/no ?** soruşur. Əgər bu, sizə mane olarsa, ya da siz **clonehdd**-ni cron-a yerləşdirməyi planlaşdırırsınızsa, onda aşağıdakı opsiya sizin köməyinizə çatacaq.

```
-force
```

Bundan sonra heç bir sual verilməyəcək. Sonra aşağıda göstərildiyi kimi, nəticə çıxacaq, ya da **-force** əmri ilə cron-la işə salmışınızsa, onda yönləndirdiyiniz faylin çıxışına baxa bilərsiniz.

Clone parameters:

```
Source partition: /dev/da0  
Dest partition: /dev/da1  
Swap size: 512 MB  
Safe dumping: Disabled  
Free space on DST: 100 MB  
Fstab device name: da0  
---  
[OK] Found devices for clone procedure  
[OK] DST partitions are not in use  
---  
Source partition  
/usr size: 10322MB, used: 2797MB  
/var size: 3788MB, used: 1MB
```

```
/ size: 963MB, used: 572MB
/tmp size: 944MB, used: MB
Total: 16018 MB, used: 3370 MB
---
[OK] Device dal has enough free space
DATA ON DEVICE dal WILL BE DESTROYED NOW!
Continue? [yes/no]:yes
Wait 10 seconds before start: 10 9 8 7 6 5 4 3 2 1
[OK] Device /dev/dal made clean
[OK] New slice created
---
Destination device partitions:
SWAP size: 512 MB
/ size 1200 MB
/tmp size 1176 MB
/var size 4721 MB
/usr size 12863 MB
---
[INF] Last partition were increased for 1 blocks
[OK] Partitions were created successfully
---

[OK] Partition /tmp was formatted successfully
Starting dump/restore procedure...      [OK]

[OK] Partition /var was formatted successfully
Starting dump/restore procedure...      [OK]

[OK] Partition /usr was formatted successfully
Starting dump/restore procedure...      [OK]

[OK] Partition / was formatted successfully
Starting dump/restore procedure...
[WARN] Partition / moving in unsafe mode!      [OK]
[OK] file /etc/fstab generated successfully
```

Artıq sizin **da0** diskinin tam nüsxəsi olan **da1** diskiniz mövcuddur.

# BÖLÜM 12

## SFTP server Chroot, Tomcat8, MPD5 StS VPN, MPD5 PPTP RA VPN, NTOP netflow

- / OpenSSH SFTP server CHroot directory
- / FreeBSD 10.1 x64 Tomcat8
- / FreeBSD MPD5 Site-to-Site VPN
- / FreeBSD MPD5 PPTP (Remote Access VPN)
- / FreeBSD NTOP (NetFlow Trafikin monitoring edilməsi)

Adətən istənilən şirkətdə sistem inzibatçından FTP server tələb edilir, ancaq FTP serverdə istifadəçi ilə server arasında olan yol şifrələnmir. Elə məqam ola bilər ki, VPN server qaldırmaq imkanı yoxdur və qısa bir zamanda istifadəçiye şifrələnmiş kanal vasitəsilə data ötürmək imkanı tələb edilir. Bu halda sizin köməyinizə OpenSSH üzərində olan SFTP çatacaq və başlığımızda onun qurulması göstərilir.

Əgər web programçı öz kodunu Java-da yazarsa, sizə Java-nın işləməsi üçün web server tələb ediləcək. Bu başlıqla java programlarının işləməsi üçün Tomcat8 web serveri qurulacaq və işləməsi sınaqdan keçiriləcək. Əksər hallarda şirkətlərin tələbi yaranır ki, digər şirkətlərin daxili şəbəkəsini təhlükəsiz kanal vasitəsilə görə bilsin. Adətən bu hallarda IPSec istifadə edilir, ancaq bu başlığımızda eyni işi PPTP vasitəsilə görəcəyik. İşin görülməsi üçün MPD5 program təminatından istifadə ediləcək. Əgər şirkətinizin daxili şöbəsində olan resurslara qeyri-iş zamanları və hansısa iş ezamiyəti vaxtında yetki almaq istəsəniz, RA VPN tələb ediləcək. Bunun situasiya tələbi üçün MD5 PPTP ilə RA rejimində qurulacaq. Əgər şirkətinizin trafikinin istənilən istiqamətə gedişətini monitoring etmək istəsəniz, sizə Netflow tələb ediləcək. Bu başlıqla onun qrafik rejimdə nəticə əldə etməklə qurulması açıqlanır.

# **OpenSSH SFTP server CHroot directory**

Sizdən tələb oluna bilər ki, hansısa uzaqda yerləşən istifadəcidən şifrələnmiş kanal vasitəsilə məlumat alasınız. Bu halda VPN qurmalarınız və istifadəçiə uyğun VPN profile-la lazımi program təminatı yükləyib quraşdırılmalıdır. İstənilən halda VPN quraşdırılması müəyyən vaxt alacaq. Bunun ən qısa yolu SFTP-dir. Ancaq SFTP quraşdırıldıqda həmin istifadəçi SSH yetkiyə də sahib olacaq. Bu başlığımız məhz istifadəçinin yalnız SFTP-dən istifadə etməsindən dənmişir, ancaq istifadəçi ev qovluğunu sərhədini aşmaq istədikdə kök qovluğundan daha da yuxarı səviyyəyə qalxa bilməyəcək.

Başlığımızda **FreeBSD 10.1 x64**, **OpenSSH\_6.2p2**, **OpenSSL 0.9.8y** istifadə edilir.

**/etc/ssh/sshd\_config** quraşdırma faylında qulaq aslığı portu dəyişirik və sonuna aşağıdakı sətirləri əlavə edirik:

**Port 20000**

- Tehlükəsizlik üçün portunu dəyişirik.

**AllowUsers cavid faxri**

- Bu iki təyin edilən istifadəcidən başqa istifadəçi sistemə daxil ola bilməz.

**Match User cavid**

- Əgər bu istifadəçi olsa, aşağıdakı opsiyaları ona mənimsədəcək.

**X11Forwarding no**

- X-i forward edə bilməsin.

**AllowTcpForwarding no**

- TCP forward edə bilməsin.

**AllowAgentForwarding no**

- Agent forward edə bilməsin.

**PermitTunnel no**

- Tunelə izin vermirik.

```
ForceCommand internal-sftp  
ChrootDirectory /home/cavid
```

- Ancaq SFTP istifadə edə bilsin, SSH bağlıdır.
- Maksimal qalxa biləciyi qovluq. Bu qovluğun hüquqları root istifadəçi və wheel qrupu olmalıdır.

```
Match User faxri  
    X11Forwarding no  
    AllowTcpForwarding no  
    AllowAgentForwarding no  
    PermitTunnel no  
    ForceCommand internal-sftp  
    ChrootDirectory /home/faxri
```

```
echo 'sshd_enable="YES"' >> /etc/rc.conf
```

- SSHD daemon-u sistem StartUP faylına əlavə edirik ki, sistem yenidənyüklənməsindən sonra avtomatik işə düşsün.

```
/etc/rc.d/sshd restart
```

- SSH daemon-u yenidən işə salırıq ki, dəyişikliklərimiz mənimədilsin.

```
mkdir -p /home/cavid/cavid
```

- Yaradacağımız istifadəçi üçün öncədən CHRoot qovluğun altında ev qovluğu yaradırıq.

**/home/cavid** CHroot qovluğu mütləq **root:wheel** üzvü olmalıdır.

**/home/cavid/cavid** qovluğu isə, **cavid** adlı istifadəçinin ev qovluğuudur.

```
chown root:wheel /home/cavid
```

- **Cavid** istifadəçisi üçün CHroot qovluq bu olacaq, ona görə də qovluğu **root** istifadəçi və **wheel** qrupunun üzvü edirik.

İndi isə istifadəçini yaradırıq.

```
adduser      - Cavid adlı yeni istifadəçini sistemə əlavə edirik.
```

Username: **cavid**

Full name: **Cavid Bayramov**

Uid (Leave empty for default):

Login group [cavid]:

```

Login group is cavid. Invite cavid into other groups? []:
Login class [default]:
Shell (sh csh tcsh nologin) [sh]: csh
Home directory [/home/cavid]: /home/cavid/cavid
                                         - Ev qovluğuna diqqət yetirin. Ev qovluğu
                                         avtomatik cavid istifadəçi adı və qrupun yetkisini
                                         alacaq.

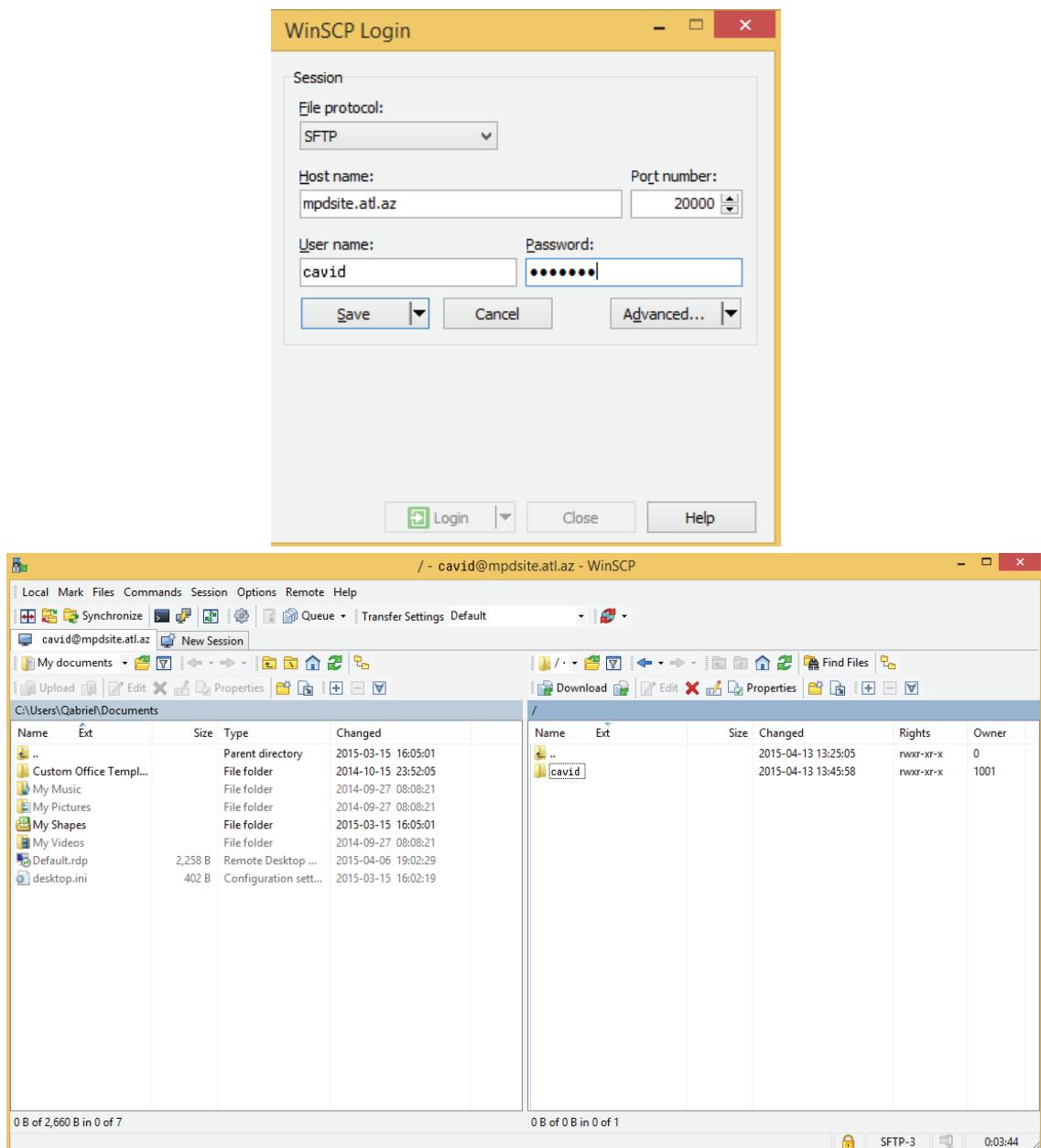
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username : cavid
Password : *****
Full Name : Cavid Bayramov
Uid       : 1001
Class     :
Groups    : cavid
Home      : /home/cavid/cavid
Home Mode :
Shell     : /bin/csh
Locked    : no
OK? (yes/no): yes
adduser: INFO: Successfully added (cavid) to the user database.
Add another user? (yes/no): no
Goodbye!

```

```
cd /home/cavid/cavid
rm .*
```

- Təhlükəsizlik üçün.  
- İstifadəçinin ev qovluğunda sistem  
profilərini silirik.

WinSCP ilə test etsəniz, görəcəksiniz ki, **cavid** adlı istifadəçi ilk olaraq CHroot qovluğuna düşür və öz adı altında olan digər bir qovluq görür. Onun ancaq öz adına olan qovluqda yazma və oxuma yetkisi olur.



Eyni ilə **faxrı** adlı istifadəçi üçün yaradıb test etsəniz, nəticə uyğun olacaq.

Beləliklə, hər bir yeni istifadəçi sistemə əlavə edildikdə eyni ardıcılılığı etmək kifayətdir. Və hər bir istifadəçi maksimum özündən bir ünvan öndə olan qovluğā qalxa biləcək.

# FreeBSD 10.1 x64

## Tomcat8

**Tomcat** (köhnə versiyalarda **Catalina**) – açıq kodlu servlet konteynerindən ibarətdir, hansı ki, Apache Software Foundation tərəfindən yaradılıb. Servlet-lər quruluşunu, JavaServer(JSP) quruluşunu və JavaServer Faces(JSF) quruluşunu həyata keçirir. Java dilində yazılib. Tomcat web programı işə salır. Tərkibində özünü quraşdırmaq üçün çoxlu program təminatı mövcuddur. Tomcat müstəsna web-server kimi istifadə edilir. Aşağıdakı komponentləri mövcuddur.

**Catalina** – Tomcat servlet-lərin konteyneridir. Catalina servlet-lərin və JavaServer Pages-in quruluşunu həyata keçirir.

**Coyote** – HTTP Tomcat stekinin komponentidir, hansı ki, web serverlər, ya da programlar konteyneri üçün HTTP1.1 protokolunu dəstəkləyir. Coyote daxil olan qoşulmalar üçün müəyyən təyin edilmiş TCP portda qulaq asır, müraciətləri emal edilmələri üçün Tomcat mexanizminə ötürür və cavabı yenidən müraciət edən istifadəçiye qaytarır.

**Jasper** – Tomcat-in JSP mexanizmidir. Jasper JSP faylları analiz edir ki, onları servlet kimi Java koda(hansı ki, Catalina vasitəsilə emal edilə bilər) kompilyasiya etsin. Yerinə yetirilmə müddətində Jasper avtomatik olaraq JSP faylda olan dəyişikliyi təyin edə və onu yenidən kompilyasiya edə bilər.

```
cd /usr/ports/devel/gmake
```

- Tomcat tərəfindən tələb olunduğu üçün Gmake-i yükleyirik.

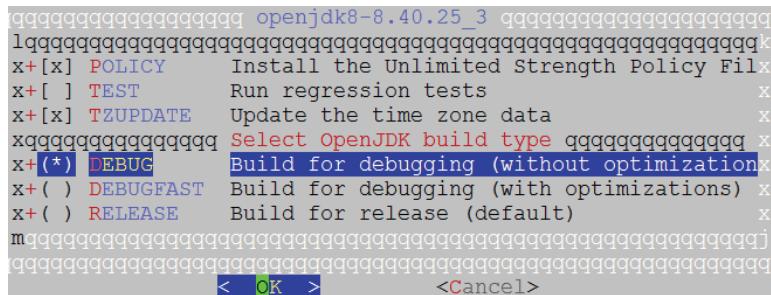
```
make install clean
```

```
cd /usr/ports/java/openjdk8
```

- Java-nı yükleyirik.

```
make config
```

- Lazımı opsiyaları seçirik.



```
make install
```

- Yüklenmə müddətində gələn paketlərin hamısını susmaya görə seçirik, yalnız IPv6-ni söndürürük.

Yüklenmə bitdikdən sonra aşağıdaki sətirləri **/etc/fstab** faylına əlavə edirik:

<b>fdesc</b>	<b>/dev/fd</b>	<b>fdescfs</b>	<b>rw</b>	<b>0</b>	<b>0</b>
<b>proc</b>	<b>/proc</b>	<b>procfs</b>	<b>rw</b>	<b>0</b>	<b>0</b>

Həmçinin CLI-dan aşağıdakı əmri işə salırıq ki, sistemi yenidənyüklənmə etmədən işə salaq.

```
mount -t fdescfs fdesc /dev/fd
mount -t procfs proc /proc
```

```
cd /usr/ports/www/tomcat8
```

- Tomcat8-i yüklemək üçün port ünvanına daxil oluruq.

```
make install
```

- Yükleyirik.

Siz Tomcat üçün lazım olan dəyişənlər siyahısına **/usr/local/apache-tomcat-8.0/bin/catalina.sh** faylında baxıb, **/usr/local/apache-tomcat-8.0/bin/setenv.sh** faylına əlavə edə bilərsiniz.

```
/usr/local/apache-tomcat-8.0/bin/setenv.sh faylında Tomcat-in qlobal dəyişənlərini elan edirik.
```

```
CATALINA_HOME=/usr/local/apache-tomcat-8.0/
JAVA_HOME=/usr/local/openjdk8/jre/
```

```
/usr/local/apache-tomcat-8.0/conf/tomcat-users.xml
```

- Faylı aşağıdakı kimi quraşdırırıq ki, admin panel-ə istifadəçi adı və şifrə ilə daxil ola bilək.

```
<?xml version='1.0' encoding='utf-8'?>
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
    version="1.0">
    <role rolename="manager-gui"/>
    <role rolename="manager-script"/>
    <role rolename="manager-jmx"/>
    <role rolename="manager-status"/>
    <role rolename="admin-gui"/>
    <role rolename="admin-script"/>
        <user username="admin" password="freebsd" roles="manager-gui,manager-
script,manager-jmx,manager-status,admin-gui,admin-script"/>
</tomcat-users>
```

**/usr/local/apache-tomcat-8.0/bin/startup.sh** - Skripti işə salırıq ki, Tomcat-ın qlobal dəyişənlərini elan edək.

```
Using CATALINA_BASE:      /usr/local/apache-tomcat-8.0
Using CATALINA_HOME:      /usr/local/apache-tomcat-8.0/
Using CATALINA_TMPDIR:   /usr/local/apache-tomcat-8.0/temp
Using JRE_HOME:          /usr/local/openjdk8/jre/
Using CLASSPATH:         /usr/local/apache-tomcat-8.0//bin/bootstrap.jar:/usr/local/
apache-tomcat-8.0/bin/tomcat-juli.jar
Tomcat started.
```

**ee /usr/local/apache-tomcat-8.0/conf/server.xml**

- Əsas quraşdırma faylına baxırıq ki, Tomcat-ın qulaq asdığı portu təyin edək.

**echo 'tomcat8\_enable="YES"' >> /etc/rc.conf** - Tomcat-ı StartUP-a əlavə edirik ki, reboot-dan sonra işləsin.

**/usr/local/etc/rc.d/tomcat8 start** - Tomcat-ı işə salırıq.

**netstat -na|grep 8080** - Tomcat portu susmaya görə **8080** olur.

Istənilən WEB browser-dən Tomcat serverin IP ünvanına daxil oluruq.

**http://server\_ip\_address:8080**

The screenshot shows the Apache Tomcat 8.0.18 homepage. At the top, there's a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. To the right of the navigation bar is a link to 'Find Help'. Below the navigation bar is the Apache Software Foundation logo. A green banner at the top of the main content area says 'If you're seeing this, you've successfully installed Tomcat. Congratulations!'. To the left of this banner is a cartoon cat icon. To the right are three buttons: 'Server Status', 'Manager App', and 'Host Manager'. The main content area is divided into several sections: 'Developer Quick Start' (with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications/Tomcat Versions), 'Managing Tomcat' (with links to Release Notes, Changelog, Migration Guide, Security Notices, and CATALINA\_HOME/conf/tomcat-users.xml), 'Documentation' (with links to Tomcat 8.0 Documentation, Tomcat 8.0 Configuration, Tomcat Wiki, and CATALINA\_HOME RUNNING.txt), and 'Getting Help' (with links to FAQ and Mailing Lists, including tomcat-announce, tomcat-users, taglibs-user, and tomcat-dev).

WEB vasitəsilə Tomcat serverimiz program idarə etmə interfeysi nə daxil oluruq və quraşdırduğumız istifadəçi adı ilə şifrəni daxil edirik:

The screenshot shows the same Apache Tomcat 8.0.18 homepage as above, but with an 'Authentication Required' dialog box overlaid. The dialog box has a question mark icon and asks for a username and password. It specifies that the site is "Tomcat Manager Application". The 'User Name:' field contains "admin" and the 'Password:' field contains "\*\*\*\*\*". There are "OK" and "Cancel" buttons at the bottom of the dialog box. The rest of the page content is visible behind the dialog, including the 'Developer Quick Start', 'Managing Tomcat', 'Documentation', and 'Getting Help' sections.

Son nəticədə alınan səhifə aşağıdakı kimi olacaq:

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a header with the Apache Software Foundation logo and a yellow cat icon. Below the header, the title "Tomcat Web Application Manager" is centered. A message box at the top left contains the text "Message: OK". The main area is titled "Manager" and includes links for "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The "Applications" section has a table with columns: Path, Version, Display Name, Running, Sessions, and Commands. The table lists several applications: "Welcome to Tomcat" (Path: /, Version: None specified, Running: true, Sessions: 0), "Tomcat Documentation" (Path: /docs, Version: None specified, Running: true, Sessions: 0), "Servlet and JSP Examples" (Path: /examples, Version: None specified, Running: true, Sessions: 0), "Tomcat Host Manager Application" (Path: /host-manager, Version: None specified, Running: true, Sessions: 0), and "Tomcat Manager Application" (Path: /manager, Version: None specified, Running: true, Sessions: 2). Each application row has a "Commands" button group containing "Start", "Stop", "Reload", and "Undeploy" buttons, along with an "Expire sessions" link.

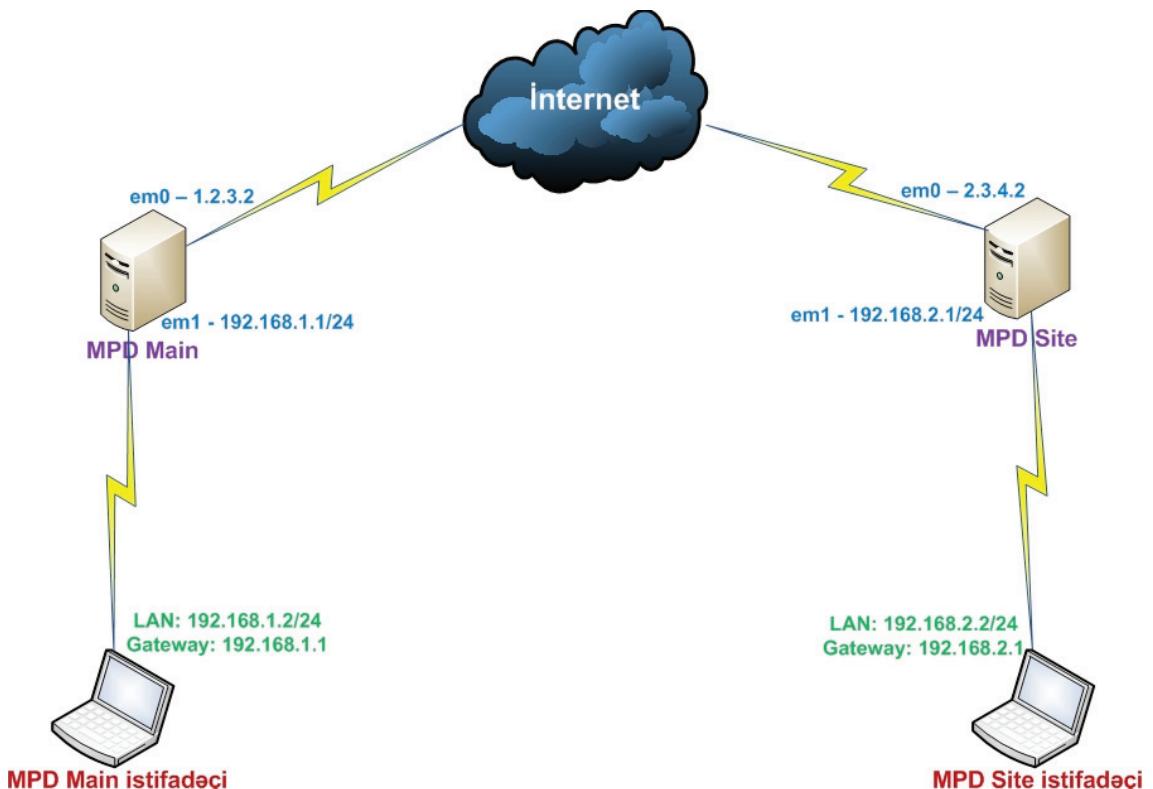
Test etmək üçün <https://tomcat.apache.org/tomcat-8.0-doc/appdev/sample/> ünvanından **sample.war** faylınlı endirib, serverimizin **/usr/local/apache-tomcat-8.0/webapps** ünvanına yükleyirik. Sonda bir daha hansısa WEB browser vasitəsilə müraciət edib, WAR nüsxəmizi test edirik. Nəticə aşağıdakı şəkildəki kimi olacaq:

The screenshot shows the home page of the "Sample 'Hello, World' Application". It features a yellow cat icon on the left and the title "Sample 'Hello, World' Application" in bold. Below the title, a short text states: "This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide." A note below says: "To prove that they work, you can execute either of the following links:" followed by two bullet points: "To a [JSP page](#)" and "To a [servlet](#)".

# FreeBSD MPD5 Site-to-Site VPN

İki ədəd FreeBSD maşınımız var. Hər birində də iki ədəd şəbəkə kartı var.

Məqsədimiz FreeBSD машınları arxasında olan daxili şəbəkələrin şifrələnmiş kanalla bir-birini görməsidir. Buna Site-to-Site VPN deyilir. Ancaq biz Site-to-Site VPN-i PPTP vasitəsilə quracaq. Topologiyamız aşağıdakı şəkildəki kimi olacaq:



```
portsnap fetch extract update
```

- Hər iki maşında portları yeniləyirik.

**Qeyd:** Mütləq nəzərə alın ki, serverləri qonşu PUBLIC aralığında istifadə etsəniz, VPN qurulmayıcaq. Tam ayrı PUBLIC şəbəkələrlə olmalıdır və aralarında routing işləməlidir.

Hər iki maşında da kernel-i lazımi opsiyalarla kompilyasiya edirik.

Əgər Firewall istifadə edəcəksinizsə, onda aşağıdakı opsiyalar sizə lazım olacaq. Əsas odur ki, lazımsız driver-lər və IPv6-ni söndürəsiz.

```
cd /sys/`uname -p`/conf
```

```
cp GENERIC kernel
```

- Kernel faylinin qovluğuna daxil oluruq.

- Susmaya görə olan kernel **GENERIC** faylini bize uyğun olan adda digər fayla nüsxələyirik.

Kernel faylımiza bize lazım olan opsiyaları nüsxələyirik.

```
# IPFW options
```

```
options      IPFIREWALL
options      IPFIREWALL_VERBOSE
options      IPFIREWALL_VERBOSE_LIMIT=10
options      IPDIVERT
options      DUMMYNET
options      IPFIREWALL_NAT
options      LIBALIAS
```

Hər iki maşında da MPD5-i yükləyirik.

```
whereis mpd5
```

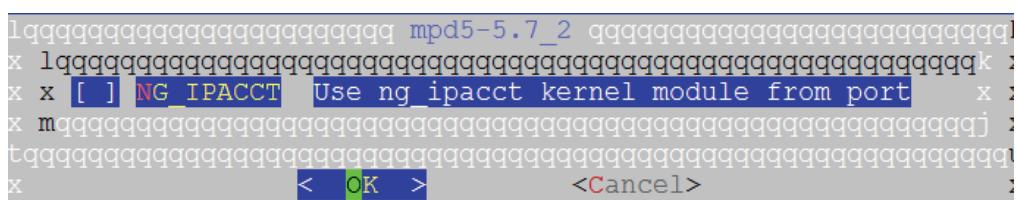
- Portlarda olan ünvanını tapırıq.

```
cd /usr/ports/net/mpd5
```

```
make config
```

- Ünvanına daxil oluruq.

- Heç bir əlavə modul seçmirik.



```
make install clean
```

- Yükləyirik.

Hər iki maşında da MPD5-in quraşdırma fayllarını rezerv nüsxələyirik və adlarını dəyişib əsl adlarla əvəz edirik.

```
cd /usr/local/etc/mpd5/
```

```
mkdir sample
```

```
cp *.sample sample
```

- MPD5 yüklenmiş qovluğa daxil oluruq.

- 'sample' adlı qovluq yaradıraq.

- 'sample' sonluğu ilə bitən bütün quraşdırma fayllarını 'sample' adlı qovluğa nüsxələyirik.

```
mv mpd.conf.sample mpd.conf
```

- Sample quraşdırma faylinin adını dəyişib əsl adla əvəz edirik.

```
mv mpd.secret.sample mpd.secret
```

- Şifrlər yerləşən faylin nüsxəsini əsl ilə əvəz edirik.

```
mv mpd.script.sample mpd.script
```

- Skript nüsxə faylini real faylla əvəz edirik.

## İlk maşında olan quraşdırırmalarımıza başlayaq.

Serverimizin StartUP quraşdırma faylı **/etc/rc.conf**-un məzmunu aşağıdakı kimi olacaq:

```
sshd_enable="YES"
```

```
dumpdev="NO"
```

```
ifconfig_em0="inet 1.2.3.2 netmask 255.255.255.0"
```

```
ifconfig_em1="inet 192.168.1.1 netmask 255.255.255.0"
```

```
defaultrouter="1.2.3.1"
```

```
hostname="main.az"
```

```
firewall_enable="YES"
```

```
firewall_type="OPEN"
```

```
gateway_enable="YES"
```

- ROUTER rejimini aktiv edirik ki, trafikin bir şəbəkə kartından digərinə keçidi olsun.

- MPD5-i StartUP-a əlavə edirik.

```
mpd_enable="YES"
```

Aşağıdakı IP və şəbəkə kartları ardıcılılığı, sözsüz ki, sizdə özünüzə uyğun təyin ediləcək:

Public NETIF - em0

Public IP - 1.2.3.2

Local NETIF - em1

Local user NET - 192.168.1.0/24

MPD5-in quraşdırma faylini açıqlayaq.

`/usr/local/etc/mpd5/mpd.conf` quraşdırma faylinin strukturu aşağıdakı kimidir:

```
startup:
  set user cavid freebsd admin
    - Serverin inzibatçısı cavid adında və freebsd
      şifrəsində olacaq.
  set user faxri freebsd
    - Adı istifadəçi adı isə faxri və şifrəsi freebsd
      olacaq.
  set console self 127.0.0.1 5005
    - MPD5 daemon konsoluna localhost-dan girişi
      5005-ci portdan təyin edirik.
  set console open
    - Konsoldan girişi açırıq.
  set web self 0.0.0.0 5006
    - WEB-dən girişi serverimizin qulaq aslığı
      istənilən IP ünvanda qulaq asmasını deyirik
      (Ancaq bu, təhlükəlidir və daxili IP yazsanız,
      daha yaxşı olar).
  set web open
    - WEB vasitəsilə girişi aktivləşdiririk.
  log -radius -rep -ipv6cp +bund2 +lcp2 +auth2 +ipcp2 +ccp2 +ecp2 +phys2
    - Lazım olan jurnalları aktivləşdiririk, lazımsızları
      isə söndürürük ('+' aktiv edir, '-' passiv edir).

default:
  load pptp_vpn
    - pptp_vpn profaylı yükleyirik.

pptp_vpn:
  create bundle static B1
  set ipcp ranges 192.168.1.1/32 192.168.2.1/32
    - Serverin öz daxili IP ünvani, uzaq serverin daxili
      IP ünvani.
  set iface route 192.168.2.0/24
    - Uzaq serverdə görmək istədiyimiz şəbəkə.
  set bundle enable compression
  set ccp yes mppc
  set mppc yes e40
  set mppc yes e128
  set bundle enable crypt-reqd
  set mppc yes stateless
  create link static L1 pptp
  set link action bundle B1
```

```

set link no pap chap eap
set link yes chap
set auth authname "jamal"           - SiteToSite istifadəçi adı.
set auth password "freebsd"         - SiteToSite şifrə.
set link mtu 1460                   - Paketin maksimum uzunluğu.
set link keep-alive 10 75
set link max-redial 0
set pptp self 1.2.3.2             - Serverin öz PUBLIC IP ünvanı.
set pptp peer 2.3.4.2              - Qarşı tərəfin Public IP-si.
set link enable incoming
open
load dial-in

```

# İdarə edən tərəfin çağırış quraşdırması

**dial-in:**

```

create bundle static pptp0          - Sessiya açırıq.
set ipcp ranges 192.168.3.210/32 192.168.3.211/32
                                         - Server və istifadəçi üçün IP ünvanları təyin edirik.
                                         Daha yaxşı olar ki, hər iki tərəfdə olmayan
                                         aralıqdan istifadə edəsiniz.
                                         - Qarşı tərəfdə görmək istədiyimiz daxili şəbəkə.

set iface route 192.168.2.0/24
set bundle enable compression
set ccp yes mppc
set mppc yes e40
set mppc yes e128
set bundle enable crypt-reqd
set mppc yes stateless
create link static lpptp0 pptp
set link action bundle pptp0
set link no pap
set link yes chap
set auth authname "jamal"          - Qeydiyyatdan keçəcəyimiz istifadəçi adı.
set auth password "freebsd"        - Qeydiyyatdan keçəcəyimiz şifrə.
set link mtu 1460
set link keep-alive 10 75
set link max-redial -1

```

```

set pptp peer 2.3.4.2           - Qarşı tərəfin PUBLIC IP ünvanı.
set link enable incoming

cat /usr/local/etc/mpd5/mpd.secret

jamal          "freebsd"

```

- Qarşı tərəfin qoşulması üçün aşağıdakı istifadəçi adı və şifrəni fayla əlavə edirik ki, qeydiyyatdan keçə bilsin.

- Quraşdırma faylında yazdığımız istifadəçi adı və şifrə.

## İkinci serverimizin quraşdırılması

Serverimizin StartUP quraşdırma faylı **/etc/rc.conf** aşağıdakı kimi olacaq:

```

sshd_enable="YES"
dumpdev="NO"
ifconfig_em0="inet 2.3.4.2 netmask 255.255.255.0"
ifconfig_em1="inet 192.168.2.1 netmask 255.255.255.0"
defaultrouter="2.3.4.1"
hostname="site.az"
firewall_enable="YES"
firewall_type="OPEN"
gateway_enable="YES"

mpd_enable="YES"

```

- ROUTER rejimini aktiv edirik ki, bir şəbəkə kartından digərinə axının keçmə imkanı yaradılsın.
- MPD5-i StartUP-a əlavə edirik.

Aşağıdakı IP və şəbəkə kartları ardıcılılığı, sözsüz ki, sizdə özünüzə uyğun təyin ediləcək:

Public NETIF	- em0
Public IP	- 2.3.4.2
Local NETIF	- em1
Local user NET	- 192.168.2.0/24

MPD5-in quraşdırma faylını açıqlayaq.

**/usr/local/etc/mpd5/mpd.conf** quraşdırma faylımız aşağıdakı kimi olacaq:  
**startup:**

```

set user cavid freebsd admin
set console self 127.0.0.1 5005

```

```

set console open
set web self 0.0.0.0 5006
set web open
log -radius -rep -ipv6cp +bund2 +lcp2 +auth2 +ipcp2 +ccp2 +ecp2
+phys2 +phys3 +iface2 +frame      - Lazım olan jurnalları aktivləşdiririk, lazımsızları
isə bağlayırıq. ('+' aktiv edir, '-' passiv edir.)

default:
load pptp_vpn

pptp_vpn:
create bundle static B1
set ipcp ranges 192.168.2.1/32 192.168.1.1/32
                                         - Serverin öz daxili IP ünvani, uzaq serverin daxili
                                         IP ünvani.
set iface route 192.168.1.0/24      - Qarşı tərəfdə olan görmək istədiyimiz daxili
                                         şəbəkə.

set bundle enable compression
set ccp yes mppc
set mppc yes e40
set mppc yes e128
set bundle enable crypt-reqd
set mppc yes stateless
create link static L1 pptp
set link action bundle B1
set link no pap chap eap
set link yes chap
set auth authname "jamal"          - SiteToSite istifadəçi adı.
set auth password "freebsd"        - SiteToSite şifrə.
set link mtu 1460
set link keep-alive 10 75
set link max-redial 0
set pptp self 2.3.4.2            - Serverin öz PUBLIC IP ünvani.
set pptp peer 1.2.3.2             - Qarşı tərəfin Public IP ünvani.
set link enable incoming
open
load dial-in

```

```
# İdare edilən tərəfin quraşdırılmaları  
dial-out:  
    create bundle static vpn0  
    set iface route 192.168.1.0/24      - Qarşı tərəfdə görmək istədiyimiz daxili şəbəkə.  
    create link static lvpn0 pptp  
    set link action bundle vpn0  
    set auth authname "jamal"  
    set auth password "freebsd"  
    set link mtu 1460  
    set link keep-alive 20 75  
    set link max-redial 0  
    set pptp peer 1.2.3.2              - Qarşı tərəfin Public IP ünvani.  
    open
```

```
cat /usr/local/etc/mpd5/mpd.secret  
jamal          "freebsd"
```

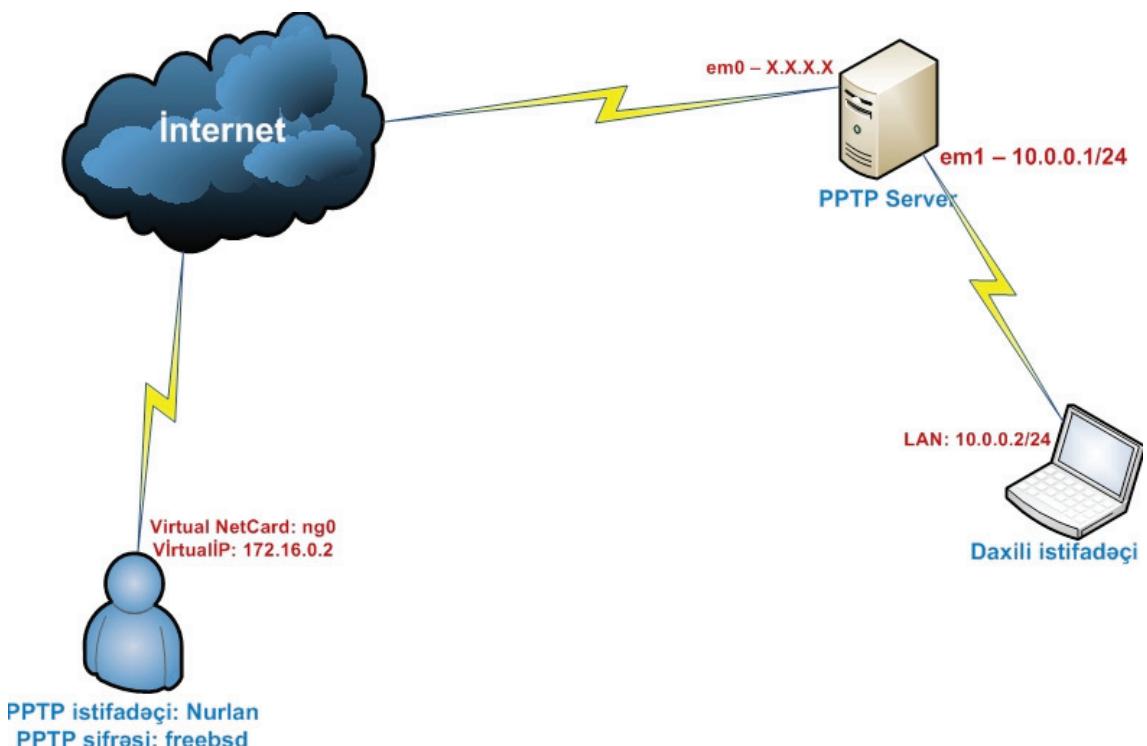
- Fayla qarşı tərəfin qoşulması üçün istifadəçi adı və şifrəsini əlavə edirik ki, autentifikasiyadan keçə bilsin.  
- Quraşdırma faylında yazdığımız istifadəçi adı və şifrəsi.

Sonda hər iki istifadəcidən bir-birlərinə trafik ötürüb test edin.

# FreeBSD MPD5 PPTP (Remote Access VPN)

Məqsədimiz FreeBSD server üzərində MPD5 istifadə edərək **PPTP** (Point-to-Point Tunneling Protocol) vasitəsi ilə istifadəçilərin VPN ilə qoşulmasını təmin etməkdir. Müştəri PPTP vasitəsilə VPN "Remote Access" serverimizə qoşulacaq. GRE(Gateway Routing Encapsulation)

Şəbəkə quruluşumuz aşağıdakı kimi olacaq:



İki şəbəkə kartımız var: **em0** və **em1**.

1. **em0 Public**
2. **em1 "Private LAN"**

**/etc/rc.conf StartUP** quraşdırma faylımız aşağıdakı kimi olacaq:

```
ifconfig_em0="inet X.X.X.X netmask 255.255.255.224"
    - Nəzərimizdə olan PUBLIC IP ünvan.

ifconfig_em1="inet 10.0.0.1 netmask 255.255.255.0"
    - Daxili şəbəkə üçün gateway.
    - Public Gateway.

defaultrouter="X.X.X.1"
hostname="pptpmpd.lan"
sshd_enable="YES"
gateway_enable="YES"
mpd_enable="YES"
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
    - Interfeyslər arası keçidə izin veririk.
    - MPD-ni işə salırıq.
    - Syslogd-ni işə salırıq.
    - Syslogd daemon-un ünvanı.
    - Syslog server yalnız daxildə dinləsin.
    - IPFW-ni aktivləşdiririk.
    - IPFW-nin qaydaları susmaya görə açıqdır.
    - Daxili istifadəçilər üçün NAT-ı açırıq.
    - Daxili istifadəçilərin internetə çıxması üçün
      istifadə edilən NATD interfeysimiz.

firewall_enable="YES"
firewall_type="OPEN"
natd_enable="YES"
natd_interface="em0"
```

Firewall-un kompilyasiya edilmədən işə salınması üçün **/boot/loader.conf** faylına aşağıdakı sətirləri əlavə edirik:

```
ipfw_load="YES"
ipdivert_load="YES"
    - IPFW Firewall-u modullardan çağırırıq.
    - NATD-ni modullardan çağırırıq.
```

```
cd /usr/ports/net/mpd5
make config
make install clean
    - MPD5-i portlardan yükleyirik
      (PPTP-PPPoE qoşulmalar üçün)
    - Heç bir modul seçmirik.
    - Yükleyirik.

cd /usr/local/etc/mpd5/
mkdir sample
    - MPD5-in quraşdırma qovluğuna daxil oluruq.
    - MPD5-in qovluğunda sample adlı qovluq
      yaradıb bütün nüsxə quraşdırma fayllarını ora
      nüsxələyib rezerv üçün saxlayaqq.
```

```

cp *.sample sample                                - Nüsxələyirik.

mv mpd.conf.sample mpd.conf                      - Bütün nüsxə quraşdırma fayllarının adlarını
                                                       conf-a dəyişirik ki, mpd5 onları çağırı bilsin.

mv mpd.secret.sample mpd.secret
mv mpd.script.sample mpd.script

ee mpd.conf                                      - Quraşdırma faylımız.

startup:
# Web Admin İstifadəçi "admin" və şifrəsi "freebsd"
    set user admin freebsd admin
# Console IP və PORT
    set console self 127.0.0.1 5005
    set console open
# WEB IP və PORT
    set web self 0.0.0.0 5006
    set web open

default:
    load pptp_server

pptp_server:
# Müştərilərə veriləcək IP ünvan aralığı
# 'poolsat' qrupa 172.16.0.2 və 172.16.0.254 aralığı təyin edirik.
    set ippool add poolsat 172.16.0.2 172.16.0.254
    create bundle template B
    set iface enable proxy-arp
    set iface idle 0
    set iface enable tcpmssfix
    set ipcp yes vjcomp
# İstifadəçi serverə qoşulan kimi bu skript işə düşəcək.
    set iface up-script /root/scripts/vpn/login.sh
# İstifadəçi serverdən disconnect edilən kimi bu skript işə düşəcək.
    set iface down-script /root/scripts/vpn/logout.sh
# 'poolsat' aralığından gələn istifadəçilər üçün görünən gateway '172.16.0.1'
# IP-si olacaq. (Real olmayan IP ünvanda yazmaq olar.)
    set ipcp ranges 172.16.0.1/32 ippool poolsat

```

```

# Burada biz DNS serveri təyin edirik.
      set ipcp dns 8.8.8.8 8.8.4.4

# Microsoft-un Point-to-Point compression (MPPC)-ni aktiv edirik.(Sixılma)
      set bundle enable compression
      set ccp yes mppc
      set mppc yes compress e40 e56 e128 stateless

# PPTP üçün 'L' adlı şablon yaradırıq.
      create link template L pptp

# Şablonun verilənlərlə birləşməsi
      set link action bundle B

# Multilinkdə bəzi xərclər olur, amma 1500 MTU olur.
      set link enable multilink
      set link yes acfcomp protocomp
      set link no pap chap eap
      set link enable chap
      set link enable chap-msv1
      set link enable chap-msv2

# MTU-nu kiçildirik ki, GRE-nin paket fragmentasiyasından qaça bilək.
      set link mtu 1460
      set link keep-alive 10 60

# Burada biz PPTP qəbul edən interfeysi göstəririk, hansı ki, public IP-yə
# malikdir.
      set pptp self em0
# Qəbulu izin veririk.
      set link enable incoming

```

**mkdir -p /root/scripts/vpn/** - Skriptlərimiz üçün qovluq yaradırıq.

Skriptlərimiz üçün faylları və jurnal faylini yaradırıq.

```

touch /root/scripts/vpn/login.sh
touch /root/scripts/vpn/logout.sh
touch /var/log/mpd.connect.log

```

Skriptlərimizi yerinə yetirilən edirik:

```

chmod +x /root/scripts/vpn/login.sh
chmod +x /root/scripts/vpn/logout.sh

```

```
ee /root/scripts/vpn/login.sh      - İstifadəçinin qoşulan kimi işə salacağı skripti yazırıq.  
#!/bin/sh
```

```
DATE=`date +%d-%m-%Y_%H:%M:%S`  
  
echo "UP $DATE user:$5 ip:$8 local_ip:$4 $1" >> /var/log/mpd.connect.log  
# Göstərilən strukturu '/var/log/mpd.connect.log' faylına yazırıq.
```

**\$DATE** - Dəyişən sadəcə gün, ay, il, saat, saniyə formasını mənimsəyir.  
**\$5** - Qoşulan istifadəçi adı.  
**\$8** - Hansı Public IP ünvandan RA (**Remote Access**) ilə VPN serverimizə qoşulurlar.  
**\$4** - İstifadəciyə verilən IP ünvan.  
**\$1** - İstifadəçi üçün generasiya edilən virtual şəbəkə kartı.

```
ee /root/scripts/vpn/logout.sh    - İstifadəçinin disconnect edilən kimi işə salınacaq skripti.  
#!/bin/sh
```

```
DATE=`date +%d-%m-%Y_%H:%M:%S`  
  
echo "DOWN $DATE user:$5 local_ip:$4 $1" >> /var/log/mpd.connect.log  
# Göstərilən strukturu '/var/log/mpd.connect.log' faylına yazırıq.
```

**\$DATE** - Dəyişən sadəcə gün, ay, il, saat, saniyə formasını mənimsəyir.  
**\$5** - Qoşulan istifadəçi adı.  
**\$4** - İstifadəciyə verilən IP ünvan.  
**\$1** - İstifadəçi üçün generasiya edilən virtual şəbəkə kartı.

Jurnallar aşağıdakı formada çıxacaq.

```
UP 14-04-2015_10:01:11 user:nurlan ip:5.44.39.103 local_ip:172.16.0.2 ng0  
DOWN 14-04-2015_10:04:04 user:nurlan local_ip:172.16.0.2 ng0  
UP 14-04-2015_10:11:23 user:nurlan ip:5.44.39.30 local_ip:172.16.0.2 ng0  
DOWN 14-04-2015_10:14:24 user:nurlan local_ip:172.16.0.2 ng0  
UP 14-04-2015_10:15:03 user:cavid ip:5.44.39.30 local_ip:10.11.11.100 ng0  
ee mpd.secret
```

- Qoşulmalar üçün istifadəçi adı və şifrə faylı.

MyLogin	MyPassword
PeerLogin	PeerPassword

```

# Bu istifadəçilərə statik təyin etdiyimiz IP ünvanları veriləcək.
cavid          "freebsd"      10.11.11.100
faxri          "freebsd"      10.11.11.11

# Bu iki istifadəçiye təyin etdiyimiz 'poolsat' aralığında
# olan IP ünvanlarından azad olanı veriləcək.
# Yəni 172.16.0.2-172.16.0.254 aralığında olan IP-lər.
ramil          "freebsd"
nurlan         "freebsd"

ee /etc/syslog.conf
!mpd
*:.*           /var/log/mpd.log

touch /var/log/mpd.log
chmod 600 /var/log/mpd.log

/etc/rc.d/syslogd restart
/usr/local/etc/rc.d/mpd5 start

ps axw | grep mpd

```

- MPD5-in jurnallarını seçdiyimiz fayla filtrlərdirik.  
(Faylin sonuna əlavə edirik.)

- mpd.log faylı yaradırıq.  
- Yalnız root istifadəçi üçün bu fayla yetki veririk.

- Syslog daemon-u yenidən işə salırıq.

- "ee /etc/rc.conf"-dan şərhi silməyi unutmayın.

- Proseslərdə mpd-nin işləməsini axtarıraq.

**Qeyd:** Firewall istifadə edirikdə, 1723-cü portu açmağı unutmayın.

Firewall-da aşağıdakı qaydalarla lazımi portları aça bilərsiniz:

```

LanOut=em0
LanIn=em1
VpnIp="172.16.0.0/24"
cmd="ipfw -q add"

# VPN girişə və çıkışa icazə veririk.
$cmd allow tcp from any to me 1723 in via ${LanOut}
$cmd allow tcp from me 1723 to any out via ${LanOut}
$cmd allow gre from me to any out via ${LanOut}
$cmd allow gre from any to me in via ${LanOut}

```

```

$cmd allow ip from any to ${VpnIp} out
# VPN IP-sindən Mail serverə yetki veririk.
$cmd allow ip from ${VpnIp} to any in via ${MailServer}
# MailServer-dən tcp ilə bütün VPN istifadəçilərinə yetki veririk.
$cmd allow tcp from ${MailServer} to ${VpnIp}
# VPN istifadəçilərə də həmçinin.
$cmd allow tcp from ${VpnIp} to ${MailServer}
$cmd allow udp from ${VpnIp} to ${MailServer}

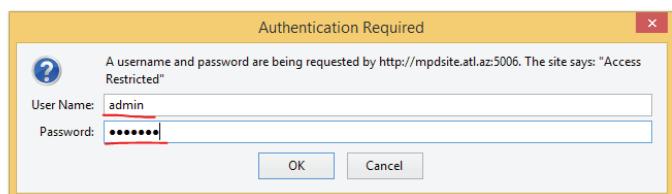
```

Sonda smart telefon və ya kompüterdə PPTP client quraşdırıb serverimizə qoşuluruq. İstifadəçi bazası **/usr/local/etc/mpd5/mpd.secret** faylındadır.

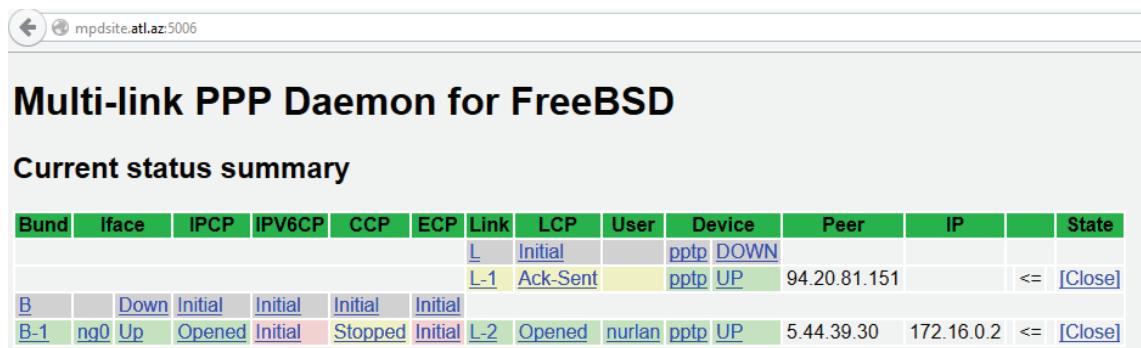
İnzibatçı üçün istifadəçi adı və şifrə **/usr/local/etc/mpd5/mpd.conf** faylında təyin etdiyimizdir.

 mpdsite.atl.az:5006

Authorization Required



Nəticədə bir istifadəçi onlayn olduğu halda olan şəkil aşağıdakı kimi olacaq:

 mpdsite.atl.az:5006

## Multi-link PPP Daemon for FreeBSD

### Current status summary

Bund	Iface	IPCP	IPV6CP	CCP	ECP	Link	LCP	User	Device	Peer	IP		State	
						L	Initial		pptp	DOWN				
						L-1	Ack-Sent		pptp	UP	94.20.81.151	<=	[Close]	
B	ng0	Down	Initial	Initial	Initial	Initial								
B-1		Up	Opened	Initial	Stopped	Initial	L-2	Opened	nurlan	pptp	UP	5.44.39.30	172.16.0.2	<= [Close]

# FreeBSD NTOP (NetFlow Trafikin monitoring edilməsi)

**NTOP** – Bu paket Cisco Router üzərindən keçən bütün trafiki qrafiklərlə port və protokollara görə göstərmək üçündür. Eyni zamanda o, avadanlıqların xəritəsini də çəkir. Bu səbəbdən NTOP-un üzərinə BackBone(Arxa Router)-in NetFlow paketlərini yönəltməniz daha məqsədə uyğun olar. NTOPT Cisco-nun NetFlow **v5/v7/v9** versiyalarını dəstəkləyir.

Ancaq onu unutmayın ki, NTOPT bütün bu qrafik statistikaları özündə yalnız ilk yenidənyüklən-məyədək saxlayır. Sistem yenidən işə düşəndə o, bütün işlərə yenidən başlayacaq. Yadda saxladığı ancaq RRDTool-un qrafikləri olur. Trafik NTOPT serverin üzərinə gəldiyinə görə o serverin resurslarını bir az çox vermək lazımdır. 4GB DDR, Dual Core CPU 2.13 GHZ, HDD 20GB.

İmkanları:

- Trafiki (IP: TCP, UDP, ICMP, GRE, IPSEC) protokollarına görə çeşidləyir.
- Trafikin daxilində çoxlu metodlarla filtrləmə.
- Trafiki yalnız RRDTool formatında saxlayır.
- NetBios adlarının və Mail adlarının müəyyən edilməsi.
- Passiv şəkildə kompüterlərin OS-larının təyin edilməsi.
- IP trafikini protokollara göstərmək.
- Trafikin ünvanına və təyinatına görə analiz edilməsi.
- Strukturun sxem Subnet/MASK şəklində göstərmə imkanı.
- NetFlow/sFlow kollektor rejimində işləyərək, Cisco/Juniper Router və Switch-lərinin paketlərini dəstəkləyir.
- Trafiki RMON formatında generasiya edir.

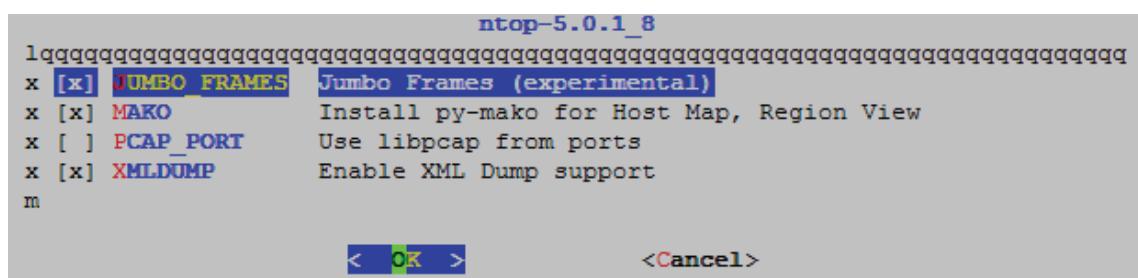
System Control-a lazımi dəyişənləri əlavə etməliyik ki, sistemdən böyük paketlər keçə bilsin.  
`/etc/sysctl.conf` kernel StartUP faylımiza aşağıdakı sətirləri əlavə edirik:

```
net.inet.tcp.rfc1323=1  
net.inet.ip.intr_queue_maxlen=1000  
kern.ipc.maxsockbuf=900000000  
net.inet.tcp.sendspace=300000000  
net.inet.tcp.recvspace=300000000  
net.inet.udp.recvspace=300000000  
net.inet.tcp.recvspace=300000000
```

NTOP-u portlardan yükleyəcəyik. Onun üçün də portları yeniləyirik ki, ən yeni versiyasını yükleyək.

```
portsnap fetch extract update
```

```
cd `whereis ntop | awk '{ print $2 }'` - Port ünvanına daxil oluruq.  
make config - Lazımi modulları seçirik.
```



```
make install - Heç bir halda clean etməyin, çünkü kompilyasiya  
rehash - edilmiş kodlarda bizə lazım ona fayl olacaq.  
- Əmr bazasını yeniləyirik.
```

Yüklənmə uzun çəkəcək, çünkü asılılıqda olan çoxlu paketlər yüklenib kompilyasiya ediləcək. Ancaq yüklənmə bitdikdən sonra kompilyasiya edilən qovluqdan ntop-un quraşdırma faylini NTOP quraşdırma qovluğununa nüsxələyirik.

```
cp /usr/ports/net/ntop/work/ntop-5.0.1/packages/FreeBSD-ports/net/ntop/files/  
ntop.conf.sample /usr/local/etc/ntop.conf
```

Əgər yenə də səhvən ‘**make install clean**’ əmrini yiğmisiñizsa, onda elə yenə də portun daxilində ‘make extract’ əmrini daxil etsəniz, həmin faylı əldə edə biləcəksiniz.

Faylin icra yetkisini əlindən alırıq.

```
chmod -x /usr/local/etc/ntop/ntop.conf
```

Sistemə ‘**ntop**’ adlı istifadəçi və qrup əlavə edirik.

**adduser** əmri ilə sistemə ‘**ntop**’ adlı istifadəçi əlavə etdikdə ‘**UID**’ və ‘**GID**’-i 2001 təyin edirik, çünki onu bir azdan istifadə edəcəyik. Ntop-a şifrə təyin edirik və ‘**nologin**’ shell veririk ki, uzaqdan giriş mümkün olmasın. Və ev qovluğu kimi, ‘**ntop**’-un bazası ünvanını təyin edirik. Baza ünvanı ntop-u yükleyəndə ən sonda mesaj kimi çap edilir:

```
adduser                                         - Ntop istifadəçisini sistemə əlavə edirik.  
Username: ntop  
Full name: Ntop User  
Uid (Leave empty for default): 2001  
Login group [ntop]:  
Login group is ntop. Invite ntop into other groups? []:  
Login class [default]:  
Shell (sh csh tcsh nologin) [sh]: nologin  
Home directory [/home/ntop]: /var/db/ntop  
Home directory permissions (Leave empty for default):  
Use password-based authentication? [yes]:  
Use an empty password? (yes/no) [no]:  
Use a random password? (yes/no) [no]:  
Enter password: şifre  
Enter password again: təkrar_şifre  
Lock out the account after creation? [no]:  
Username   : ntop  
Password   : *****  
Full Name  : Ntop User  
Uid        : 2001  
Class      :  
Groups     : ntop  
Home       : /var/db/ntop  
Home Mode  :  
Shell      : /usr/sbin/nologin  
Locked     : no
```

```
OK? (yes/no): yes
adduser: INFO: Successfully added (ntop) to the user database.
Add another user? (yes/no): no
Goodbye!
```

```
cat /etc/passwd | grep ntop
ntop:*:2001:2001:Ntop User:/var/db/ntop:/usr/sbin/nologin
```

NTOP-un baza ünvanını öz istifadəcisinə və qrupuna mənimsədirik.  
**chown -R ntop:ntop /var/db/ntop**

NTOP-un quraşdırma faylini düzəldək.

```
ee /usr/local/etc/ntop/ntop.conf
## Hansı istifadəçi adından NTOP işləyəcək?
--user ntop
```

```
## Mütləq şərh yerləşdiririk. Əks halda, işə düşməyəcək.
# --set-pcap-nonblock
```

# NTOP-un bazasının ünvanı.

```
--db-file-path /var/db/ntop
```

```
# NTOP bu şəbəkə kartı ilə NetFlow paketləri qəbul edəcək.
--interface em0
```

## NO-MAC-i siz SPAN istifadə etdiyiniz halda, işlədə bilərsiniz. Bizim halda lazımdır.

```
##--no-mac
# HTTP serverlə işləmək istəsək, aşağıdakı sintaksisi istifadə edə bilərik
##--http-server 3000
```

```
# HTTPS serverdə işləməsini deyirik.
--https-server 3001
```

```
# NTOP-un daxili şəbəkə kimi nəzərə alacağı şəbəkə aralığı.
--local-subnets 192.168.0.0/16,172.0.0.0/8
```

```
# NTOP-a müvəqqəti bir domain adı verin. Verməsəniz də, özü avtomatik verəcək.
```

```
--domain ntop.az
```

# Deyirik ki, NTOP daemon kimi işləsin.

```
--daemon
```

```
/usr/local/bin/ntop -h
```

- Bütün quraşdırma siyahısını görə bilərik.

Ntop-un jurnalları susmaya görə '/var/log/messages' faylinə yiğilir.

NTOP-u işə salmazdan önce WEB vasitəsilə daxil olmağa, 'admin' istifadəçi adı üçün şifrə təyin edirik.

Burada **2001** isə bir az önce danışdığını **ntop** istifadəçisinin **UID** və **GID**-dir.

```
root@ntop:~ # /usr/local/bin/ntop -P /var/db/ntop -u 2001 -A
```

```
Tue Apr 14 20:51:45 2015 Initializing gdbm databases
```

```
Tue Apr 14 20:51:45 2015 Setting administrator password...
```

```
ntop startup - waiting for user response!
```

```
Please enter the password for the admin user: şifre
```

```
Please enter the password again: təkrar_şifre
```

```
Tue Apr 14 20:52:19 2015 Admin user password has been set
```

```
Tue Apr 14 20:52:19 2015 Admin password set...
```

NTOP-un WEB interfeysinə giriş üçün **3000** və **3001**-ci portlara firewall-da icazə verməyi unutmayın.

NTOP-u StartUP-a əlavə edirik.

```
echo 'ntop_enable="YES"' >> /etc/rc.conf
```

```
echo 'ntop_flags="@/usr/local/etc/ntop/ntop.conf"' >> /etc/rc.conf
```

```
/usr/local/etc/rc.d/ntop start # NTOP-u işə salırıq.
```

Unutmayın, orada bir BUG çıxacaq. Təcrübədən irəli gələrək qeyd edə bilərəm FreeBSD 10.1 istifadə zamanı NTOP start edildikdən sonra yalnız tcp6-da qulaq aşındı, bu səbəbdən kerneli tcpv6-sız kompilyasiya etmək məcburiyyətində qaldım.

```
cd /sys/`uname -p`/conf
```

```
cp GENERIC newkern
```

```
ee newkern
```

```
#options          INET6
```

```
cd ../../..
```

```
make buildkernel KERNCONF=newkern
```

```
make installkernel KERNCONF=newkern
```

```
reboot
```

- Lazımsız driver-lərdən və 'IPV6'-dan başqa hər şey yerində qaldı.

- Qarşısına şərh yazılıq ki, kompilyasiya edilməsin.

- '/usr/src' ünvanına daxil oluruq ki, kompilyasiya edək.

- Kompilyasiya edirik.

- Yükləyirik.

- Sistemi yenidən yükləyirik.

Sonda gedirik WEB serverimizə [https://server\\_ip\\_address:3001](https://server_ip_address:3001) və admin loginlə sistemə daxil oluruq.

The screenshot shows the ntop web interface. At the top, there are tabs for Global Statistics, em0 Report, Protocol Distribution, and Application Protocols. Below these, there's a table for Network Interface(s) showing details for em0. The table includes columns for Name, Device, Type, Speed, Sampling Rate, MTU, Header, Address, and IPv6 Addresses. It also shows Local Domain Name (freebsd.lan), Capturing Since (Tue Apr 14 21:33:28 2015 [1:13]), Hosts ([112 active] [120 total]), and Active Sessions (205 [Max: 205]). At the bottom of the interface, there's a footer with copyright information and a search bar.

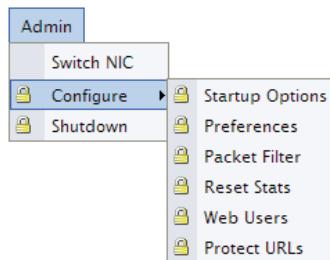
Report created on Tue Apr 14 21:34:41 2015 [root uptime: 1:13]  
Generated by ntop v0.9.1 (0.9.1.085) on FreeBSD 10.1  
© 1999-2012 by Luca Deri, built: Apr 14 2015 19:14:48  
Version: the CURRENT stable version  
Listening on [em0] for all packets (i.e. without a filtering expression)  
Web reports include all interfaces (merged)

Bizim halda serverin IP ünvani **10.50.3.137**-dir. Öncə plugin-lərə baxırıq. Hamisının status 'Active'-ində 'Yes' olmalıdır. Əgər deyilsə, özümüz 'YES' edirik.

The screenshot shows the 'Available Plugins' section. On the left, there's a sidebar with 'Plugins' and 'Admin' tabs, and a list of available plugins: cPacket, ICMPWatch, NetFlow, Round-Robin Database, sFlow, and All. The 'All' tab is selected. In the main area, there's a table titled 'Available Plugins' with columns for View, Configure, Description, Version, Author, and Active [click to toggle]. The table lists five plugins:

View	Configure	Description	Version	Author	Active [click to toggle]
	cPacket	This plugin is used collect traffic statistics emitted by cPacket's cTap devices. Received flow data is reported as a separate 'NIC' in the regular <b>ntop</b> reports. Remember to switch the reporting NIC.	0.1	L.Deri	Yes
	icmpWatch	This plugin produces a report about the ICMP packets that ntop has seen. The report includes each host, byte and per-type counts (sent/received).	2.4a	L.Deri	Yes
	NetFlow	This plugin is used to setup, activate and deactivate NetFlow support. <b>ntop</b> can both collect and receive NetFlow V1/V5/V7/V9 and IPFIX (draft) data. Received flow data is reported as a separate 'NIC' in the regular <b>ntop</b> reports. Remember to switch the reporting NIC.	4.4	L.Deri	Yes
	rrdPlugin	This plugin is used to setup, activate and deactivate ntop's rrd support. This plugin also produces the graphs of rrd data, available via a link from the various 'Info about host xxxx' reports.	2.9	L.Deri	Yes
	sFlow	This plugin is used to setup, activate and deactivate ntop's sFlow support. <b>ntop</b> can both collect and receive sFlow data. Note that ntop.org is a member of the sFlow consortium. Received flow data is reported as a separate 'NIC' in the regular <b>ntop</b> reports. Remember to switch the reporting NIC.	3.0	L.Deri	Yes

Sonra sistemdə admin olmaq üçün gedirik **Admin** -> **Configure** -> **Startup Options** admin istifadəçi adı və şifrəni daxil edirik.



Capture interfeysimiz kimi 'em0' alətini mənimsədib '**Save Prefs**' edirik.

**Configure ntop**

[ Basic Prefs ] [ Display Prefs ] [ IP Prefs ] [ FC Prefs ] [ Advanced Prefs ] [ Debugging Prefs ]

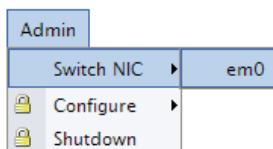
Preference	Configured Value
Capture Interfaces (-i)	<input checked="" type="checkbox"/> em0 <input type="checkbox"/> lo0
Capture Filter Expression (-B)	<input type="text"/> Restrict the traffic seen by ntop. BPF syntax.
Packet sampling rate (-C)	<input type="text"/> 0 Sampling rate [1 = no sampling]
HTTP Server (-w)	<input type="text"/> 3000 HTTP Server [Address:]Port of ntop's web interface
HTTPS Server (-W)	<input type="text"/> 0 HTTPS Server [Address:]Port of ntop's web interface
Enable Session Handling (-z)	<input checked="" type="radio"/> Yes <input type="radio"/> No
Enable Protocol Decoders (-b)	<input checked="" type="radio"/> Yes <input type="radio"/> No
Flow Spec (-F)	<input type="text"/> Flow is a stream of captured packets that match a specified rule
Local Subnet Address (-m)	<input type="text"/> Local subnets in ntop reports (use , to separate them). Mandatory for packet capture files
Known Subnet Address (-M)	<input type="text"/> Known subnets in ntop reports (use , to separate them). Mandatory for packet capture files
Sticky Hosts (-c)	<input type="radio"/> Yes <input checked="" type="radio"/> No Don't purge idle hosts from memory
Track Local Hosts (-g)	<input type="radio"/> Yes <input checked="" type="radio"/> No Capture data only about local hosts
Disable Promiscuous Mode (-s)	<input type="radio"/> Yes <input checked="" type="radio"/> No Don't set the interface(s) into promiscuous mode
Run as daemon (-d)	<input type="radio"/> Yes <input checked="" type="radio"/> No Run Ntop as a daemon

[Save Prefs](#) [Restore Defaults](#)

Except as indicated, settings take effect at next startup

See [Show Configuration](#) for runtime values

**Qeyd:** Ancaq unutmayın ki, əgər siz 'Capture Interfaces'-də şərt olaraq şəbəkə kartını mənimsətsəniz, onda virtual alətin üstündə olan trafiki görə bilməyəcəksiniz. Onun üçün siz trafikə device-ı dəyişib baxmalısınız. Aşağıdakı şəkildə göstərilir.



NetFlow Virtual alətini əlavə edirik.

Plugins -> NetFlow -> View/Configure -> Add NetFlow Device

Aşağıdakı quraşdırmaları əlavə edirik.

**NetFlow Device: Olduğu adla qalır**

**Local Collector UDP Port: 9995** ('Set Port' Button mütləq sıxılmalıdır.)

**Virtual NetFlow Interface Address: 172.0.9.5/255.255.0.0** (Set Interface Address mütləq sıxılmalıdır)

**Enable Session Handling: YES**

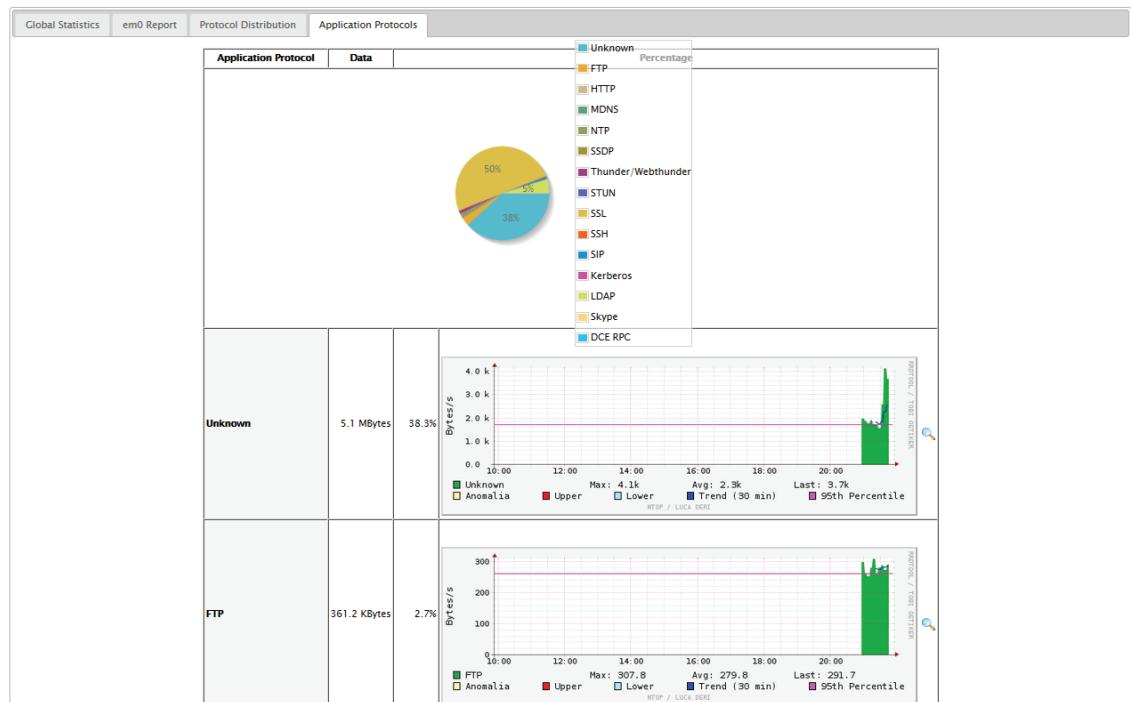
Sonra serverə 'netstat -na' edib baxırıq, listen UDP port **9995** olmalıdır.

```
udp4      0      0 *.9995          *.*
```

Sonda isə Cisco Routerin üstündə aşağıdakı əmrləri daxil edirik.

```
router(config)#ip flow-export destination 172.0.9.5 9995
router(config)#ip flow-export source GigabitEthernet 0/1
router(config)#ip flow-export version 5
router(config)#ip flow-cache timeout active 1
router(config)#ip flow-cache timeout inactive 15
```

Sonda da WEB-İə bütün trafikə baxırsınız.



# BÖLÜM 13

## IPSec StS VPN, FreeBSD-Cisco StS VPN, Stunnel, HAST Cluster

- / FreeBSD IPSec Site-to-Site VPN
- / FreeBSD ilə Cisco arasında IPSec vasitəsilə Site-to-Site VPN qurulması
- / FreeBSD Stunnel
- / FreeBSD HAST Cluster

Öncəki başlığımızda şirkətin daxili şəbəkəsini digər şirkətin daxili şəbəkəsi ilə əlaqələndirmək üçün qurulan VPN MPD5 üzərində oldu. Ancaq bu başlıqda eyni tələb İPsec üzərində ediləcək. Nəzərə alın ki, şirkətlərin daxili resurslarının yayılması tələbi yarandıqda hər şirkətin özünəməxsus olan bir brendin avadanlığı və ya program təminatı ola bilər. Onlardan əksər istifadə ediləni Cisco-dur və bu başlıqda Cisco 7200 router ilə FreeBSD əməliyyat sistemi arasında VPN qurulması açıqlanacaq. Elə hallar ola bilər ki, siz qısa bir zamanda açıq tekst yollaya bilmə imkanına sahib olan hansısa bir daemonu qurmuşunuz və artıq işi dayandırmadan kanalın şifrələnməsi tələbi yaranmışdır. Bu halda sizin köməyinizə Stunnel çatacaq. Başlığımızda Stunnel qurulacaq və sınaqdan keçiriləcək. Sizdən FreeBSD əməliyyat sisteminin üzərində işləyən program təminatının dayanıqlı işləməsi tələbi yarana bilər. Bu halda aktiv/aktiv və aktiv/passiv rejimlərdən birini qurma tələbi yaranır. Başlığımız aktiv/passiv klaster üçün sinxronizasiya ediləcək diskin qurulmasını açıqlayır və onu real rejimdə sınaqdan keçirir.

# FreeBSD IPSec Site-to-Site VPN

**IPSec** (IP Security) – verilənlərin müdafiəsini təmin etmək üçün protokollar dəstindən ibarətdir, hansı ki, şəbəkə protokolu IP vasitəsilə ötürülür. Əsliyin(autentifikasiya) sübutunu həyata keçirməyə, bütövlüyün yoxlanışına və IP paketlərin şifrələnməsinə icazə verir. IPSec tərkibində, həmçinin internet şəbəkəsində açarların mübadiləsində müdafiə üçün protokollara sahibdir. Əsasən, təşkilatlarda VPN qoşulmalarında istifadə edilir. Bu başlığımızda IPSec vasitəsilə iki FreeBSD server arasında Site-to-Site VPN quraşdıracaqıq.

IPSec iki protokoldan ibarətdir:

- *Encapsulation Security Payload (ESP)*, simmetrik şifrələnmə (Blowfish, 3DES) alqoritməri istifadə edərək IP paket verilənlərini şifrələyərək üçüncü tərəfdən müdafiə edir.
- *Authentication Header (AH)*, IP paketin başlığını üçüncü tərəfin müdaxiləsindən, kriptoqrafik kontrol hesabının dəyişdirilməsi üsulundan qoruyur və müdafiə olunan keşlənmə funksiyası ilə IP paketlərin başlıqlarının sütunlarını keşləyir.

Tələbdən asılı olaraq, ESP və AH birlikdə və ya ayrılıqda istifadə edilə bilər.

İPSec heç bir asılılıq olmadan iki host arasında olan trafikin şifrələnməsi üçün istifadə edilə bilər. Ya da iki korporativ şəbəkənin bir-birini təhlükəsiz kanal vasitəsilə görməsini təmin etmək mümkündür.

**Qeyd:** FreeBSD, həmçinin avadanlıq səviyyəsində IPSec dəstəkləyir. Ancaq susmaya görə olan IPsec-də olan bütün funksionallığı sahib deyil. İşə salmaq üçün isə kernel-inizə **options FAST\_IPSEC** əlavə edib, kompilyasiya etməyiniz kifayətdir.

Bütün testlər Vmware Workstation üzərində aparılmışdır.

İki ədəd FreeBSD 10.1 x64 serverimiz var. Serverlərimiz haqda məlumatları açıqlayacaq.

İlk serverimiz

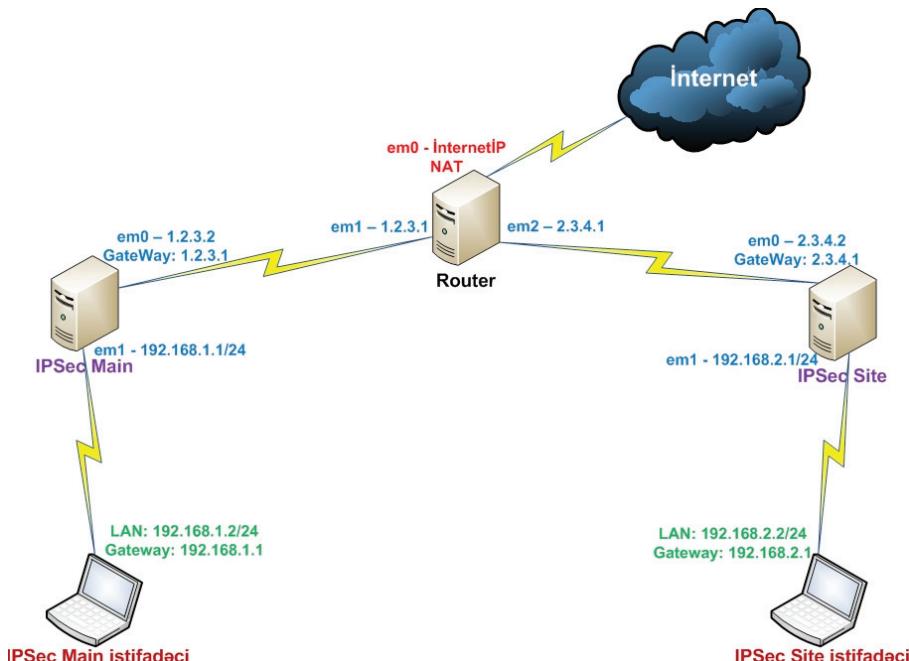
Hostname: **ipsec1.freebsd.lan**  
Public IP: **1.2.3.2**  
LAN IP: **192.168.1.0/24**

İkinci serverimiz

Hostname: **ipsec2.freebsd.lan**  
Public IP: **2.3.4.2**  
LAN IP: **192.168.2.0/24**

**Qeyd:** Siz eyni IPSec VPN-lə eyni PUBLİC IP üzərində, həmçinin istifadəçilər üçün NAT-da quraşdırı bilərsiniz. Ancaq **ipsec-tools** portun kompilyasiya zamanı NAT-T modulunu seçməyi unutmayın.

Şəbəkə quruluşumuz aşağıdakı şəkildəki kimi olacaq:



## **ilk serverin ipsec1.freebsd.lan quraşdırılması**

Serverimizin şəbəkə kartları:

```
em0      - 1.2.3.2  
em1      - 192.168.1.1
```

```
cd /sys/amd64/conf  
cp GENERIC ipsekkern
```

- Kernel-i IPSec üçün kompilyasiya edirik.  
- Kernel faylını nüsxələyirik.

**ipsekkern** adlı kernel faylımızda uyğun dəyişiklikləri edirik.

```
ident ipsekkern
```

- Identin adını dəyişib mütləq **ipsekkern** yazırıq.

```
device      crypto  
device      gif  
options     IPFIREWALL  
options     IPFIREWALL_VERBOSE  
options     IPFIREWALL_VERBOSE_LIMIT=1000  
# IPSEC  
options     IPSEC  
options     IPSEC_DEBUG
```

```
cd ../../..  
make buildkernel KERNCONF=ipsekkern  
make installkernel KERNCONF=ipsekkern
```

- Üç ünvan geri qayıdırıq ki, kompilyasiyanı işə salaq.  
- Kompilyasiyaya başlayırıq.  
- Yeni kernel yükleyirik.

Command line ilə **GIF** (Generic tunnel interface) yaratmaq üçün aşağıdakı əmrlərdən istifadə etmək lazımdır:

```
ifconfig gif0 create  
ifconfig gif0 1.2.3.2 2.3.4.2  
ifconfig gif0 inet 192.168.1.1 192.168.2.1 netmask 0xffffffff  
route add -inet 192.168.2.0/24 192.168.2.1
```

```
netstat -rn | grep 192.168.2.0/24  
192.168.2.0/24      192.168.1.1      UGS          em1
```

- Route cədvəlimizə baxırıq.

İlk serverimizin **/etc/rc.conf** StartUP quraşdırma faylinin məzmunu aşağıdakı kimidir:

```
ifconfig_em0="inet 1.2.3.2 netmask 255.255.255.0"  
ifconfig_em1="inet 192.168.1.1 netmask 255.255.255.0"
```

```
defaultrouter="1.2.3.1"
hostname="ipsecl.freebsd.lan"
gif_interfaces="gif0"
gifconfig_gif0="1.2.3.2 2.3.4.2"
ifconfig_gif0="inet 192.168.1.1 192.168.2.1 netmask 255.255.255.0"
sshd_enable="YES"
inetd_enable="YES"
gateway_enable="YES"
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/firewall.conf"
static_routes="vpn"
route_vpn=" -net 192.168.2.0/24 192.168.1.1 "
racoon_enable="YES"
racoon_flags="-F -f /usr/local/etc/racoon/racoon.conf -l /var/log/racoon.log"
ipsec_enable="YES"
ipsec_file="/etc/ipsec.conf"
```

IPSec quraşdırmları üçün **/etc/ipsec.conf** quraşdırma faylinin məzmunu aşağıdakı kimi olacaq:

```
#!/sbin/setkey -f
flush;
spdflush;
```

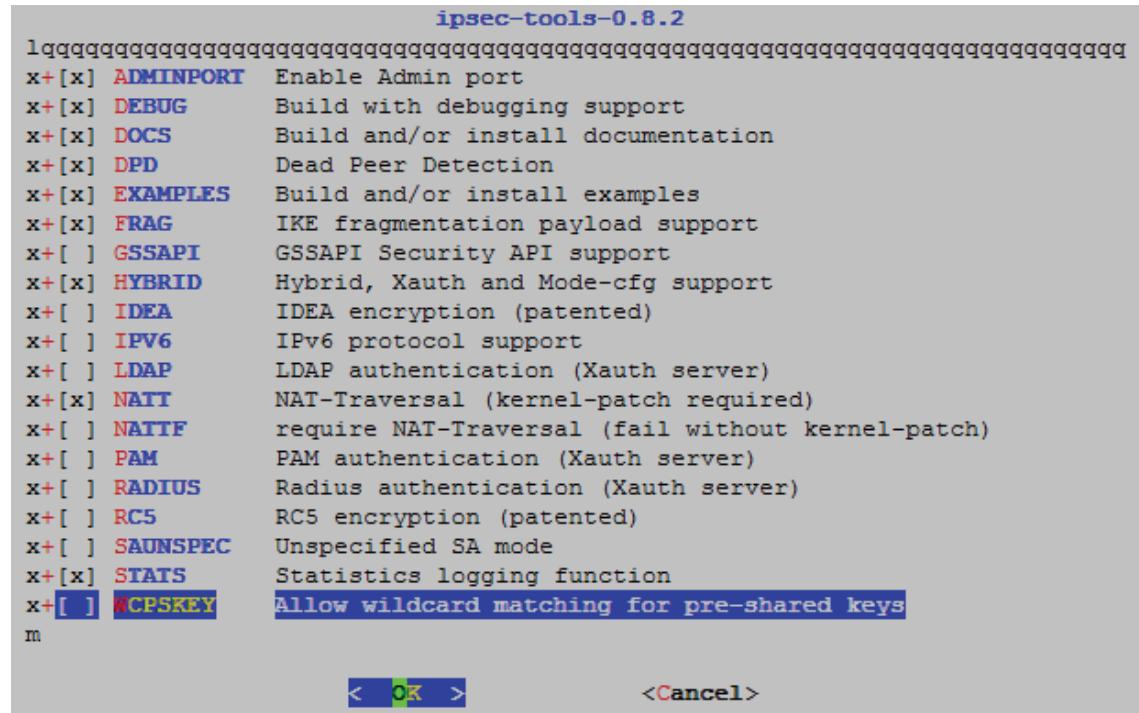
```
spdadd 192.168.1.0/24 192.168.2.0/24 any -P out ipsec esp/tunnel/1.2.3.2-2.3.4.2/require;
spdadd 192.168.2.0/24 192.168.1.0/24 any -P in ipsec esp/tunnel/2.3.4.2-1.2.3.2/require;
```

Firewall üçün **/etc/firewall.conf** faylımiza aşağıdakı sətirləri əlavə edirik ki, sistem yenidənyüklənməsindən sonra avtomatik işə düşsün.

```
ipfw add 00001 allow ip from any to any via gif0
ipfw add 00002 allow udp from 1.2.3.2 to 2.3.4.2 isakmp
ipfw add 00003 allow udp from 2.3.4.2 to 1.2.3.2 isakmp
ipfw add 00004 allow esp from 1.2.3.2 to 2.3.4.2
ipfw add 00005 allow esp from 2.3.4.2 to 1.2.3.2
ipfw add 00006 allow ipencap from 1.2.3.2 to 2.3.4.2
ipfw add 00007 allow ipencap from 2.3.4.2 to 1.2.3.2
ipfw add 00100 allow ip from any to any via lo0
```

```
ipfw add 00200 deny ip from any to 127.0.0.0/8  
ipfw add 00300 deny ip from 127.0.0.0/8 to any
```

```
cd /usr/ports/security/ipsec-tools/      - IPSec aletlerinin port ünvanına daxil olurq ki,  
make config                                yükleyək.  
                                              - Lazımi modulları seçirik.
```



```
make install clean          - Yükleyirik.
```

VPN quraşdırma **/usr/local/etc/racoon/racoon.conf** faylımızı aşağıdakı kimi edirik:  
path pre\_shared\_key "/usr/local/etc/racoon/psk.txt";

```
listen
{
    isakmp 1.2.3.2;
}
```

```
# Debug etmək üçün öncədən bu sətri yazırıq. İşlədikdən sonra silmək olar.
```

```

log debug2;

remote 2.3.4.2
{
exchange_mode aggressive;

my_identifier address;

lifetime time 24 hour;

proposal {
encryption_algorithm 3des;
hash_algorithm sha1;
authentication_method pre_shared_key;
dh_group 2;
}
}

sainfo anonymous
{
pfs_group 2;
lifetime time 12 hour;
encryption_algorithm 3des, blowfish, des, rijndael;
authentication_algorithm hmac_sha1, hmac_md5;
compression_algorithm deflate;
}

```

**cat /usr/local/etc/racoon/psk.txt**  
**2.3.4.2 freebsd**

- Pre Shared Key yazılmış fayla karşı tərəfin IP ünvanı və açarı əlavə edilir.

**chmod 600 /usr/local/etc/racoon/psk.txt**

- Açıq faylinin oxunma və yazma yetkisini yalnız root istifadəçisi üçün veririk.

**rm /usr/local/etc/rc.d/racoon**

- Yüklənmə müddətində yaranan StartUP skriptini silirik.

```

/usr/local/etc/rc.d/racoon skriptini yaradırıq və tərkibinə aşağıdakı sətirləri əlavə edirik:
#!/bin/sh

case "$1" in

start)
if [ -x /usr/local/sbin/racoon ]; then
/usr/local/sbin/racoon -f /usr/local/etc/racoon/racoon.conf && echo -n 'racoon'
fi
;;

stop)
/usr/bin/killall racoon && echo -n ' racoon'
;;

*)
echo "Usage: `basename $0` { start | stop }"
exit 64
;;
esac

```

<code>chmod +x /usr/local/etc/rc.d/racoon</code>	- Skripti yerinə yetirən edirik.
<code>/usr/local/etc/rc.d/racoon start</code>	- Racoon-u işə salırıq.
<code>touch /var/log/racoon.log</code>	- Jurnallarımız üçün fayl yaradırıq.
<code>tail -n2 /var/log/racoon.log</code>	- Jurnal faylında uyğun sətri görməlisiniz.
<code>2015-04-15 20:41:06: INFO: IPsec-SA established: ESP/Tunnel 1.2.3.2[500]-&gt;2.3.4.2[500] spi=180542164(0xac2dad4)</code>	
<code>2015-04-15 20:41:06: DEBUG: ===</code>	
<code>setkey -DP</code>	- Şifrələnmiş paketlərin gedişinə baxırıq. - <b>D</b> - SAD (Security Association Database) tutur,

hansı ki, kernel-də Security Policy Database(SPD) kimi tanınır.

-P - SPD verilənlərini də tutur.

```
192.168.2.0/24[any] 192.168.1.0/24[any] any
    in ipsec
    esp/tunnel/2.3.4.2-1.2.3.2/require
    spid=2 seq=1 pid=775
    refcnt=1
```

```
192.168.1.0/24[any] 192.168.2.0/24[any] any
    out ipsec
    esp/tunnel/1.2.3.2-2.3.4.2/require
    spid=1 seq=0 pid=775
    refcnt=1
```

**racoonctl -ll ss isakmp**

- İlk işimiz İSAKMP SA (Security Association) və IPSec SA-nı yoxlamaqdır.

Source	Destination	Cookies	ST	S	V	E	Created	Phase2
<b>1.2.3.2.500</b>	<b>2.3.4.2.500</b>	<b>c52a5d35e497f2d8:16ef91ebfdcf4be</b>	<b>9</b>	<b>R</b>	<b>10</b>	<b>A</b>	<b>2015-04-15 20:57:23</b>	<b>1</b>

**tcpdump -n -e -i gif0**

- İkinci tərəfin client maşınınından ilk maşınınımızın client IP ünvanına ping paketlər yollayaraq, gif şəbəkə kartına qulaq asa bilərik.

```
22:25:11.599845 AF IPv4 (2), length 64: 192.168.2.1 > 192.168.1.2: ICMP echo reply, id 1, seq 21, length 40
22:25:12.613381 AF IPv4 (2), length 64: 192.168.2.1 > 192.168.1.2: ICMP echo reply, id 1, seq 22, length 40
22:25:13.613872 AF IPv4 (2), length 64: 192.168.2.1 > 192.168.1.2: ICMP echo reply, id 1, seq 23, length 40
```

## **İkinci serverin ipsec2.freebsd.lan quraşdırması**

Serverimizin şəbəkə kartları:

```
em0      - 2.3.4.2
em1      - 192.168.2.1
```

**cd /sys/amd64/conf**

- Kernel-i IPSec üçün kompilyasiya edirik.

**cp GENERIC ipsekkern**

- Kernel faylıını nüsxələyirik.

```

ipsekkern adlı kernel faylımızda uyğun dəyişiklikləri edirik.
ident ipsekkern - Identin adını dəyişib mütləq ipsekkern yazırıq.
device      crypto
device      gif
options     IPFIREWALL
options     IPFIREWALL_VERBOSE
options     IPFIREWALL_VERBOSE_LIMIT=1000
# IPSEC
options     IPSEC
options     IPSEC_DEBUG

cd ../../.. - Üç ünvan geri qayıdırıq ki, kompilyasiyanı işə
              salaq.
make buildkernel KERNCONF=ipsekkern - Kompilyasiyaya başlayırıq.
make installkernel KERNCONF=ipsekkern - Yeni kernel yükleyirik.

```

Command line ilə **GIF** (Generic tunnel interface) yaratmaq üçün aşağıdakı əmrlərdən istifadə etmək lazımdır:

```

ifconfig gif0 create
ifconfig gif0 2.3.4.2 1.2.3.2
ifconfig gif0 inet 192.168.2.1 192.168.1.1 netmask 0xffffffff
route add -inet 192.168.1.0/24 192.168.1.1

netstat -rn | grep 192.168.1.0/24 - Route cədvəlimizə baxırıq.
192.168.1.0/24      192.168.2.1      UGS      em1

```

İkinci serverimizin **/etc/rc.conf** StartUP quraşdırma faylinin məzmunu aşağıdakı kimidir:

```

ifconfig_em0="inet 2.3.4.2 netmask 255.255.255.0"
ifconfig_em1="inet 192.168.2.1 netmask 255.255.255.0"
defaultrouter="2.3.4.1"
hostname="ipsec2.freebsd.lan"
gif_interfaces="gif0"
gifconfig_gif0="2.3.4.2 1.2.3.2"
ifconfig_gif0="inet 192.168.2.1 192.168.1.1 netmask 255.255.255.0"
sshd_enable="YES"
inetd_enable="YES"

```

```
gateway_enable="YES"
firewall_enable="YES"
firewall_type="UNKNOWN"
firewall_script="/etc/firewall.conf"
static_routes="vpn"
route_vpn=" -net 192.168.1.0/24 192.168.2.1"
racoon_enable="YES"
racoon_flags="-F -f /usr/local/etc/racoon/racoon.conf -l /var/log/racoon.log"
ipsec_enable="YES"
ipsec_file="/etc/ipsec.conf"
```

İPSec quraşdırması için `/etc/ipsec.conf` quraşdırma faylinin məzmunu aşağıdakı kimi olacaq:

```
#!/sbin/setkey -f
flush;
spdflush;
```

```
spdadd 192.168.2.0/24 192.168.1.0/24 any -P out ipsec esp/tunnel/2.3.4.2-1.2.3.2/require;
spdadd 192.168.1.0/24 192.168.2.0/24 any -P in ipsec esp/tunnel/1.2.3.2-2.3.4.2/require;
```

Firewall üçün `/etc/firewall.conf` faylımiza aşağıdakı sətirləri əlavə edirik ki, sistem yenidənyüklənməsindən sonra avtomatik işə düşsün.

```
ipfw add 00001 allow ip from any to any via gif0
ipfw add 00002 allow udp from 2.3.4.2 to 1.2.3.2 isakmp
ipfw add 00003 allow udp from 1.2.3.2 to 2.3.4.2 isakmp
ipfw add 00004 allow esp from 2.3.4.2 to 2.3.4.2
ipfw add 00005 allow esp from 1.2.3.2 to 2.3.4.2
ipfw add 00006 allow ipencap from 2.3.4.2 to 1.2.3.2
ipfw add 00007 allow ipencap from 1.2.3.2 to 2.3.4.2
ipfw add 00100 allow ip from any to any via lo0
ipfw add 00200 deny ip from any to 127.0.0.0/8
ipfw add 00300 deny ip from 127.0.0.0/8 to any
```

Eyni ilə ilk serverimizdə olduğu kimi, İPSec-Tool portunu `/usr/ports/security/ipsec-tools/` ünvanından eyni modullarla yükləyirik.

VPN quraşdırma `/usr/local/etc/racoon/racoon.conf` faylımızı aşağıdaki kimi edirik:  
`path pre_shared_key "/usr/local/etc/racoon/psk.txt";`

```
listen
{
    isakmp 2.3.4.2;
}

# Debug etmek için önceden bu satırı yazırıq. İslədikdən sonra silmək olar.
log debug2;

remote 1.2.3.2
{
    exchange_mode aggressive;

    my_identifier address;

    lifetime time 24 hour;

    proposal {
        encryption_algorithm 3des;
        hash_algorithm shal;
        authentication_method pre_shared_key;
        dh_group 2;
    }
}

sainfo anonymous
{
    pfs_group 2;
    lifetime time 12 hour;
    encryption_algorithm 3des, blowfish, des, rijndael;
    authentication_algorithm hmac_shal, hmac_md5;
    compression_algorithm deflate;
}
```

```
cat /usr/local/etc/racoon/psk.txt  
1.2.3.2 freebsd
```

- Pre Shared Key yazılmış fayla qarşı tərəfin IP ünvanı və açarı əlavə edilir.

```
chmod 600 /usr/local/etc/racoon/psk.txt
```

- Açıq faylinin oxunma və yazma yetkisini yalnız root istifadəçisi üçün veririk.

```
rm /usr/local/etc/rc.d/racoon
```

- Yüklənmə müddətində yaranan StartUP skriptini silirik.

/usr/local/etc/rc.d/racoon skriptini yaradırıq və tərkibinə aşağıdakı sətirləri əlavə edirik:  
#!/bin/sh

```
case "$1" in  
  
start)  
if [ -x /usr/local/sbin/racoon ]; then  
/usr/local/sbin/racoon -f /usr/local/etc/racoon/racoon.conf && echo -n '  
racoon'  
fi  
;;  
  
stop)  
/usr/bin/killall racoon && echo -n ' racoon'  
;;  
*)  
echo "Usage: `basename $0` { start | stop }"  
exit 64  
;;  
esac
```

```
chmod +x /usr/local/etc/rc.d/racoon
```

- Skripti yerinə yetirən edirik.

```
/usr/local/etc/rc.d/racoon start
```

- Racoon-u işə salırıq.

```
touch /var/log/racoon.log
```

- Jurnallarımız üçün fayl yaradırıq.

```
tail -n2 /var/log/racoon.log           - Jurnal faylında uygun sətri görməlisiniz.  
2015-04-15 20:41:06: INFO: IPsec-SA established: ESP/Tunnel 2.3.4.2[500]-  
>1.2.3.2[500] spi=36251977(0x2292949)  
2015-04-15 20:41:06: DEBUG: ===
```

```
setkey -DP           - Şifrələnmiş paketlərin gedişinə baxırıq.  
192.168.1.0/24[any] 192.168.2.0/24[any] any  
    in ipsec  
    esp/tunnel/1.2.3.2-2.3.4.2/require  
    spid=2 seq=1 pid=728  
    refcnt=1  
192.168.2.0/24[any] 192.168.1.0/24[any] any  
    out ipsec  
    esp/tunnel/2.3.4.2-1.2.3.2/require  
    spid=1 seq=0 pid=728  
    refcnt=1
```

```
racoonctl -ll ss isakmp           - İlk işimiz İSAKMP SA (Security Association) və  
                                IPSec SA-nı yoxlamaqdır.  
Source      Destination Cookies          ST S V E   Created      Phase2  
2.3.4.2.500 1.2.3.2.500  c52a5d35e497f2d8:16ef91ebfdcf4be 9 I 10 A 2015-04-15 20:57:22 1
```

```
racoonctl ss esp           - SA və ESP-ni çap edəcək.
```

# **FreeBSD ilə Cisco arasında IPSec vasitəsilə Site-to-Site VPN qurulması**

Məqsədimiz FreeBSD 10.1 x64 server ilə Cisco 7200 Router arasında IPSec tunel yaratmaqdır və bu tunel üzərindən daxili şəbəkədə olan istifadəçilərin trafikini şifrələyərək bir-birlərinə tanıtmaqdır. Gördüyüümüz bütün testlər GNS3 və VMWare Workstation vasitəsilə edilmişdir. Cisco 7200 routerimizi GNS3 virtual cloud vasitəsilə Windows məşinimizin LoopBack virtual şəbəkə kartına bridge etmişik. Həmçinin VMWare Workstation virtual mühitimizdə 4 ədəd məşin mövcuddur. İki ədəd **Windows7**, **FreeBSD PF-NAT** və **FreeBSD İPsec**. Windows7 desktop-un biri FreeBSD İPsec serverin istifadəçisidir, hansı ki, sayesində tunellə cisco tərəfi görməlidir. Digər Windows7 desktop isə Cisco 7200 Router-in istifadəçisidir. FreeBSD PF-NAT serveri isə Router və NAT server rolunu oynayır və Cisco 7200 ilə FreeBSD İPsec server üçün Gateway-dir. IPSec tunel uğurlu qurulduqdan sonra Windows məşinlər bir-birlərini görməlidirlər.

FreeBSD məşinə şəbəkə kartları və İP ünvanları aşağıdakı kimiidir:

em0 - 1.2.3.2

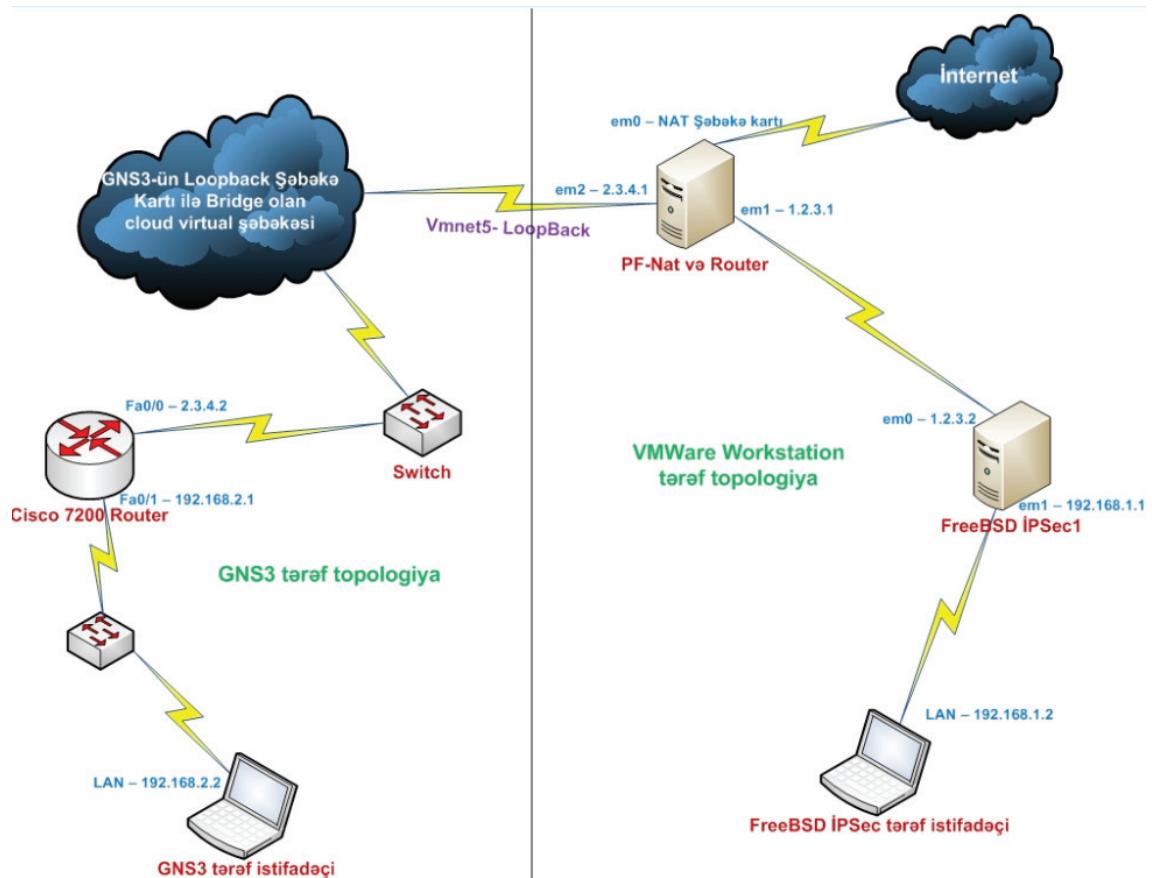
em1 - 192.168.1.1

Cisco 7200 Router-də olan şəbəkə kartları aşağıdakı kimiidir:

Fa0/0 - 2.3.4.2

Fa0/1 - 192.168.2.1

Şəbəkə quruluşumuz şəkildəki kimi olacaq:



### FreeBSD maşınının hazırlanması

Öncəki başlığımızda, yəni "FreeBSD IPSec Site-to-Site VPN" qurulmasında portlardan tələb edilən üçüncü tərəf program təminatını, yəni **ipsec-tools** FreeBSD IPSec serverimizə yükləyirik. Həmçinin kernel-imizə tələb edilən imkanları əlavə edib kompilyasiya edirik. **/etc/rc.conf** quraşdırma faylımız aşağıdakı kimi olacaq:

```
hostname="ipsecl.freebsd.lan"
sshd_enable="YES"
ifconfig_em0="inet 1.2.3.2/24"
ifconfig_em1="inet 192.168.1.1/24"
defaultrouter="1.2.3.1"
firewall_enable="YES"
```

```

firewall_type="OPEN"
gateway_enable="YES"

gif_interfaces="gif0"
gifconfig_gif0="1.2.3.2 2.3.4.2"
ifconfig_gif0="inet 192.168.1.1 192.168.2.1 netmask 255.255.255.0 mtu 1280"
static_routes="vpn"
route_vpn="-net 192.168.2.0/24 192.168.1.1"
racoon_enable="YES"
racoon_flags="-F -f /usr/local/etc/racoon/racoon.conf -l /var/log/racoon.log"
ipsec_enable="YES"
ipsec_file="/etc/ipsec.conf"

```

`/usr/local/etc/racoon/racoon.conf` quraşdırma faylımızın məzmunu aşağıdakı kimi olacaq:

```

path include "/usr/local/etc/racoon" ;
path pre_shared_key "/usr/local/etc/racoon/psk.txt" ;
log debug2;

padding {
    maximum_length 20;
    randomize off;
    strict_check off;
    exclusive_tail off;
}
listen {
    isakmp 1.2.3.2 [500]; # Öz IP ünvanımız, hansında ki, qulaq asacaq.
}

```

# Susmaya görə olan vaxt aralıqlarını təyin edirik.

```

timer {
    # Bu mənə hər bir uzaq maşın üçün dəyişdirilə bilər.
    counter 5;          # Göndərilmə cəhdinin maksimal sayı.
    interval 20 sec;    # Yenidən göndərilmə arasında olan maksimal vaxt.
    persend 1;          # Hər bir göndəriş üçün olan paket sayı.
    # Hər bir fazonın bitməsi üçün təyin edilən gözləmə vaxtı.
}

```

```

    phase1 30 sec;
    phase2 15 sec;
}

# İlk faza üçün quraşdılmalarımızı təyin edirik.
remote 2.3.4.2      # Uzaq tərəfin dünya IP ünvani.
{
    exchange_mode main;          # Hər iki fazada aqressiv razılışmanı
                                    # bütün IOS-lar dəstəkləmir.
    doi ipsec_doi;
    situation identity_only;
    nonce_size 16;
    lifetime time 60 min;
    initial_contact on;
    support_proxy on;
    proposal_check obey;
    proposal {
        encryption_algorithm 3des;  # Şifrlənmə alqoritmi.
        hash_algorithm sha1;       # Şifrlənmə alqoritmi.
        authentication_method pre_shared_key;
                                    # Autentifikasiya metodu
                                    # ümumi açardadır.
        dh_group 2;              #Diffie Hellman açarının uzunluğu
                                    # (2-ci qrup - 1024 bitdir).
    }
}
}

# Misalımız onu göstərir ki, 192.168.1.0/24 şəbəkəsindən 192.168.2.0/24
# şəbəkəsinə gələn paketlər emal olunmaya gedir.
# İlk şəbəkə öz şəbəkəmiz, ikinci isə uzaq şəbəkədir.
# any 'istənilən port' mənasını verir.
sainfo subnet 192.168.1.0/24 any address 192.168.2.0/24 any {
    pfs_group 2;
    lifetime time 24 hour;
    encryption_algorithm aes;
    authentication_algorithm hmac_shal;
    compression_algorithm deflate;
}

```

Şifrələnmə faylımız **/usr/local/etc/raccoon/psk.txt** tərkibi aşağıdakı kimi olacaq  
(Pre-Shared-Key: **freebsdcisco**):

#### 2.3.4.2 freebsdcisco

**/etc/ipsec.conf** faylımızın tərkibi isə aşağıdakı kimi olacaq:

```
#!/sbin/setkey -f
```

```
flush;
```

```
spdflush;
```

```
# 10.50.3.0/24 şəbəkəsindən çıxan ESP trafiki üçün qayda.
```

```
spdadd 192.168.1.0/24 192.168.2.0/24 any -P out ipsec esp/tunnel/1.2.3.2-2.3.4.2/require;
```

```
# 10.50.3.0/24 şəbəkəsinə daxil olan ESP trafiki üçün qayda.
```

```
spdadd 192.168.2.0/24 192.168.1.0/24 any -P in ipsec esp/tunnel/2.3.4.2-1.2.3.2/require;
```

Aşağıdakı addımları Cisco 7200 Router-də edilən quraşdırılmalardan sonra etmək lazımdır:

```
ping -c2 192.168.2.1
```

- Cisco Router-imizdə olan Virtual LoopBack şəbəkə kartının IP ünvanına 2 ədəd ping paket yollayıb cavab alırıq.

```
PING 192.168.2.1 (192.168.2.1): 56 data bytes
```

```
64 bytes from 192.168.2.1: icmp_seq=0 ttl=255 time=67.893 ms
```

```
64 bytes from 192.168.2.1: icmp_seq=1 ttl=255 time=78.170 ms
```

```
setkey -D
```

- SAD verilənlərinə baxırıq.

```
1.2.3.2 2.3.4.2
```

```
esp mode=tunnel spi=3943473247(0xeb0ca05f) reqid=0(0x00000000)
E: rijndael-cbc 726f0816 2902674d f7d0efc8 185ff67c
A: hmac-sha1 3940debf 5b8581af 8ed1c138 3a8a66fc 9fe0a1b2
seq=0x00000004 replay=4 flags=0x00000000 state=mature
created: Apr 16 15:18:27 2015 current: Apr 16 15:22:12 2015
diff: 225(s) hard: 86400(s) soft: 69120(s)
last: Apr 16 15:19:35 2015 hard: 0(s) soft: 0(s)
current: 608(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 4 hard: 0 soft: 0
sadb_seq=1 pid=717 refcnt=2
```

### **2.3.4.2 1.2.3.2**

```
esp mode=tunnel spi=184175989(0x0afa4d75) reqid=0(0x00000000)
E: rijndael-cbc 7135900c ca645fc9 ac996849 15f832b1
A: hmac-shal 869e43e0 1bcl5d11 6fdcee65 c28ee5cd 961cf9d0
seq=0x00000004 replay=4 flags=0x00000000 state=mature
created: Apr 16 15:18:27 2015 current: Apr 16 15:22:12 2015
diff: 225(s) hard: 86400(s) soft: 69120(s)
last: Apr 16 15:19:35 2015 hard: 0(s) soft: 0(s)
current: 416(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 4 hard: 0 soft: 0
sadb_seq=0 pid=717 refcnt=1
```

**root@ipsecl:~ # racoonctl -ll ss isakmp** - İSAKMP SA (Security Association) və  
IPSec SA-nı yoxlayırıq.

Source	Destination	Cookies	ST	S	V	E	Created	Phase2
<b>1.2.3.2.500</b>	<b>2.3.4.2.500</b>	<b>a93ce768a8bf bffa:3b380ab741f1430b</b>	<b>9</b>	<b>I</b>	<b>10</b>	<b>M</b>	<b>2015-04-16 15:18:27</b>	<b>1</b>

**tcpdump -n -e -i gif0** - Virtual gif0 şəbəkə kartımızda olan trafikə baxırıq.  
15:28:13.655266 AF IPv4 (2), length 104: 192.168.2.1 > 192.168.1.2: ICMP echo request, id 24, seq 0, length 80  
15:28:13.717923 AF IPv4 (2), length 104: 192.168.2.1 > 192.168.1.2: ICMP echo request, id 24, seq 1, length 80  
15:28:13.796035 AF IPv4 (2), length 104: 192.168.2.1 > 192.168.1.2: ICMP echo request, id 24, seq 2, length 80  
15:28:13.858517 AF IPv4 (2), length 104: 192.168.2.1 > 192.168.1.2: ICMP echo request, id 24, seq 3, length 80  
15:28:13.966621 AF IPv4 (2), length 104: 192.168.2.1 > 192.168.1.2: ICMP echo request, id 24, seq 4, length 80

#### Serverin təhlükəsizliyi

VPN quraşdırımlarından sonra bu işi görmək mütləqdir.

Aşağıdakı sətirləri **/etc/sysctl.conf** quraşdırma faylinə əlavə edirik:

```
# "Qara dəlik" adlanan quraşdırmanın edirik ki,  
# bağlı olan portlara gələn paketlərin skan etmə prosesini çətinləşdirir.  
net.inet.tcp.blackhole=2  
net.inet.udp.blackhole=1
```

```
# IP paket üçün təsadüfi ID generasiya edirik ki, fingerprint imkanı olmasın  
# və hücumçunun paketləri sistemləşdirmə imkanı çətinləşsin.
```

```
net.inet.ip.random_id=1  
net.inet.icmp.maskrepl=0
```

```
# Socket növbəsi həcmi təyin edirik.  
kern.ipc.somaxconn=1024  
  
# IP redirekti dayandırırıq.  
net.inet.icmp.drop_redirect=1  
net.inet.icmp.log_redirect=1  
net.inet.ip.redirect=0  
net.inet6.ip6.redirect=0  
  
# TCP qoşulmaların bufer həcmi təyin edirik.  
net.inet.tcp.sendspace=32768  
net.inet.tcp.recvspace=32768  
  
# ARP cədvəlini hər 20 dəqiqədən bir yeniləyirik.  
net.link.ether.inet.max_age=1200  
  
# Bütün aidiyyəti olmayan müraciətlərə cavab vermənin qarşısını alırıq.  
net.inet.ip.sourceroute=0  
net.inet.ip.accept_sourceroute=0  
net.inet.icmp.bmcastecho=0  
net.inet.icmp.maskrepl=0  
  
# Adı istifadəçilərə digər UID-lə açılmış proseslərə baxışa qadağa  
# təyin edirik.  
security.bsd.see_other_uids=0
```

Aşağıdakı sətirləri **/etc/rc.conf** faylına əlavə edirik:  
# Sistemin hər yüklenməsində tmp qovluğununu təmizləyirik.  
**clear\_tmp\_enable="YES"**

```
# Əgər sistemdə NFS istifadəsi planlaşdırılmışsa,  
# onu və portmap daemonu dayandırırıq.  
nfs_server_enable="NO"  
nfs_client_enable="NO"  
portmap_enable="NO"  
icmp_bmcastecho="NO"
```

```

fsck_y_enable="YES"
log_in_vain="YES"

# Susmaya görə sistemin təhlükəsizlik səviyyəsi -1 olur(kernel level).
# Bu, az qorunan deməkdir.
# Əgər müdafiə olunan səviyyə istifadə etmək lazımlı olarsa,
# onda 2 və ya daha güclü qorunan olan 3 təyin etməniz daha düzgün olar.
kern_securelevel_enable="YES"
kern_securelevel="2"

# ICMP redirect mesajları hücum üçün istifadə edilə bilər.
# Ona görə də onlara məhəl qoymuruq.
icmp_drop_redirect="YES"
icmp_log_redirect="YES"

# Sistemə giriş etdikdə konsol-a kernel versiyası və
# sistem versiyası çıxmasının qarşısını alırıq (Message of Day).
update_motd="NO"

# Aşağıdakı kernel opsiyası FreeBSD Router, firewall olduqda yararlı olur.
# Sistemə SYN/FIN paketlərini qəbul etməməsi üçün quraşdırırıq.
tcp_drop_synfin="YES"

```

### Cisco 7200 Router hazırlanması

İlk işimiz **Fa0/0** şəbəkə kartını quraşdırıb IP ünvan təyin etmək, susmaya görə olan yol və route yazmaqdır. Sonra isə VPN quraşdırılmalarımıza keçirik.

```

R1#conf t
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 2.3.4.2 255.255.255.0
R1(config-if)#crypto map MY_MAP

```

- Bu sintaksis MY\_MAP adlı CryptoMap yaratıldıqdan sonra işləyəcək. Ona görə bu sətri VPN quraşdırmasından sonra etməlisiniz.

```

R1(config)#ip default-gateway 2.3.4.1
R1(config)#ip route 0.0.0.0 0.0.0.0 2.3.4.1

```

- Susmaya görə olan yol sətri işləmədiyinə görə  
 - Route-nu statik daxil edin.

**isakmp** siyaseti. Qarşı tərəfdən gələn cavabların uyğunluğu **/usr/local/etc/racoon/racoon.conf** faylında **sainfo** başlığında göstərilmişdir:

```
R1(config)#crypto isakmp policy 20
R1(config-isakmp)#encr 3des
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 2
R1(config-isakmp)#lifetime 3600
R1(config-isakmp)#crypto isakmp identity address
R1(config)#crypto isakmp key freebsdCisco address 1.2.3.2
```

**transform-set** yaradırıq:

```
R1(config)#crypto ipsec transform-set MY_MAP esp-aes esp-sha-hmac
```

**CryptoMap** yaradırıq:

```
R1(config)#crypto map MY_MAP 1 ipsec-isakmp
R1(config-crypto-map)#set peer 1.2.3.2
R1(config-crypto-map)#set security-association lifetime seconds 86400
R1(config-crypto-map)#set transform-set MY_MAP
R1(config-crypto-map)#set pfs group2
R1(config-crypto-map)#match address 150
```

Şifrələnməli olan "maraqlı" trafiki açıqlayıraq. crypto ACL kontekstində olan "**permit**" bildirir ki, "şifrələ", ancaq "**deny**" isə "şifrələmə":

```
R1(config)# access-list 150 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
```

Həmçinin CryptoMap-i şəbəkə kartına təyin edirik:

```
R1(config)#interface fastethernet0/0
R1(config-if)#crypto map MY_MAP
```

Test edə bilməyimiz üçün virtual şəbəkə kartı yaradıb IP ünvan təyin edirik:

**Qeyd:** Sizin halda LoopBack yox, ikinci şəbəkə kartı olacaq və onu quraşdırıb, Cisco Router tərəfdə olan istifadəçilərə susmaya görə olan yol kimi təyin etməlisiniz.

```
R1(config)#interface Loopback0
R1(config-if)#ip address 192.168.2.1 255.255.255.0
```

FreeBSD IPsec tərəfdə olan Windows maşına ping ataraq yoxlanış edirik:

```
R1(config)#do ping 192.168.1.2 source 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.2.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/77/124 ms
```

#### Quraşdırımların yoxlanılması

İllk ping paketdən sonra həmin anda da IKE1 fazası işə düşməlidir. Cisco tərəfdən onun statusuna aşağıdakı əmr ilə baxırıq:

```
R1#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state      conn-id status
2.3.4.2      1.2.3.2    QM_IDLE   1002 ACTIVE
```

Əgər IKE1 uğurla qoşulmuşdursa, IPSec SA yoxlayırıq:

```
R1#show crypto ipsec sa interface fastEthernet 0/0
interface: FastEthernet0/0
  Crypto map tag: MY_MAP, local addr 2.3.4.2

  protected vrf: {none}
  local  ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
  current_peer 1.2.3.2 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 134, #pkts encrypt: 134, #pkts digest: 134
    #pkts decaps: 232, #pkts decrypt: 232, #pkts verify: 232
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

  local crypto endpt.: 2.3.4.2, remote crypto endpt.: 1.2.3.2
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
  current outbound spi: 0xAFA4D75(184175989)
```

```
PFS (Y/N): Y, DH group: group2

inbound esp sas:
spi: 0xEBOCA05F(3943473247)
transform: esp-aes esp-sha-hmac ,
in use settings ={Tunnel, }
conn id: 3, flow_id: SW:3, sibling_flags 80000040, crypto map: MY_MAP
sa timing: remaining key lifetime (k/sec): (4299128/84080)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xAFA4D75(184175989)
transform: esp-aes esp-sha-hmac ,
in use settings ={Tunnel, }
conn id: 4, flow_id: SW:4, sibling_flags 80000040, crypto map: MY_MAP
sa timing: remaining key lifetime (k/sec): (4299128/84080)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```

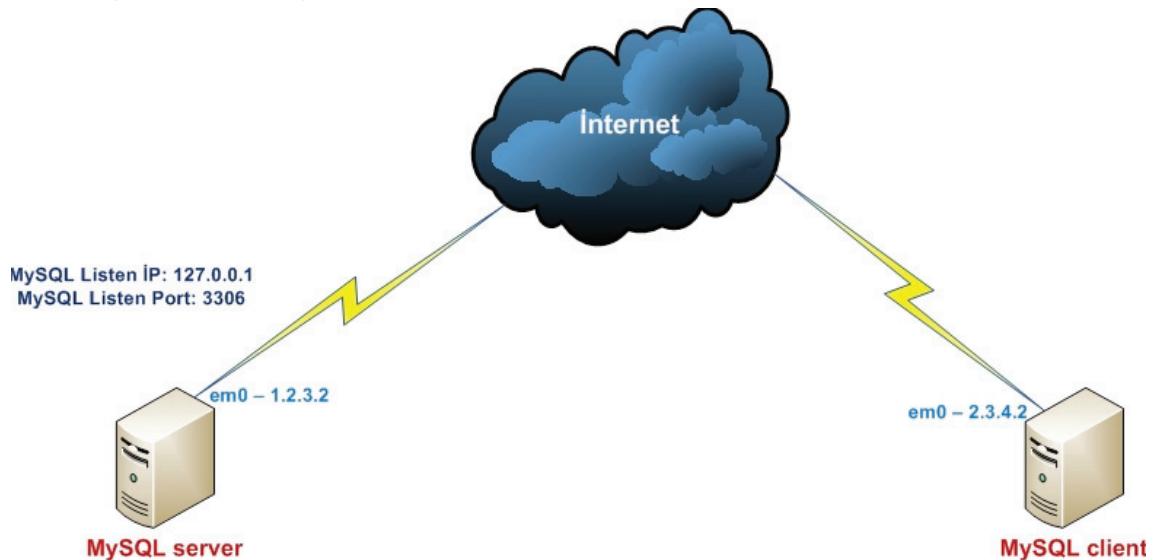
# FreeBSD Stunnel

Bizdən tələb oluna bilər ki, çox önəmli data-nın ötürülməcəyi servisin trafikini şəbəkədə istifadə edərkən şifrələmək lazımdır. Misal olaraq, hansısa PUBLİC İP ünvanda təyin edilmiş müəyyən bir portun üzərində ancaq bir servis işləyir, servis yazılıarkən trafikin şifrələnib ötürülməsi programçı tərəfindən nəzərə alınmamışdır və istismarda olan xidməti dayandırmadan yenə də trafiki şifrələmək mütləq şərtidir. Bunu test etmək üçün başqa bir misal MySQL-i çəkə bilerik. MySQL susmaya görə 3306-ci portda qulaq asır və müraciətləri **cleartext** kimi qəbul edir. İki ədəd serverimiz mövcuddur. Onlar PUBLİC-dədirlər. Əsas serverimizdə MySQL 127.0.0.1 İP üzərində 3306-ci port-da qulaq asır və digər serverimizdə isə MySQL client şifrələnmiş kanalla bu İP ünvan və porta qoşulmalıdır.

Serverlərimizin adları aşağıdakı kimidir:

**stunnelmysqlserver**  
**stunnelmysqlclient**

Şəbəkə quruluşumuz aşağıdakı şəkildəki kimidir:



## **Stunnel MySQL serverimizin qurulması**

```
portsnap fetch extract update
```

- Sistemimizin portlarını yeniléyirik.

Portlardan Stunnel və MySQL server-i yükleyirik.

```
root@stunnelmysqlserver:/ # cd /usr/ports/security/stunnel  
root@stunnelmysqlserver:/usr/ports/security/stunnel # make config
```

```
root@stunnelmysqlserver:/usr/ports/security/stunnel # make install
```

```
root@stunnelmysqlserver:/ # cd /usr/ports/databases/mysql55-server/  
root@stunnelmysqlserver:/usr/ports/databases/mysql55-server # make config
```

```
root@stunnelmysqlserver:/usr/ports/databases/mysql55-server # make install
```

MySQL serverimizi quraşdırırıq:

```
cp /usr/local/share/mysql/my-large.cnf /etc/my.cnf
```

- Quraşdırma faylını nüsxələyirik.

**/etc/my.cnf** faylına aşağıdakı sətirləri əlavə edirik ki, MySQL **127.0.0.1** İP ünvanında dinləsin və jurnalları **/var/log/mysql.log** faylına yazsın.

```
[mysqld]
log = /var/log/mysql.log
bind-address = 127.0.0.1
```

**chown mysql:mysql /var/log/mysql.log** - Jurnalların yazılı bilməsi üçün fayla yetki veririk.

**echo 'mysql\_enable="YES'" >> /etc/rc.conf** - MySQL daemon-u StartUP-a əlavə edirik.  
**/usr/local/etc/rc.d/mysql-server start** - MySQL serverini işə salırıq.

**/usr/local/bin/mysql\_secure\_installation** - MySQLi quraşdırırıq və şifrə təyin edirik.

Change the root password? [Y/n] **Y**

New password: **şifre**

Re-enter new password: **şifre \_ təkrar**

Remove anonymous users? [Y/n] **Y**

Disallow root login remotely? [Y/n] **Y**

Remove test database and access to it? [Y/n] **Y**

Reload privilege tables now? [Y/n] **Y**

Uzaqdan daxil olub istifadə etmək üçün baza yaradaq və həmin bazaya istənilən İP ünvandan lazımi istifadəçi ilə daxil olmağa izin verək. Bu, bize Stunnel-i test etmək üçün lazım olacaq:

**mysql -uroot -pşifre** - MySQL konsola daxil oluruq.

Istənilən İP ünvandan qoşula bilməsi üçün cavid adlı MySQL istifadəçisini yaradırıq.

**mysql> CREATE USER 'cavid'@'%' IDENTIFIED BY 'freebsd';**

Test etmək üçün Stunnel adlı baza yaradırıq.

**mysql> CREATE DATABASE stunnel;**

**stunnel** bazasının istənilən cədvəlinə **cavid** adlı istifadəçi adı ilə istənilən İP ünvandan girişinə izin veririk.

**mysql> GRANT ALL PRIVILEGES ON stunnel.\* TO 'cavid'@'%';**

#### Stunnel-in quraşdırılması

```
cd /usr/local/etc/stunnel
mkdir certs ; cd certs
```

- Stunnel quraşdırma qovluğuna daxil oluruq.  
- Sertifikat üçün qovluq yaradırıq və ora daxil oluruq.

MySQL-ə təhlükəsiz kanalla qoşula bilməsi üçün 1 cüt açar generasiya edirik.

```
openssl req -new -x509 -days 365 -nodes -out mysql.cert -keyout mysql.key
```

Generating a 1024 bit RSA private key

.....+++++

.....+++++

writing new private key to 'mysql.key'

----

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

----

Country Name (2 letter code) [AU]:**AZ**

State or Province Name (full name) [Some-State]:**Baku**

Locality Name (eg, city) []:**Narimanov**

Organization Name (eg, company) [Internet Widgits Pty Ltd]:**ATLGroup**

Organizational Unit Name (eg, section) []:**ATL**

Common Name (e.g. server FQDN or YOUR name) []:**atl.az**

Email Address []:**cavid@gmail.com**

Sertifikat qoşluğunun bütövlükdə MySQL client işləyən serverimizə nüsxələyirik.

```
root@stunnelmysqlserver:~ # scp -r /usr/local/etc/stunnel/certs/
```

```
root@2.3.4.2:/usr/local/etc/stunnel/
```

**/usr/local/etc/stunnel/stunnel.conf** quraşdırma faylına aşağıdakı sətirləri əlavə edirik(Sertifikat və açarın ünvanını təyin edirik):

```
cert = /usr/local/etc/stunnel/certs/mysql.cert
```

```
key = /usr/local/etc/stunnel/certs/mysql.key
```

```
chroot = /var/tmp/lib/stunnel
```

# PİD faylı ünvanı mütləq aşağıdakı kimi təyin edilməlidir, çünki

# PİD faylı üçün **/var/tmp/lib/stunnel** qoşluğununa müraciət ediləcək.

```

pid = /stunnel.pid
setuid = stunnel
setgid = stunnel
debug = 7

# Jurnal faylı ünvanı mütləq aşağıdakı kimi təyin edilməlidir, çünki
# jurnal faylı üçün /var/tmp/lib/stunnel qovluğuna müraciət ediləcək.
output = /stunnel.log

# Aşağıdakı sətirlərlə deyirik ki, serverimizin 1.2.3.2 IP ünvanına 3307-ci
# porta müraciət gələrsə, onu 127.0.0.1 IP ünvanında 3306-ci porta yönəldir.
[mysqls]
accept = 1.2.3.2:3307
connect = 127.0.0.1:3306

```

**mkdir -p /var/tmp/lib/stunnel** - Stunnel Chroot qovluq yaradırıq.

**touch /var/tmp/lib/stunnel/stunnel.log** - Jurnal faylini yaradırıq.

**chown -R stunnel:stunnel /var/tmp/lib/stunnel**

- Qovluğu Stunnel istifadəçi adı və qrupuna mənimsedirik.

**chmod -R 700 /var/tmp/lib/stunnel/**

- Chroot qovluğuna yalnız Stunnel istifadəçi üçün tam yetki veririk.

**/etc/rc.conf** StartUP faylinə aşağıdakı sətirləri əlavə edirik ki, Stunnel sistem yenidənyüklənməsində avtomatik işə düşsün.

```

stunnel_enable="YES"
stunnel_pidfile="/var/tmp/lib/stunnel/stunnel.pid"

```

**/usr/local/etc/rc.d/stunnel start**

- Stunnel daemon-u işə salırıq.

## **Stunnel MySQL client-imizin qurulması**

```
portsnap fetch extract update
```

- Serverimizin portlarını yeniləyirik.

İlk isimiz MySQL için client programın yüklenməsidir.

```
root@stunnelmysqlclient:~ # cd /usr/ports/databases/mysql55-client/  
root@stunnelmysqlclient:/usr/ports/databases/mysql55-client # make config
```

```
root@stunnelmysqlclient:/usr/ports/databases/mysql55-client # make install
```

Sonra eyni<sup>de</sup> portlardan Stunnel-i yükleyirik.

```
root@stunnelmysqlserver:/ # cd /usr/ports/security/stunnel
```

```
root@stunnelmysqlserver:/usr/ports/security/stunnel # make config
```

```
root@stunnelmysqlserver:/usr/ports/security/stunnel # make install
```

Eyniła Stunnel-i client olaraq qurasdırırıq

```
cd /usr/local/etc/stunnel
```

- Stunnel gurasdırma görevi üçün daxil oluruz

```
mkdir certs ; cd certs
```

- Sertifikat üçün geyləq varadırıq və orq daxil olurıq

```
scp -r root@1.2.3.2:/usr/local/etc/stunnel/certs/
```

- Certs qovluğunu uzaq serverdən yerləşdiyimiz serverin certs qovluğuna nüsxələvirik

```

/usr/local/etc/stunnel/stunnel.conf quraşdırma faylına aşağıdakı sətirləri əlavə edirik:
# Sertifikat və açarın ünvanını təyin edirik.
# Chroot qovluğu təyin edirik.
cert = /usr/local/etc/stunnel/certs/mysql.cert
key = /usr/local/etc/stunnel/certs/mysql.key
chroot = /var/tmp/lib/stunnel/

# PID faylı ünvanı mütləq aşağıdakı kimi təyin edilməlidir, çünkü
# PID faylı üçün /var/tmp/lib/stunnel qovluğuna müraciət ediləcək.
pid = /stunnel.pid
setuid = stunnel
setgid = stunnel

# Jurnal səviyyəsini ən yüksək təyin edirik.
debug = 7

# Jurnal faylı ünvanı mütləq aşağıdakı kimi təyin edilməlidir, çünkü
# jurnal faylı üçün /var/tmp/lib/stunnel qovluğuna müraciət ediləcək.
output = /stunnel.log

# Client rejimini təyin edirik.
client = yes

# Aşağıdakı sətirlərlə deyirik ki, serverimizin 127.0.0.1 İP ünvanında
# 3307-ci portuna müraciət gələrsə,
# müraciəti 1.2.3.2 İP ünvanlı serverin 3307-ci portuna yönləndir.
[mysqls]
accept = 127.0.0.1:3307
connect = 1.2.3.2:3307

```

Lazım olan Chroot qovluq, jurnal faylıının yaradılması və yetkilərin verilməsini client maşınımızda da edirik.

**mkdir -p /var/tmp/lib/stunnel** - Stunnel Chroot qovluq yaradırıq.

**touch /var/tmp/lib/stunnel/stunnel.log** - Jurnal faylını yaradırıq.

```
chown -R stunnel:stunnel /var/tmp/lib/stunnel  
- Qovluğunu Stunnel istifadəçi adı və qrupuna  
mənimsədirik.
```

```
chmod -R 700 /var/tmp/lib/stunnel/  
- Chroot qovluğuna yalnız Stunnel istifadəçi üçün  
tam yetki veririk.
```

**/etc/rc.conf** StartUP faylına aşağıdakı sətirləri əlavə edirik ki, Stunnel sistem yenidənyüklənməsində avtomatik işə düşsün.

```
stunnel_enable="YES"  
stunnel_pidfile="/var/tmp/lib/stunnel/stunnel.pid"
```

```
/usr/local/etc/rc.d/stunnel start  
- Stunnel daemon-u işə salırıq.
```

Sonda MySQL serverimizə cavid adlı istifadəçi ilə qoşulub test edirik:

```
root@stunnelmysqlclient:~ # mysql -h 127.0.0.1 -P 3307 -u jamal -p  
Enter password: sifre  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| stunnel |  
+-----+  
2 rows in set (0.00 sec)
```

# FreeBSD HAST Cluster

Məqsədimiz FreeBSD əməliyyat sistemi üzərində olan diskimizin avtomatik olaraq digər FreeBSD serverin diskinə sinxronizasiya edilməsidir. Görüləcək işlər FreeBSD 10.1 x64 üzərində olacaq. İki ədəd node-muz var və onların üzərində identik resurslarımız mövcuddur. Aşağıda isə həmin resursları siyahılıyırıq.

**Node1**-də bütün şəbəkə kartlarının IP ünvan ardıcılılığı sonda **10** rəqəmi ilə bitir. **Node2**-də isə bütün şəbəkə kartlarının IP adres ardıcılığı sonda **20** rəqəmi ilə bitir. VIP-lər isə hər yerdə **100** rəqəmi ilə bitir.

**da0** - Sistem diskı.

**dal** - Hast istifadə edəcəyi fiziki disk.

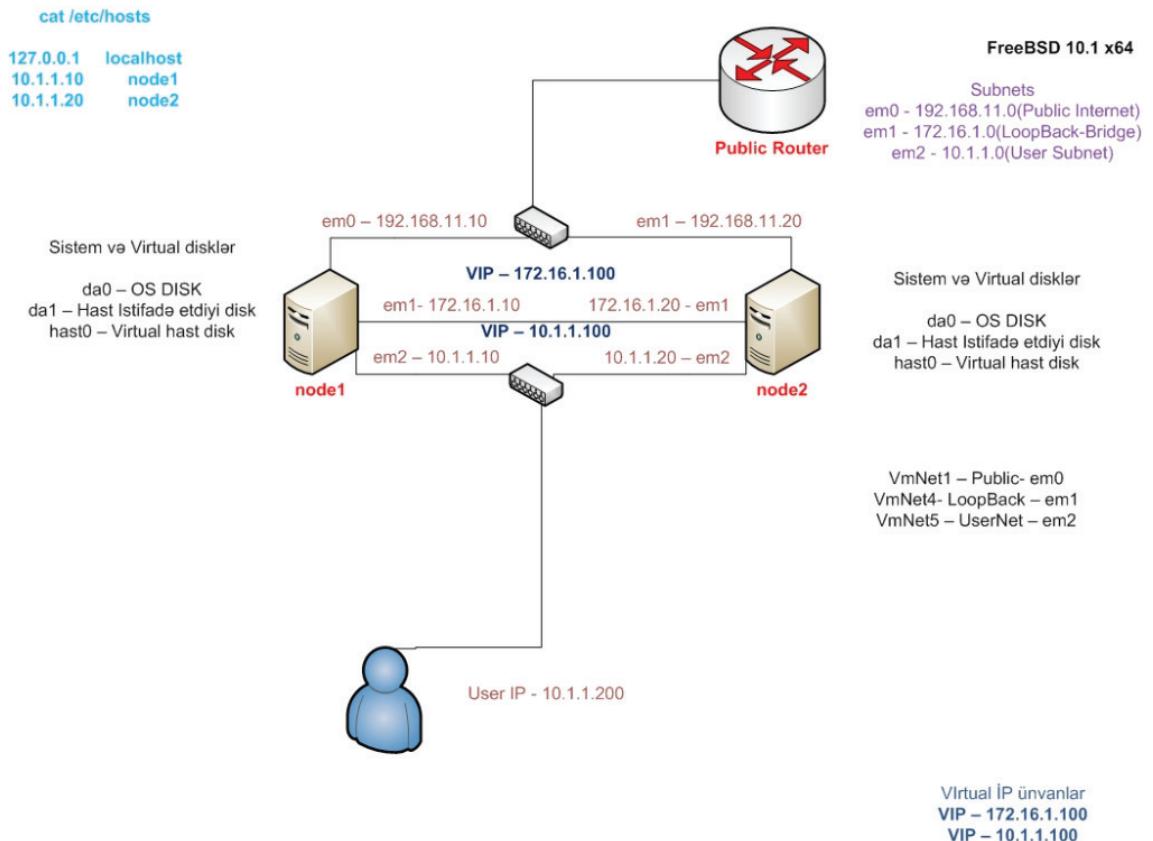
**hast0** - Hər iki serverdə hast tərəfindən yaradılıb özünə dal adlı fiziki diskini mənimseyən Virtual Diskin adı.

**em0** - Public Net(192.168.11.0) VmNet1

**em1** - VIP-Carp(172.16.1.0) VmNet4-LoopBack.

**em2** - HastSyncronization(10.1.1.0) VmNet5.

Şəbəkə quruluşu aşağıdakı şəkildəki kimi olacaq:



Əvvəlcə hər iki node-da yüklənmə bitdikdən sonra şəbəkəni quraşdırıb, lazımsız servisləri söndürürük və gələcək servislər üçün sətirlərimizi hazır edirik.

#### **node1-ucun network ve startup script**

```
cat /etc/rc.conf
```

```
hostname="node1"
ifconfig_em0="inet 192.168.11.10 netmask 255.255.255.0"
ifconfig_em1="inet 172.16.1.10 netmask 255.255.255.0"
ifconfig_em1_alias0="vhid 100 pass freebsd advskew 10 alias 172.16.1.100/32"
ifconfig_em2="inet 10.1.1.10 netmask 255.255.255.0"
defaultrouter="192.168.11.1"
sshd_enable="YES"
dumpdev="NO"
```

```

# Firewall
#pf_enable="YES"                                - Kernel kompilyasyasından sonra işləyəcək.
#pf_rules="/etc/pf.conf"
#pflog_enable="YES"
#pflog_logfile="/var/log/pflog"

# Disabled Services
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_rebuild_aliases="NO"
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"
icmp_drop_redirects="YES"

# Services
#hastd_enable="YES"                            - Kernel-ə 'options GEOM_GATE' opsiyasını
                                                elavə etməyi unutmayın.
#hastd_program="/sbin/hastd"                  - HAST-in daemon faylı.
#hastd_flags="-d -d -c /etc/hast.conf -P /var/run/hastd.pid" - HAST-in config və PID faylı.
#ucarp_enable="YES"                            - Portlardan '/usr/ports/net/ucarp'-i
                                                yükledikdən sonra aktivləşdirəcəyik.
#ucarp_addr="10.1.1.100"                      - Cluster-in Virtual IP ünvanı.
#ucarp_if="em2"                                - İstifadə edilən şəbəkə kartı.
#ucarp_src="10.1.1.10"                        - node-nun IP ünvanı.
#ucarp_pass="freebsd"                         - node-lar arası identifikasiya şifrəsi.
#ucarp_vhid=42                                 - Cluster-in unikal identifikasiatoru.
#ucarp_upscript="/usr/local/sbin/ucarp-up"    - Statusun "passivdən>aktivə" keçidində işə
                                                düşəcək skript.
#ucarp_downscript="/usr/local/sbin/ucarp-down" - statusun "aktivdən>passivə" keçidində işə
                                                düşəcək skript.

```

```

node2-üçün network və StartUP skript /etc/rc.conf

hostname="node2"
ifconfig_em0="inet 192.168.11.20 netmask 255.255.255.0"
ifconfig_em1="inet 172.16.1.20 netmask 255.255.255.0"
ifconfig_em1_alias0="vhid 100 pass freebsd advskew 10 alias 172.16.1.100/32"
ifconfig_em2="inet 10.1.1.20 netmask 255.255.255.0"
defaultrouter="192.168.11.1"
sshd_enable="YES"
dumpdev="NO"

# Firewall
#pf_enable="YES"                                     - Kernel kompilyasiya edildikdən sonra işləyəcək.
#pf_rules="/etc/pf.conf"
#pflog_enable="YES"
#pflog_logfile="/var/log/pflog"

# Disabled Services
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_rebuild_aliases="NO"
syslogd_enable="YES"
syslogd_program="/usr/sbin/syslogd"
syslogd_flags="-ss"
icmp_drop_redirects="YES"

# Services
#hastd_enable="YES"                                 - Kernel-ə 'options GEOM_GATE' opsiyasını
                                                       əlavə etməyi unutmayın.
#hastd_program="/sbin/hastd"                         - HAST-in daemon faylı.
#hastd_flags="-d -d -c /etc/hast.conf -P /var/run/hastd.pid" - HAST-in config və PID faylı.
                                                       - Portlardan '/usr/ports/net/ucarp'-i
                                                       yükledikdən sonra aktivləşdirin.
#ucarp_enable="YES"                                  - Cluster-in Virtual IP ünvani.
                                                       - İstifadə edilən şəbəkə kartı.

#ucarp_addr="10.1.1.100"
#ucarp_if="em2"

```

```

#ucarp_src="10.1.1.20"           - node-nun IP ünvanı.
#ucarp_pass="freebsd"            - node-lar arası identifikasiya şifresi.
#ucarp_vhid=42                  - Cluster-in unikal identifikatoru.
#ucarp_upscript="/usr/local/sbin/ucarp-up"
                                - Statusun "passivdən>aktivə" keçidində işə
                                düşəcək skript.
#ucarp_downscript="/usr/local/sbin/ucarp-down"
                                - Statusun "aktivdən>passivə" keçidində işə
                                düşəcək skript.
portsnap fetch extract update - Hər iki node-də portları yeniləyirik.

```

Hər iki node üçün Kernel-ə aşağıdakı modulları əlavə edib kompilyasiya edirik.

```

cd /sys/amd64/conf
cp GENERIC kernel

```

```

kernel faylinin sonuna aşağıdakı sətirləri əlavə edirik.
ident kernel                                - Kernel adını bizi uyğun olana dəyişirik.
# Carp Device
device carp

# Device's for PF
device      pf
device      pflog
device      pfsync

# Options For PF
# ALTQ (Alternate Queries)
options      ALTQ
options      ALTQ_CBQ
options      ALTQ_RED
options      ALTQ_RIO
options      ALTQ_HFSC
options      ALTQ_PRIQ
options      ALTQ_NOPCC

# HAST
options GEOM_GATE

```

**Qeyd:** Kernel-in kompilyasiyasında mütləq IPV6 (Yeni INET6)-ni saxlayın. Həst IPv6 olmadan işə düşməyəcək.

```
cd /usr/src                                - Kernel kompilyasiya ediləcək ünvanına  
                                            daxil oluruq.  
make buildkernel KERNCONF=kernel          - Kompilyasiya edirik.  
make installkernel KERNCONF=kernel         - Yükləyirik.
```

Kernel-in kompilyasiyası bitdikdən sonra hər iki node-də '/etc/rc.conf' faylında PF-ə aid olan sətirlərin qarşısından şəhri silib, lazımi faylı yaradırıq və içində aşağıdakı sətirləri əlavə edirik.

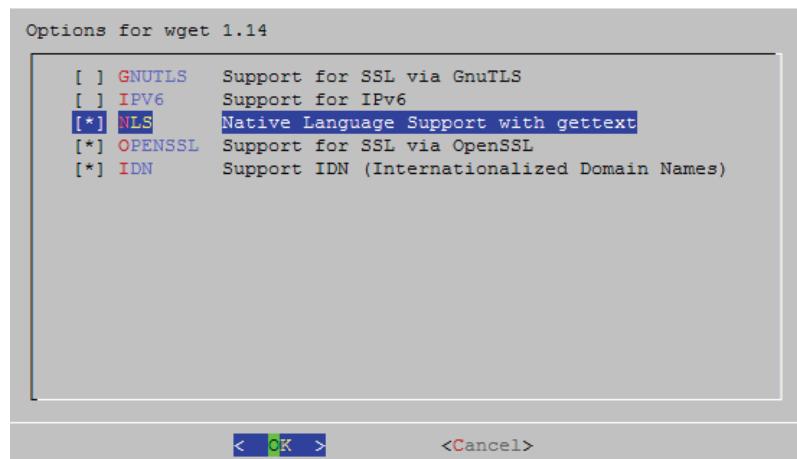
```
ee /etc/pf.conf                            - PF_in StartUP faylı.  
    pass in all  
    pass out all
```

```
reboot                                    - Serverimizi yenidənyüklənmə edək ki, kernel və  
                                            portlar yenilənsin.
```

Aşağıda göstərilən portların hamısı hər iki node-da yüklenir. (İsimizi rahatlaşdırıraq.)

```
cd /usr/ports/shells/bash  
make install clean
```

```
cd /usr/ports/ftp/wget  
make config
```

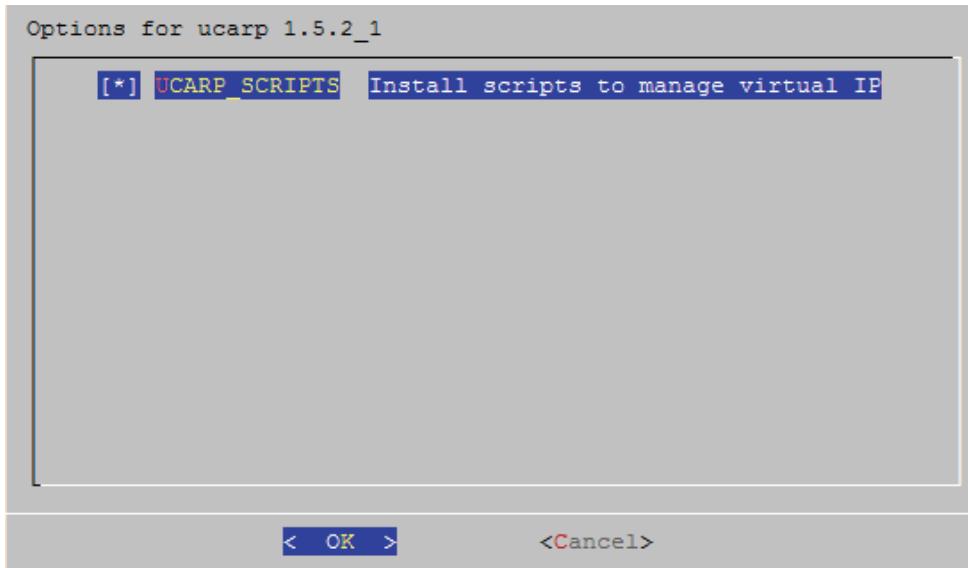


```
make install clean
```

```
cd `whereis nload | awk '{ print $2 }'`  
make install clean  
  
cd `whereis cmdwatch | awk '{ print $2 }'`  
make install clean
```

Hər iki node-də **ucarp**-ı yükleyirik ki, bize **VIP** və **mount** skriptlərindən istifadə etmək lazımlı olacaq. Yüklədikdən sonra hər iki node-un '**/etc/rc.conf**' faylında **ucarp** sətirlərinin qarşısından şəhri silməyi unutmayın.

```
cd /usr/ports/net/ucarp/  
make config
```



```
make install clean
```

Hər iki node-da ucarp skriptlərini lazımi ünvana nüsxələyirik və yerinə yetirilmə yetkisi veririk.

```
cp /usr/share/examples/hast/* /usr/local/sbin  
chmod +x /usr/local/sbin/*
```

Hər iki node-da sshd-də aşağıdakı dəyişiklikləri edirik ki, işimiz tezleşsin.

```
ee /etc/ssh/sshd_config  
PermitRootLogin yes  
UseDNS no
```

Hər iki node-da SSH token authentifikasiya edək ki, işimizi rahatlaşdırıraq.

**ssh-keygen -t rsa** - Enter sıxaraq Sertifikat generasiya edirik.

**cd /root/.ssh/** - Hər iki node-da bu ünvana daxil oluruq.

**root@node1:/root/.ssh # cp id\_rsa.pub node1.id\_rsa.pub**  
- node1-in açarının adını dəyişirik.

**scp node1.id\_rsa.pub root@172.16.1.20:/root/.ssh**  
- node1-in açarını node2-yə nüsxələyirik.

**cat node\* >> authorized\_keys** - Açıclarımızı siyahıya salaq ki, şifrəsiz daxil olaq.

**root@node2:/root/.ssh # cp id\_rsa.pub node2.id\_rsa.pub**  
- node2-nin açarının adını dəyişirik.

**scp node2.id\_rsa.pub root@172.16.1.10:/root/.ssh/**  
- node2-nin açarını node1-ə nüsxələyirik.

**cat node\* >> authorized\_keys** - Açıclarımızı siyahıya salaq ki, şifrəsiz daxil olaq.

Hər iki node-də '/etc/hosts' faylini quraşdırıraq.

**cat /etc/hosts**  
127.0.0.1 localhost  
10.1.1.10 node1  
10.1.1.20 node2

Hər iki node-da hast quraşdırmasını edək. Bu quraşdırma faylini yaratdıqdan sonra hər iki node-da '/etc/rc.conf'-da hast-a aid olan sətirlərin qarşısından şərhi silməyi unutmayın.

**ee /etc/hast.conf**

```
resource hast0 {  
    on node1 {  
        local /dev/dal  
        remote 10.1.1.20  
    }  
    on node2 {  
        local /dev/dal  
}
```

- Sinxronizasiya üçün yaradılacaq virtual disk resursu.  
- node1-in hosts faylında olan adı.  
- node1-in sinxronizasiya ediləcək fiziki disk.  
- node2-nin sinxronizasiya ediləcək IP ünvanı.  
  
- node2-nin hosts faylında olan adı.  
- node2-nin sinxronizasiya ediləcək fiziki disk.

```

    remote 10.1.1.10           - node1-in sinxorinzasiya ediləcək IP ünvanı.
}
}
}

```

### HAST data sinxornizasiyası metodları

**memsync** - Daxili diskə yazma bitdikdən və uzaq node-a datanın çatması haqqında məlumat göndərdikdən sonra hesabat verir, ancaq faktiki data yazılmazdan önce göndərir. Data uzaq node də cavab göndərildikdən sonra yazılıcaq. Bu rejim gecikmənin qarşısını almaq üçün istifadə edilir və etibarlıdır. Ancaq çatışmamazlığı odur ki, secondary server-ə məlumat ötürülməzdən önce report çatdırır, sonra şəbəkədə problem olur və kiçik dataların bəziləri data primary node-dən secondary-yə köçürürləndə itə bilər. Belə problemlər tək-tək ola bilər. Ancaq hal-hazırda memsync demək olar ki, istifadə edilmir.

**fullsync** - Daxili node və uzaq node-larda yazılmaya başa çatdıqdan sonra hesabat ötürür. Bu, çox etibarlı və gec işləyən replikasiya metodudur. Həmçinin susmaya görə olan sinxornizasiya metodudur.

**async** - Daxili diskə yazılmaya bitdikdən sonra hesabat verir. Bu, sürətli və təhlükəli replikasiya metodudur. Bu, o halda istifadə edilir ki, hər iki node arasında çoxlu gecikmələr olur. Bu rejim demək olar ki, artıq istifadə edilmir. **Hast**-in jurnalları '/var/log/messages' faylına yiğilir.

Lazımı skriptlərdə lazımı dəyişiklikləri edək. Aşağıdakı addımları hər iki node üçün edirik.

**ee /usr/local/sbin/ucarp.sh**

**addr="10.1.1.100"**

**pass="freebsd"**

- Faylda olan aşağıdakı sətirlərdə dəyişiklik edirik.

- VIP ünvanı istifadəçilər üçün.

- Node-lar UCARP-la öz aralarında bu şifrə ilə danışacaq.

- Ucarp qoşulması üçün node1-in IP ünvanı.

- Ucarp qoşulması üçün node1-in şəbəkə kartının adı.

- Ucarp qoşulması üçün node2-nin IP ünvanı.

- Ucarp qoşulması üçün node2-nin şəbəkə kartının adı.

**nodea\_srcip="10.1.1.10"**

**nodea\_ifnet="em2"**

**nodeb\_srcip="10.1.1.20"**

**nodeb\_ifnet="em2"**

**upscript="/usr/local/sbin/vip-up.sh"** - Faylda /usr/local/sbin/ucarp\_up.sh skriptini işə salması üçün lazımi redaktə edirik.

**downscript="/usr/local/sbin/vip-down.sh"**

- Faylda /usr/local/sbin/ucarp\_down.sh skriptini işə salması üçün lazımi redaktə edirik.

```

ee /usr/local/sbin/ucarp-up           - Faylında dəyişiklik edib,
    /sbin/ifconfig "$1" alias "$2" netmask 255.255.255.0
                                            - Sətrindən sonra aşağıdakı sətirləri əlavə edirik.
    set -m
    /usr/local/sbin/ucarp_up.sh &
    set +m

ee /usr/local/sbin/ucarp-down        - Faylinin sonuna aşağıdakı sətri əlavə edib yadda
                                            saxlayaraq çıxırıq.
    /usr/local/sbin/ucarp_down.sh &

ee /usr/local/sbin/ucarp_up.sh       - Faylda aşağıdakı dəyişənləri özümüzə uyğun olaraq
    resource="hast0"                 dəyişirik.
    fstype="UFS"                   - HAST-in virtual resursunun adı.
    mountpoint="/hast/hast0"        - İstifadə edəcəyimiz File sistemin tipi.
                                            - HAST-in virtual alətinin mount ediləcək qovluğunun tam
                                            ünvani.

ee /usr/local/sbin/ucarp_down.sh     - Eynilə bu faylda da aşağıdakı dəyişənləri özümüzə
    resource="hast0"                 uyğun olaraq dəyişirik.
    fstype="UFS"                   - HAST-in virtual resursunun adı.
    mountpoint="/hast/hast0"        - İstifadə edəcəyimiz File sistemin tipi.
                                            - HAST-in virtual alətinin mount ediləcək qovluğunun
                                            tam ünvani.

```

#### Sonra aşağıdakı əmrləri CLI-dən ardıcıl olaraq daxil edirik

```

mkdir -p /hast/hast0           - Hər iki node-da virtual alətin mount ediləcəyi qovluğu
                                yaradırıq.

hastctl create hast0          - Hər iki node-da hast0 adlı resurs yaradırıq.

hastd
root@node1:/root # hastctl status
hast0:
role: init                    - Hələ ki role naməlum statusdadır.
provname: hast0               - Virtual resursumuzun statusuna node1-də baxırıq.
localpath: /dev/dal           - Local Fiziki alət.

```

```
extentsize: 0 (0B)
keepdirty: 0
remoteaddr: 10.1.1.20          - Uzaq HAST istifadə edən IP ünvani.
replication: fullsync         - Sinxornizasiya metodu(susmaya görə bu olur).
dirty: 0 (0B)
statistics:
  reads: 0
  writes: 0
  deletes: 0
  flushes: 0
  activemap updates: 0
```

**hastctl role primary hast0** - node1-də **hast0** diskini **primary** edirik ki, sinxronizasiya buradan node2-yə getsin.

**root@node1:/usr/local/sbin # hastctl status** - Node1-də statusa baxırıq.

```
hast0:
  role: primary                  - Artıq Primary-dir.
  provname: hast0
  localpath: /dev/dal
  extentsize: 2097152 (2.0MB)
  keepdirty: 64
  remoteaddr: 10.1.1.20
  replication: fullsync
  status: degraded
  dirty: 0 (0B)
  statistics:
    reads: 23
    writes: 0
    deletes: 0
    flushes: 0
    activemap updates: 0
```

**root@node2:/ # hastctl role secondary hast0** - node2-nin virtual **hast0** diskini secondary edirik ki, qəbul edən olsun.

**root@node2:/ # hastctl status** - Statusuna baxırıq.

```
hast0:  
  role: secondary - Artıq secondary-dir.  
  provname: hast0  
  localpath: /dev/dal  
  extentsize: 2097152 (2.0MB)  
  keepdirty: 0  
  remoteaddr: 10.1.1.10  
  replication: fullsync  
  status: complete  
  dirty: 0 (0B)  
  statistics:  
    reads: 0  
    writes: 0  
    deletes: 0  
    flushes: 0  
    activemap updates: 0
```

```
root@node1:/ # newfs -U /dev/hast/hast0 - node1-də, yeni Primary olan serverimizdə  
hast0 virtual diskini UFS2 fayl sistemə  
format edirik.
```

```
root@node1:/ # mount /dev/hast/hast0 /hast/hast0/  
- node1-də format etdiyimiz virtual diskı hast üçün  
yaradığımız qovluğa mount edirik.
```

Hər iki node-da servislərimizin proseslərdə olduğunu yoxlayaq.

Öncə UCARP-ı proseslərdə yoxlayaq.

```
root@node1:/hast/hast0 # ps axf|grep ucarp - Node1-də Ucarp-ı yoxlayaq. Aşağıdakı sətərə  
uyğun olmalıdır.  
1210 ?? Ss 0:01.51 /usr/local/sbin/ucarp -i em2 -f daemon -B -p freebsd  
-z -u /usr/local/sbin/ucarp-up -d /usr/local/sbin/ucarp-down -s 10.1.1.10 -a  
10.1.1.100 -i em2 -v 42
```

```
root@node2:/ # ps axf|grep ucarp - Node2-də Ucarp-ı yoxlayaq, aşağıdakı sətərə  
uyğun olmalıdır.
```

```
1210 ?? Ss      0:00.74 /usr/local/sbin/ucarp -i em2 -f daemon -B -p freebsd  
-z -u /usr/local/sbin/ucarp-up -d /usr/local/sbin/ucarp-down -s 10.1.1.20 -a  
10.1.1.100 -i em2 -v 42
```

Sonra HAST-ı proseslərdə yoxlayaqq.

```
root@node1:/hast/hast0 # ps -ax | grep hast  
1190 ?? Ss      0:00.26 /sbin/hastd -d -d -c /etc/hast.conf -P /var/run/hastd.pid  
1528 ?? IJ      0:00.35 hastd: hast0 (primary) (hastd)
```

```
root@node2:/ # ps -ax | grep hast      - Node2-də HAST-ı yoxlayaqq, aşağıdakı sətirlərə  
1190 ?? Ss      0:00.26 /sbin/hastd -d -d -c /etc/hast.conf -P /var/run/hastd.pid  
1510 ?? IJ      0:00.21 hastd: hast0 (secondary) (hastd)
```

```
root@node1:/hast/hast0 # ifconfig em2|grep inet  
inet 10.1.1.10 netmask 0xffffffff broadcast 10.1.1.255  
inet 10.1.1.100 netmask 0xffffffff broadcast 10.1.1.100
```

Test edək.

```
mkdir /hast/hast0/papka/      - Cluster diskimizdə node1-də qovluq yaradırıq.  
touch /hast/hast0/papka/file{1,2,3} - Cluster diskimizdə papka adlı qovluqda 3 ədəd  
fayl yaradaqq.  
cat '/dev/zero' > /hast/hast0/papka/file1  
- Əmri daxil etdikdən 6 saniyə sonra 'Ctrl+C'  
əmri ilə break etməsəniz, disk tamam dolacaq  
(Biz informasiyanın nüsxələnməsini test edirik).
```

```
root@node1:/ # pkill -USR2 -f 'ucarp'
- node1-də ucarp-a "USR2" siqnalı ötürürük ki,
Cluster digər node-a keçid etsin.
```

```
root@node2:/ # df -h
Filesystem      Size  Used  Avail Capacity Mounted on
/dev/da0p2      18G   4.5G   12G   27%    /
devfs          1.0k   1.0k   0B   100%   /dev
/dev/hast/hast0 19G   438M   17G   2%    /hast/hast0
- Həqiqətən də, 438MB-lıq informasiyamız sinxron olub.
```

```
root@node2:/ # ifconfig em2 | grep inet
- node2-də UCARP-in yaratdığı VIP-ə baxırıq.
(10.1.1.100)
inet 10.1.1.20 netmask 0xffffffff broadcast 10.1.1.255
inet 10.1.1.100 netmask 0xffffffff broadcast 10.1.1.255
```

```
root@node1:/ # tail -f /var/log/messages
- node1-də jurnallara baxırıq.
Mar 12 16:49:10 node1 ucarp[1210]: [WARNING] Switching to state: BACKUP
Mar 12 16:49:10 node1 ucarp[1210]: [WARNING] Spawning
[/usr/local/sbin/ucarp-down em2 10.1.1.100]
```

**Split-Brain** olan halda. Yəni, şəbəkədə problem olsa, hər iki **node**-un statusu **Primary** olacaq və bu haqda '**/var/log/messages**' faylında oxşar jurnal çap ediləcək. Bu situasiyanın aradan qaldırılmasının yeganə yolu passiv nəzərdə tutulacaq node-da virtual hast diskini yenidən yaratmaq lazımdır. Aşağıdakı ardıcılıqlıda həmin secondary sayılacaq node-da əmrləri daxil edirik.

<b>hastctl role init hast0</b>	- <b>hast0</b> resursumuzu neytral elan edirik, yəni nə aktiv, nə də passivdir.
<b>hastctl create hast0</b>	- <b>hast0</b> adlı resursu yenidən yaradırıq.
<b>hastctl role secondary hast0</b>	- passiv nəzərdə tutduğumuz node-dakı <b>hast0</b> resursunu secondary elan edirik.
<b>hastctl status</b>	- Resursumuzun statusuna baxırıq.

Uğurlu netice aşagıdakı kimi olmalıdır:

```
root@node1:~ # hastctl status
Name      Status   Role          Components
hast0    complete primary        /dev/dal      10.1.1.20

root@node2:~ # hastctl status
Name      Status   Role          Components
hast0    complete secondary       /dev/dal      10.1.1.10
```

# BÖLÜM 14

## CDP, .1Q, Dinamik Routing, Bridge, LACP, LAG,CARP, DHCP Relay

- / UNİX CDP, FreeBSD .1Q və Dinamik Routing
- / Bridging (Körpülənmə)
- / Link Aggregation və Failover
- / Common Address Redundancy Protocol(CARP)
- / FreeBSD DHCP Relay quraşdırılması

Şirkətin daxilində tələb yarana bilər ki, Cisco avadanlıqları öz portlarına qoşulmuş olan avadanlıqlar haqqında ilkin məlumat almmalıdır. Bu halda sizin köməyiniziə CDP çatacaq. Başlığımız FreeBSD üzərində CDP-nin qurulmasını açıqlayacaq. Əgər FreeBSD serverin fiziki portunda çatışmamazlıq yaranarsa, siz portun birini trunk rejimində istifadə edə bilərsiniz. Ya da tələb yarana bilər ki, sistemimizin üzərində dinamik şəbəkə yayılmışlığı işini görək. Bu başlığımızda FreeBSD üzərində VLAN yaradılması və dinamik routing olan OSPF-in qurulub test edilməsi açıqlanacaq. Bir neçə şəbəkə kartının körpü edilməsi və digər mükəmməl bridge bacarıqları bu başlıqda göstəriləcək (həmçinin Spanning Tree Protokol). Wireless ilə Ethernet şəbəkə kartlarının körpülənməsi barədə də bu başlıqda danışılır. Əməliyyat sistemimiz bir neçə eyni nüsxə ilə fərqli maşınlarda klaster rejimində işləyərsə, resursun istifadəçilər üçün virtual bir IP üzərindən yayılmışına tələb yaranır. Bu başlıqda həmin tələbin qarşılanması izah edilir. Əgər sizin şəbəkədə istifadəçilər çoxlu fərqli şəbəkələrdə yerləşərsə və həmin şəbəkələr üçün bir ədəd DHCP server fərqli skopları yayımılayarsa, size DHCP relay server tələb ediləcək. Bu başlıqda DHCP relay-in qurulması və sınaqdan keçirilməsi açıqlanır.

# UNIX CDP, FreeBSD .1Q və Dinamik Routing

Başlığımızda FreeBSD serverin 2-ci səviyyədə Switch və 3-cü səviyyədə Router kimi istifadə etməyin metodlarını açıqlayacaq. Serverimizin şəbəkə kartını Trunk rejimdə və özünü router rejimdə istifadə edəcəyik. Router rejimi həm statik, həm də dinamik istifadə olunacaq. Dinamik routing protokolu isə OSPF istifadə ediləcək.

## Unix CDP (Cisco Discovery Protocol)

Unix OS üzərində Cisco-nun Cisco Discovery Protocol-nu istifadə etmək üçün CDPD adlı daemon mövcuddur, hansı ki, biz FreeBSD üzərində hal-hazırda yükləyib yoxlayacaqıq.

```
cd /usr/ports/net-mgmt/cdpd           - Bu, daemon özürdür, hansı ki, CDP-ni işə salır.  
make install                          - Yükləyirik.
```

```
echo 'cdpd_enable="YES"' >> /etc/rc.conf - CDP daemon-u StartUP-a əlavə edirik.  
/usr/local/etc/rc.d/cdpd start        - CDPD-ni işə salırıq.
```

Ancaq biz öz informasiyamızı CDP ilə digər avadanlıqlara ötürdüük. Əgər biz də cdp ilə digər avadanlıq və portları görmək istəsək, onda aşağıdakı portu yükləməliyik.

```
cd /usr/ports/net-mgmt/cdpr  
make install
```

- Port ünvanına daxil oluruz.  
- Yükləyirik.

CDPR-i (CDP Reporter) işə salaq. Hal-hazırda **em0** ələtini seçəcəyik.

```
root@cacti:/usr/ports/net-mgmt/cdpr # cdpr  
cdpr - Cisco Discovery Protocol Reporter  
Version 2.2.1  
Copyright (c) 2002-2006 - MonkeyMental.com
```

```
1. em0 (No description available)  
2. lo0 (No description available)  
Enter the interface number (1-2):1  
Using Device: em0  
Waiting for CDP advertisement:  
(default config is to transmit CDP packets every 60 seconds)  
Device ID  
    value: cacti.bvim.lan  
Addresses  
    value: 10.40.7.115  
Port ID  
    value: em0
```

## FreeBSD Vlan .1q

Əgər serverin fiziki şəbəkə kartı VLAN-nı dəstəkləyirsə, FreeBSD əməliyyat sistemi də susmaya görə VLAN dəstəkləyir. VLAN – bir fiziki şəbəkə üzərindən saysız virtual şəbəkələrin daşınması üçün istifadə edilir. Əgər bizim serverin iki şəbəkə kartı olarsa və bütün daxili istifadəçilərimizi ayrı-ayrı şəbəkələrə bölmək istəyiriksə, problem yaranacaq. Çünkü bir şəbəkə kartımız dünyaya çıxış üçün, digəri isə core switch tərəfə baxacaq. Əgər FreeBSD maşınımızı Router etsək, switch tərəfə baxan şəbəkə kartını trunk(çoxlu vlanlar dəstəkləyən) port etmək lazımdır ki, hər şəbəkə üçün ayrıca SVI(switch vlan interface) yarada bilək. Bu SVI-lar ayrı-ayrı şəbəkələrdə olan istifadəçilər üçün susmaya görə olan yol olacaq.

VLAN-ın sistem yenidənyüklənməsindən sonra avtomatik işə düşməsi üçün StartUP faylı "**/etc/rc.conf**"-a əlavə etmək lazımdır. **Vlan 779 və 597** yaradılır, bce0 şəbəkə kartından keçəcəkləri təyin edilir. (**MTU - 1** paketdə ötürülə biləcək baytlarla olan maksimal həcmidir. Aşağıdakı əmrle **MTU 1518 bayt** təyin edilir):

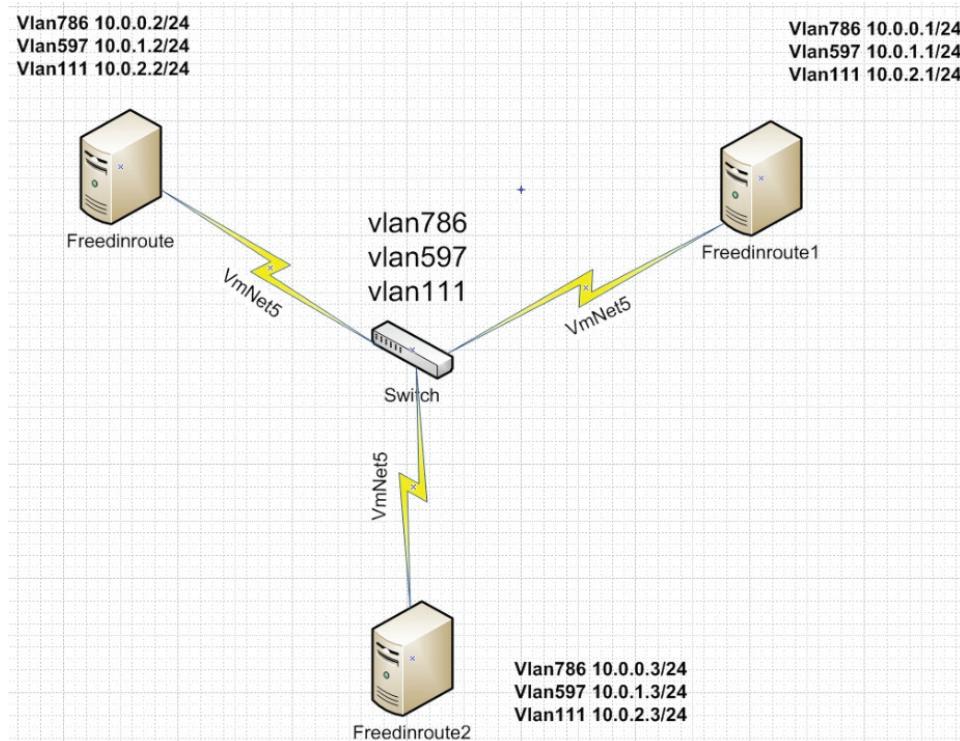
```
ifconfig_bge0="up mtu 1518"  
cloned_interfaces="vlan779 vlan597"
```

```
ifconfig_vlan779="inet x.x.x.x netmask 255.255.255.0 vlan 779 vlandev bge0"
ifconfig_vlan597="inet x.x.x.x netmask 255.255.255.0 vlan 597 vlandev bge0"
```

Eyni işi CLI'dan görmek için ise, aşağıdaki emrlərdən istifadə etmək lazımdır. Vlan4 yaradılır, IP təyin edilir və MTU həcmi verilir.

```
ifconfig_vlan4 create
ifconfig_vlan4 inet 192.168.1.162 netmask 255.255.255.252 wlan 4 vlandev em1
ifconfig_em0 mtu 1518
```

Laboratoriymız üçün **3** ədəd serverimiz var. Hər serverin **1** ədəd şəbəkə kartı var. Məqsədimiz hər serverin üzərindən **3** ədəd VLAN keçirmək və bu VLAN alətləri qonşu IP ünvanlarla danışdırmaqdır. Topologiyamız aşağıdakı kimi olacaq:



**hosta.example.com** maşınınında StartUP skripti **/etc/rc.conf**-da olan quraşdılmalarımız aşağıdakı kimi olacaq:

```
ifconfig_em0="up mtu 1518"
cloned_interfaces="vlan786 wlan597 wlan111"
```

```
ifconfig_vlan786="inet 10.0.0.1 netmask 255.255.255.0 wlan 786 wlan dev em0"
ifconfig_vlan597="inet 10.0.1.1 netmask 255.255.255.0 wlan 597 wlan dev em0"
ifconfig_vlan111="inet 10.0.2.1 netmask 255.255.255.0 wlan 111 wlan dev em0"
```

**hostb.example.com** maşınınında StartUP skripti **/etc/rc.conf**-da olan quraşdırımlarımız aşağıdakı kimi olacaq:

```
ifconfig_em0="up mtu 1518"
cloned_interfaces="vlan786 wlan597 wlan111"
ifconfig_vlan786="inet 10.0.0.2 netmask 255.255.255.0 wlan 786 wlan dev em0"
ifconfig_vlan597="inet 10.0.1.2 netmask 255.255.255.0 wlan 597 wlan dev em0"
ifconfig_vlan111="inet 10.0.2.2 netmask 255.255.255.0 wlan 111 wlan dev em0"
```

**hostc.example.com** maşınınında StartUP skripti **/etc/rc.conf**-da olan quraşdırımlarımız aşağıdakı kimi olacaq:

```
ifconfig_em0="up mtu 1518"
cloned_interfaces="vlan786 wlan597 wlan111"
ifconfig_vlan786="inet 10.0.0.3 netmask 255.255.255.0 wlan 786 wlan dev em0"
ifconfig_vlan597="inet 10.0.1.3 netmask 255.255.255.0 wlan 597 wlan dev em0"
ifconfig_vlan111="inet 10.0.2.3 netmask 255.255.255.0 wlan 111 wlan dev em0"
```

Sonda hər üç maşına yenidənyüklənmə edirik. Hər 3 maşın hər birinin IP ünvanlarını ping vasitəsilə görəcək.

## FreeBSD 10.1 Dynamic Routing

FreeBSD serveri Router rejimində çalışdırmaq imkanı mövcuddur. Router etmək üçün iki seçim var.

İlk üsul

```
sysctl -a | grep net.inet.ip.forwarding - Susmaya görə routing rejim 0 (false)-dir.
sysctl net.inet.ip.forwarding=1 - CLI vasitəsilə true edirik. Yəni Route rejimini işə salırıq.
```

**/etc/syslog.conf** kernel StartUP faylına əlavə edirik ki, yenidənyüklənmədən sonra avtomatik işə düşsün:

```
net.inet.ip.forwarding=1 - Router rejimini işə salırıq.
```

İkinci üsul

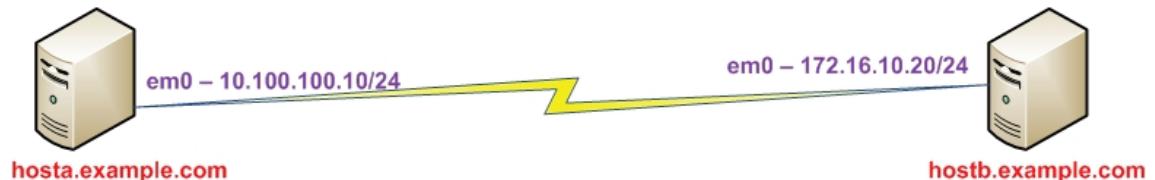
**/etc/rc.conf** StartUP faylına əlavə edirik ki, sistem yenidənyükləşməsindən sonra avtomatik işə

düşsün:

**gateway\_enable="YES"**

- Router rejimini işe salırıq.

Başqa çox maraqlı bir misal çəkə bilərik. Deyək ki, iki ədəd qonşu olan FreeBSD maşınınız var və bir-birlərinə birbaşa kabellə qoşuludurlar. Ancaq hər bir server tamam fərqli şəbəkələrdə yerləşir. Hər iki maşın FreeBSD10.1 x64-dür. Məqsədimiz bu şəbəkələri bir-birlərinə tanıtmadır. Aşağıdakı şəkildə situasiya göstərilir:



Öncə **hosta.example.com** maşından **hostb.example.com** maşına doğru statik route yazırıq ki, **172.16.10.20** IP ünvanını görə bilək:

```
root@hosta:~ # route add -host 172.16.10.20/32 10.100.100.10
```

Sintaksisi açıqlayaq:

**172.16.10.20**

- Görünməsi lazımlı olan IP ünvan.

**10.100.100.10**

- Öz serverimzdə hansı IP ünvandan keçəcək.

StartUP üçün işə **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə edirik:

**static\_routes="hostb"**

**route\_hostb="-host 172.16.10.20/32 10.100.100.10"**

Sonra **hostb.example.com** maşından **hosta.example.com** maşına doğru statik route yazırıq ki, **10.100.100.10** IP ünvanı görə bilək:

```
root@hostb:~ # route add -host 10.100.100.10/32 172.16.10.20
```

Sintaksis açıqlayaq:

**10.100.100.10**

- Görünməsi lazımlı olan IP ünvan.

**172.16.10.20**

- Öz serverimzdə hansı IP ünvandan keçəcək.

StartUP üçün işə **/etc/rc.conf** faylına aşağıdakı sətirləri əlavə edirik:

**static\_routes="hosta"**

**route\_hosta="-host 10.100.100.10/32 172.16.10.20"**

Həmçinin biz routing-i direct elan edərək yaza bilərik. Bunun üçün **-iface** (mənsəbə birbaşa çatmaq olur) adında opsiya mövcuddur. Misalda açıqlayacaq. İki ədəd serverimiz var:

**hostc.example.com - em0 - 192.168.2.1/24**

**hostb.example.com - em1 - 192.168.100.10/24**

**Hostc** maşının şəbəkəsini görmək üçün **hostb** maşından route əlavə edirik:

```
root@hostb:~ # route add -net 192.168.2.0/24 -iface 192.168.100.10
```

Əlavə etdiyimiz route haqqında məlumatı çap edirik:

```
root@hostb:~ # route get 192.168.2.0
```

```
route to: 192.168.2.0
destination: 192.168.2.0
      mask: 255.255.255.0
    gateway: 192.168.100.10
        fib: 0
   interface: em1
      flags: <UP,GATEWAY,DONE,STATIC>
recvpipe    sendpipe    ssthresh    rtt,msec      mtu      weight      expire
      0          0            0              0       1500           1             0
```

**Hostb** maşının şəbəkəsini görmək üçün hostc maşından route əlavə edirik:

```
root@hostc:~ # route add -net 192.168.100.0/24 -iface 192.168.2.1
```

Əlavə etdiyimiz route haqqında məlumatı çap edirik:

```
root@hostc:~ # route get 192.168.100.0
```

```
route to: 192.168.100.0
destination: 192.168.100.0
      mask: 255.255.255.0
    gateway: 192.168.2.1
        fib: 0
   interface: em0
      flags: <UP,GATEWAY,DONE,STATIC>
recvpipe    sendpipe    ssthresh    rtt,msec      mtu      weight      expire
      0          0            0              0       1500           1             0
```

Ancaq hostc.example.com maşını üçün **/etc/rc.conf** StartUP faylinin lazımı sintaksisleri aşağıdakı kimi olacaq:

```
static_routes="hostc"  
route_hostc="-net 192.168.100.0/24 -iface 192.168.2.1"
```

#### OSPF (Open Shortest Path First) Routing

Üç ədəd serverimiz var. Hər bir serverimizin iki ədəd şəbəkə kartı mövcuddur. Hər şəbəkə kartı öz qonşu serveri ilə eyni VMnet üzərində işləyir. Testlərimiz VmWare üzərində aparılmışdır. Virtual şəbəkələrimizi hər bir server üçün aşağıda sadalayıraq.

Vmnet5 virtual avadanlığında:

```
hosta.example.com - em0 - 192.168.0.2/24  
hostb.example.com - em0 - 192.168.0.1/24
```

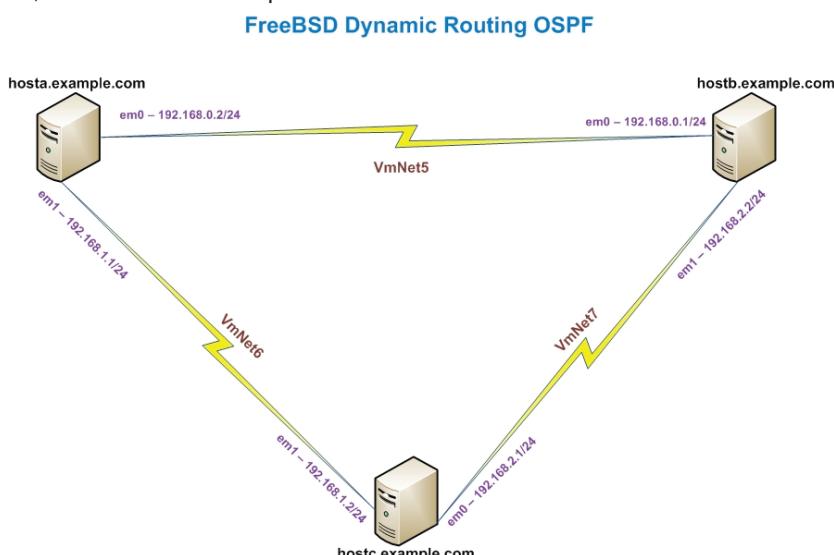
Vmnet6 virtual avadanlığında:

```
hosta.example.com - em1 - 192.168.1.1/24  
hostc.example.com - em1 - 192.168.1.2/24
```

Vmnet7 virtual avadanlığında:

```
hostc.example.com - em0 - 192.168.2.1/24  
hostb.example.com - em1 - 192.168.2.2/24
```

Topologiyamız şəkildəki kimi olacaq:



Hər üç serverdə "**portsnap fetch extract update**" etmək lazımdır.

Hər üç serverdə **quagga** paketini yüklemək lazımdır.

Hər üç server routing rejimdə çalışmalıdır.

Hər üç serverdə quagga-nı yükləyirik.

- Ünvang daxil olurug.

- Aşağıdakı modulları seçirik.

make install clean

- Yükleyirik.

Aşağıdakı qurasdırmaqları hər üç serverdə edirik:

```
cd /usr/local/share/examples/quagga
```

- Ünvana daxil oluruq.

- Bütün faylları quaggalı

```
cd /usr/local/etc/quagga
```

- Quaasq Buubay ün qaxiq.

Bütün nüsxə faylların adını dəvəsirik (evniliə hər üç serverda).

my\_bond.conf.sample bond.conf

my ospf6d.conf.sample ospf6d.conf

## my\_ospfd.conf.sample ospfd.conf

my find.conf.sample find.conf

my ripngd.conf.sample ripngd.conf

```
mv vtysh.conf.sample vtysh.conf  
mv zebra.conf.sample zebra.conf  
mv babeld.conf.sample babeld.conf  
mv isisd.conf.sample isisd.conf
```

Birinci **hostb.example.com** serveri quraşdırıq. StarUP faylımız **/etc/rc.conf**-a aşağıdaki sətirləri əlavə edirik:

```
ifconfig_em0="inet 192.168.0.1 netmask 255.255.255.0"  
ifconfig_em1="inet 192.168.2.2 netmask 255.255.255.0"  
defaultrouter="NO"  
hostname="hostb.example.com"  
sshd_enable="YES"  
quagga_enable="YES"  
quagga_flags="-d"  
quagga_daemons="zebra ripd ripngd ospfd ospf6d bgpd"  
quagga_vysh_boot="YES"  
gateway_enable="YES"
```

Serverimizin IP ünvanlarına ping getməsini yoxlayırıq. Ardınca isə **Quagga**-nı işə salırıq.

```
/usr/local/etc/rc.d/quagga start  
telnet 127.0.0.1 ospfd
```

- Quagga işə salınır.  
- Daxil olurraq Quagga-ya.

Susmaya görə Quagga-nın **tty** şifrəsi "**zebra**" və **enable** şifrəsi "**zebra**"-dir.  
Daxil olduqdan sonra Quagga-nı **OSPF** dinamik routing etməsi üçün quraşdırırıq.

```
configure terminal  
router ospf  
ospf router-id 192.168.0.1  
  
network 192.168.0.0/24 area 0  
network 192.168.2.0/24 area 1  
neighbor 192.168.0.2  
  
neighbor 192.168.2.1
```

- Qlobal quraşdırma rejiminə daxil olurraq.  
- Routing üçün **OSPF** protokol seçirik.  
- Router-id təyin edirik (istənilən şəbəkə kartımızın birini təyin etmək kifayət edir).  
- **192.168.0** şəbəkəsi üçün area sıfır seçirik.  
- **192.168.2** şəbəkəsi üçün area bir seçirik.  
- Şəbəkə kartlarımıza qonşu olan IP ünvanlarını yazırıq.  
- Şəbəkə kartlarımıza qonşu olan IP ünvanlarını yazırıq.

```
default-information originate
```

- Susmaya görə bu router-dən marşrut olunur (internet server olan).

```
Ctrl+z
```

- İlk quraşdırma səviyyəsinə qayıdırıq.

```
wr
```

- Yadda saxlayırıq.

**hosta.example.com** maşınınızın StartUP **/etc/rc.conf** faylini aşağıdakı kimi quraşdırırıq:

```
hostname="hosta.example.com"
```

```
ifconfig_em0="inet 192.168.0.2 netmask 255.255.255.0"
```

```
ifconfig_em1="inet 192.168.1.1 netmask 255.255.255.0"
```

```
defaultrouter="NO"
```

```
quagga_enable="YES"
```

```
quagga_flags="-d"
```

```
quagga_daemons="zebra ripd ripngd ospfd ospf6d bgpd"
```

```
quagga_vysh_boot="YES"
```

```
gateway_enable="YES"
```

```
sshd_enable="YES"
```

Serverimizin IP ünvanlarına ping getməsini yoxlayırıq. Ardınca isə Quagga-nı işə salırıq.

```
/usr/local/etc/rc.d/quagga start
```

- Quagga işə salınır.

```
telnet 127.0.0.1 ospfd
```

- Daxil oluruq Quagga-ya.

Şifrə və enable şifrəni daxil edirik."zebra"

```
configure terminal
```

- Qlobal quraşdırma rejiminə keçirik.

```
router ospf
```

- Routing üçün OSPF protokol seçirik.

```
ospf router-id 192.168.0.2
```

- Router-id təyin edirik (istənilən şəbəkə kartımızın birini təyin etmək kifayət edir).

```
network 192.168.0.0/24 area 0
```

- 192.168.0 şəbəkəsi üçün area sıfır seçirik

```
network 192.168.1.0/24 area 2
```

- 192.168.1 şəbəkəsi üçün area iki seçirik.

```
neighbor 192.168.0.1
```

- Şəbəkə kartlarımıza qonşu olan İP ünvanları yazırıq.

```
neighbor 192.168.1.2
```

- Şəbəkə kartlarımıza qonşu olan İP ünvanları yazırıq.

```
redistribute connected
```

- Bu, o demekdir ki, mənim interfeyslərimə qoşulan portnyorlarımıza OSPF ilə anons etmək lazımdır.

```
Ctrl+z
```

- İlk quraşdırma səviyyəsinə qayıdırıq (Ya da **do write**).

**wr**

- Yadda saxlayırıq.

```
hostc.example.com serverimizin StartUP /etc/rc.conf faylini quraşdırırıq:  
ifconfig_em0="inet 192.168.2.1 netmask 255.255.255.0"  
ifconfig_em1="inet 192.168.1.2 netmask 255.255.255.0"  
hostname="hostc.example.com"  
defaultrouter="NO"  
sshd_enable="YES"  
quagga_enable="YES"  
quagga_flags="-d"  
quagga_daemons="zebra ripd ripngd ospfd ospf6d bgpd"  
quagga_vysh_boot="YES"  
gateway_enable="YES"
```

Serverimizin IP ünvanlarına ping getməsini yoxlayırıq. Ardınca isə Quagga-nı işə salırıq.

```
/usr/local/etc/rc.d/quagga start  
telnet 127.0.0.1 ospfd
```

- Quagga işə salınır.

- Daxil olurraq Quagga-ya.

Şifrə və enable şifrəni daxil edirik."zebra"

```
configure terminal  
router ospf  
ospf router-id 192.168.2.1
```

- Qlobal rejim.

- Routing üçün OSPF protokol seçirik.

- Router-id təyin edirik (istənilən şəbəkə kartımızın birini təyin etmək kifayət edir).

```
network 192.168.1.0/24 area 2  
network 192.168.2.0/24 area 1  
neighbor 192.168.1.1
```

- 192.168.1 şəbəkəsi üçün **area** iki seçirik.

- 192.168.2 şəbəkəsi üçün **area** bir seçirik.

- Şəbəkə kartlarımıza qonşu olan IP ünvanlarını yazırıq.

- Şəbəkə kartlarımıza qonşu olan IP ünvanlarını yazırıq.

- Bu, o deməkdir ki, mənim interfeyslərimə qoşulan partnyorlarımı **OSPF** ilə anons etmək lazımdır.

- İlk quraşdırma səviyyəsinə qayıdırıq

(ya da do write).

- Yadda saxlayırıq.

**wr**

Sonda ping vasitəsilə bütün ünvanları hər 3 serverdə yoxlayırıq.

# Bridging (Körpüləmə)

**Bridge** bəzi hallarda şəbəkəni bölmək üçün istifadə olunur. Misal üçün, Ethernet seqmentində, şəbəkə seqmentində IP şəbəkələrin yaradılması işini görmədən və seqmentlərin birləşdirilməsi üçün router-dən istifadə etmədən bu işi görmək mümkündür. İki şəbəkəni bir-biri ilə birləşdirəcək avadanlıq "bridge" adlanır.

Bridge hər şəbəkə kartında olan MAC ünvanlarının öyrənməsi ilə işləyir. O, şəbəkələr arasında olan trafiki yalnız mənbə və mənsəb MAC ünvanları fərqli şəbəkələrdə olarsa, yönləndirir. Əksər yerlərdə bridge elə switch kimi olur və çox az portlu olur. Çoxlu şəbəkə kartı olan FreeBSD maşını bridge kimi işləyə bilər.

Bridge aşağıdakı hallarda istifadə edilə bilər:

Şəbəkələrin birləşdirilməsi.

Bridgin əsas əməliyyatı iki və ya daha çox şəbəkə seqmentlərinin birləşdirilməsidir. Şəbəkə avadanlığı əvəzində host bazalı bridge-in istifadə edilməsi üçün çoxlu səbəblər var. Misal üçün, kabelin məhdudlaşdırılması və ya firewall. Siz həmçinin Wireless şəbəkə kartınız ilə Ethernet arasında bridge edib, server HostAP(Access Point) kimi işlədə bilərsiniz.

Filtrləmə/Firewall-la şəbəkənin boğulması

Yönləndirmə və ya NAT olmadan firewall funksionallığına ehtiyac olarsa, bridge istifadə edilə bilər. Misal üçün, 9-cu başlığımızda olan "**PF Bridge Firewall**"

#### Network Tap

Bridge iki şəbəkə seqmentinə üzv olaraq səliqə ilə bütün Ethernet çərçivələrini **bpf(4)** və **tcpdump(1)** vasitəsilə öz bridge şəbəkə kartında yoxlaya, ya da bütün şəbəkə çərçivələrini əlavə şəbəkə kartına nüsxələyə bilir (Cisco SPAN port məntiqi).

#### Layer 2 VPN

İki Ethernet şəbəkəsinin IP link üzərindən birləşdirilməsi üçün şəbəkələrin **EtherIP** tunelinə, ya da **tap** (open VPN tərəfindən istifadə edilir) alətinə bridge edilməsi kifayətdir.

#### Layer 2 dayanıqlığı

Şəbəkə çoxlu link istifadə edərək birlikdə qoşula və təkrarlanan yolların bağlanması üçün Spanning Tree Protocol (STP) istifadə edə bilər.

Bu başlıq FreeBSD əməliyyat sisteminin **if\_bridge(4)** istifadə edərək bridge kimi quraşdırılmasını açıqlayacaq. Netgraph bridge-ləmə driver-i də istifadə edilə bilər və bu, **if\_bridge(4)**-də açıqlanır.

**Qeyd:** Paket filtrasiyası üçün istənilən firewall paketi istifadə edilə bilər(hamısı **pfil(9)** program mühitində işləyirlər). Bridge, həmçinin şəbəkəni boğmaq üçün **altq(4)** və **dummynet(4)** vasitəsilə istifadə edilə bilər.

### Bridge işə salınması

FreeBSD əməliyyat sistemində **if\_bridge(4)** kernel-in moduludur və avtomatik olaraq **ifconfig(8)** tərəfindən bridge şəbəkə kartı yarandıqda yüklenir. Həmçinin mümkündür ki, bridge avadanlığını yeni kernel faylında **device if\_bridge** sətri ilə kompilyasiya edəsiniz.

Bridge şəbəkə kartının klonlaşdırılması ilə yaradılır. Aşağıdakı kimi:

```
root@Bridgel:~ # ifconfig bridge create
```

**bridge0**

```
root@Bridgel:~ # ifconfig bridge0
```

```
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
```

```
ether 02:63:bd:f0:53:00
```

```
id 00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
```

```
maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
root id 00:00:00:00:00 priority 0 ifcost 0 port 0
```

Bridge şəbəkə kartı yaradıldıqda, avtomatik olaraq təsadüfi Ethernet ünvanı generasiya edir. **maxaddr** və **timeout** parametrləri **forwarding** cədvelində bridge-in saxlayacağı MAC ünvanlarının sayını və hər bir yazışdan önce son dəfə görünməsindən neçə saniyə sonra silinməsini idarə edir. Digər parametrlər isə **STP (Spanning Tree Protocol)**-nin necə işləyəcəyini təyin edir.

Sonra isə təyin edin ki, hansı şəbəkə kartları bridge şəbəkə kartının üzvü olacaq. Bridge-in paketləri yönləndirə bilməsi üçün bütün şəbəkə kartları və bridge qalxmış(isə salınmış) vəziyyətdə olmalıdır:

```
root@Bridgel:~ # ifconfig bridge0 addm bce0 addm bcel up
root@Bridgel:~ # ifconfig bce0 up
root@Bridgel:~ # ifconfig bcel up
```

**Qeyd:** Əgər ağıllı switch üzərindən test edirsizsə, STP işləyəcək. Problemin qarşısını öncədən almaq üçün serverdən Bridge kimi gələn şəbəkə kartlarına aid olan portları tam ayrılmış VLAN-a keçirin və sonra işinizi başlayın.

Bridge artıq **bce0** və **bcel** şəbəkə kartları arasında olan Ethernet (**frame**) çərvəni yönləndirə bilər. Həmçinin **/etc/rc.conf** StartUP faylinə aşağıdakı sətirləri əlavə edin ki, bridge şəbəkə kartınız sistemin yenidənyüklənməsində avtomatik işə düşsün:

```
cloned_interfaces="bridge0"
ifconfig_bridge0="addm bce0 addm bcel up"
ifconfig_bce0="up"
ifconfig_bcel="up"
```

Əgər bridge olan serverin IP ünvana ehtiyacı varsa, IP-ni bridge şəbəkə kartında təyin edin, ona aid olan üzv şəbəkə kartlarında deyil. Ünvan statik, ya da DHCP ilə ola bilər. Aşağıdakı misal statik IP ünvanı təyin edəcək:

```
root@Bridgel:~ # ifconfig bridge0 10.100.100.1/24
```

Həmçinin bridge şəbəkə kartına IPv6 ünvanı da təyin edə bilərsiniz. Dəyişikliyi həmişəlik etmək üçün ünvanı **/etc/rc.conf** faylinə əlavə edin.

**Qeyd:** Paket filtrasiyası aktiv olduqda, bridge edilmiş paketlər giriş filtrində önce fiziki şəbəkə kartları, ardınca isə bridge şəbəkə kartı üzərindən keçir və çıxışda da önce bridge şəbəkə

kartına düşüb, ardından fiziki şəbəkə kartlarından çıxır. Hər hansı bir addım dayandırıla bilər. Paket axınının yönlendirilməsi vacib olduqda, ən yaxşı üsul bridge şəbəkə kartının filtrasiyası yox, fiziki şəbəkə kartlarının filtrasiyasıdır. Bridge-in İP olmayan, İP paketlər olan və **Layer2 firewall** (**İPFW(8)**) işinin görülməsi üçün çoxlu quraşdırma imkanları var. Əlavə məlumat üçün **if\_bridge(4)**-ə baxın.

## Spanning Tree

Ethernet şəbəkəsində düzgün işləməsi üçün iki avadanlıq arasında yalnız bir aktiv yol mövcud ola bilər. STP protokolu loop-u təyin edir və təkrarlanan linkləri **blocked** statusuna yerləşdirir. Əgər aktiv linklərdən birində səhv olarsa, STP özünün digər ağacını (tree) hesablayır və bağlı olan yollardan birini işə salır ki, şəbəkənin bütün nöqtələrinə olan qoşulmayı bərpa etsin.

**Rapid Spanning Tree Protocol (RSTP)** - STP ilə varis olmuş əks-uyğunluğu təmin edir. RSTP qonşu switch-lərlə daha tez uyğunlaşır və informasiya mübadiləsi edir ki, **loop** yaratmadan daha tez **forward** (yönləndirmə) rejiminə keçid etsin. FreeBSD aparıcı rejimlərdə STP və RSTP dəstəkləyir, ancaq susmaya görə olan rejimdə RSTP istifadə edir.

STP-ni üzv olan şəbəkə kartlarında **ifconfig(8)** istifadə edərək işə sala bilərsiniz. Hazırkı **bridge** edilmiş fiziki **bce0** və **bce1** şəbəkə kartlarında STP-ni aşağıdakı əmrlə işə sala bilərsiniz:

```
root@Bridge1:~ # ifconfig bridge0 stp bce0 stp bce1
```

Nəticə aşağıdakı kimi olacaq:

```
root@Bridge1:~ # ifconfig bridge0
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
          ether 02:7c:55:a7:ba:00
          inet 10.100.100.1 netmask 0xffffffff broadcast 10.100.100.255
            id 68:b5:99:6d:b9:5c priority 32768 hellotime 2 fwddelay 15
            maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
            root id 68:b5:99:6d:b9:5c priority 32768 ifcost 0 port 0
            member: bce1 flags=lc7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                      ifmaxaddr 0 port 2 priority 128 path cost 20000 proto rstp
                      role designated state discarding
            member: bce0 flags=lc7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                      ifmaxaddr 0 port 1 priority 128 path cost 20000 proto rstp
                      role designated state discarding
```

Bu bridge-in spanning tree İD-si **68:b5:99:6d:b9:5c** və prioriteti **32768**-dir.  
root İD olaraq məntiq eynidir və bu, o deməkdir ki, bu tree (ağac quruluşu) üçün root bridge-dir.

Şəbəkədə STP işə salınmış digər bridge:

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500  
  
    ether 96:3d:4b:f1:79:7a  
    id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15  
    maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200  
    root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4  
    member: bce0 flags=lc7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>  
            port 4 priority 128 path cost 200000 proto rstp  
            role root state forwarding  
    member: bcel flags=lc7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>  
            port 5 priority 128 path cost 200000 proto rstp  
            role designated state forwarding
```

Bu sətir "**root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4**" göstərir ki, root bridge **00:01:02:4b:d4:50** və bu bridge-dən olan yolun dəyəri **400000**-dir. **root bridge**-ə gedən yol isə **port 4**-dən keçir, hansı ki, bce0-dir.

## Bridge şəbəkə kartının parametrləri

Bridge şəbəkə kartlarının təyin edilməsi üçün bəzi ifconfig parametrləri unikaldır. Seksiyamızda bu parametrlərdən bəzi ümumi olanları açıqlanır. Tam siyahını ifconfig(8) man-dan əldə edə bilərsiniz.

### private

Şəxsi adlandırılmış şəbəkə kartı digər şəxsi adlandırılmış şəbəkə kartına heç bir trafik ötürmür. Trafik müzakirəsiz bağlanılır, ona görə də ARP paketləri də daxil olmaqla Ethernet çərçivə yönləndirilmir. Əgər trafik seçimə əsasən bloklanmalıdırsa, yerinə firewall istifadə edilməlidir.

### span

SPAN portu bridge portdan aldığı hər bir Ethernet çərçivəni nüsxələyərək ötürür. Bridge üzərində quraşdırıla biləcək span portun sayı limitsizdir, ancaq həmin port span kimi təyin edilərsə, adı bridge portu kimi istifadə edilə bilməz. Bu, adətən digər host tərəfindən bridge-də olan SPAN portlarının birini istifadə edərək, passiv şəkildə bridge şəbəkəsini izləmək üçün lazımlıdır. Misal üçün, bütün çərçivələrin nüsxəsini bce2 şəbəkə kartına göndərmək üçün:  
**root@Bridgel:~ # ifconfig bridge0 span bce2**

Nəticə aşağıdakı kimi olacaq:

```
root@Bridgel:~ # ifconfig bridge0
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 02:7c:55:a7:ba:00
    inet 10.100.100.1 netmask 0xffffffff broadcast 10.100.100.255
        id 68:b5:99:6d:b9:5c priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
        root id 68:b5:99:6d:b9:5c priority 32768 ifcost 0 port 0
        member: bce1 flags=lc7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                  ifmaxaddr 0 port 2 priority 128 path cost 20000 proto stp
                  role designated state forwarding
        member: bce0 flags=lc7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                  ifmaxaddr 0 port 1 priority 128 path cost 20000 proto stp
                  role designated state forwarding
member: bce2 flags=8<SPAN>
ifmaxaddr 0 port 3 priority 128 path cost 20000
```

#### **sticky**

Əgər bridge-in üzvü yapışqan (**sticky**) kimi nişanlanıbsa, dinamik öyrənilmiş ünvan yazıları forwarding (yönləndirmə) keşində, statik yazıldarda müalicə olunur. Sticky(yapışqan) yazılar heç vaxt keşdən yaşı bitmiş kimi çıxarıla, ya da dəyişdirilə bilməz, hətta ünvan digər şəbəkə kartında görünə belə. Bu statik yazı ünvanlarının yönləndirmə (**forwarding**) cədvəlinde ilkin doldurulma ehtiyacının olmaması səbəbindən üstünlük verir. Bridge-in bir seqmentindən öyrənilmiş istifadəçilər digər seqmentə köçürülə bilməz.

Sticky ünvanların istifadə edilməsinin misalı bridge ilə VLAN-ların səliqəli kombinasiyasıdır, hansı ki istifadəçi şəbəkələrini İP ünvan itirmədən izolyasiya edir. Nəzərə alaqlı ki, **CustomerA** istifadəcisi **VLAN100**-də, **CustomerB** isə **VLAN101**-dədir və bridge şəbəkə kartının İP ünvanı **192.168.0.1**-dir.

```
root@Bridgel:~ # ifconfig vlan100 create
root@Bridgel:~ # ifconfig vlan101 create
root@Bridgel:~ # ifconfig bridge0 addm vlan100 sticky vlan100 addm vlan101
sticky vlan101
root@Bridgel:~ # ifconfig bridge0 inet 192.168.0.1/24
```

Bu misalda hər iki istifadəçi üçün susmaya görə olan yol (**default gateway**) **192.168.0.1** olacaq. Bridge keşi sticky olduğuna görə bu halda bir vlan-da olan istifadəçi digər vlan-da olan istifadəçinin MAC ünvanını spoof edib trafikini ələ keçirə bilməyəcək.

VLAN-lar arasında olan istenilen əlaqəni firewall, ya da aşağıdakı misaldə olduğu kimi, **private interfeyslər** bağlaya bilər:

```
root@Bridgel:~ # ifconfig bridge0 private vlan100 private vlan101
```

Bir tərəf istifadəçi digərindən tamamilə izolyasiya edilmişdir və tam /24 ünvan aralığı da alt aralıqlara bölünmədən ayrıla bilər.

Şəbəkə kartının arxasında qalan unikal mənbə **MAC** ünvanlarının sayı limitlənə bilər. Limit öz həddinə çatan kimi mövcud olan host yazılarının keşdə olan vaxtı bitənə, ya da silinənədək bəlli olmayan mənbə ünvanlarından gələn paketlər qəbul edilməyəcək.

Aşağıdakı misal **CustomerA** (istifadəçi) üçün Ethernet alətlərində **vlan100**-ə maksimal sayı **10** edir:

```
root@Bridgel:~ # ifconfig bridge0 ifmaxaddr vlan100 10
```

Bridge şəbəkə kartlar monitor rejimini də dəstəkləyir, harada ki, paketlər **bpf(4)** emalından sonra atılır və emal edilmir, ya da növbətisi üçün ötürülmür. Bu, iki və ya daha çox şəbəkə kartlarının bir **bpf(4)** axında multipleksləşdirmək üçün istifadə edilə bilər. Bu, şəbəkə qalıqlarında olan trafikin bərpa edilməsi üçün yararlıdır, hansı ki, RX/TX siqnallarını iki fərqli şəbəkə kartından ötürür. Misal üçün, dörd şəbəkə kartından gələn məlumatı bir axın kimi oxumaq üçün:

```
root@Bridgel:~ # ifconfig bridge0 addm bce0 addm bcel addm bce2 addm bce3  
monitor up  
root@Bridgel:~ # tcpdump -i bridge0
```

# Link Aggregation və Failover

FreeBSD-də **lagg(4)** interfeys var, hansı ki, sayesində çoxlu şəbəkə kartlarını səliqə ilə bir virtual şəbəkə kartı vasitəsilə birləşdirmək olur. Bu, bize rəddə davamlılıq və link birləşdirilməsi şəraiti yaradır. Aqreqasiya edilmiş ən azı bir şəbəkə kartının üzərində uğurlu qoşulma mövcud olduğu halda, Failover trafikin gedişinin davam etməsinə izin verəcək.

Link aggregation LACP protokolunu dəstəkləyən switch-lərlə çox yaxşı işləyir. LACP protokolu linklərdən individual xəbər gələn müddətədək, trafikin hər iki istiqamətdə getməsini bölməlidir.

Aggregation protokolları lagg interfeys tərəfindən dəstəklənir. Lagg interfeys hansı portların çıxan trafik üçün və hansı portların daxil olan trafik üçün istifadə edilməsini təyin edir. Aşağıdakı rejimlər **lagg(4)** tərəfindən dəstəklənir:

## **failover**

Bu rejim trafiki yalnız master port üzərində göndərir və qəbul edir. Əgər master portu görünməz olarsa, digər aktiv port istifadə ediləcək. Virtual şəbəkə kartına əlavə edilmiş ilk fiziki şəbəkə kartı master portdur və ardıcıl növbəti əlavə edilmiş fiziki şəbəkə kartlarını rəddə davamlılıq alətləri kimi istifadə ediləcək. Əgər master olmayan portun üzərinə rəddə davamlılıq səbəbindən keçid baş veribsə, orijinal olan master portu yenidən mövcud olan kimi master rejimine keçir.

## **fec/loadbalance**

Cisco Fast EtherChannel(FEC) köhnə Cisco switch-lərində təpilir. Bu statik qurulma imkanı verir və linkin monitoringi üçün aqreqasiya danışçılarını, ya da çərcivələrlə mübadiləni qonşu ilə aparmır. Əgər switch LACP dəstəkləyirsə, onu istifadə eləmək lazımdır.

## **lacp**

IEEE 802.3ad Link Aggregation Protocol(LACP) bir və ya bir neçə Link Aqreqasiya qrupları(LAG-lar) ilə çoxlu birləşdirilə biləcək imkanlarla razılığa gələ bilir. Hər bir LAG eyni sürətli portlardan ibarətdir, hansı ki, full-dupleks rejimində işləyir və trafik LAG-da yüksək ümumi sürətlə keçirilərək bölüşdürürlür. Qaydaya əsaslanıq, bir LAG portu olur və üstündə bütün portları daşıyır. Fiziki qoşulmada olan dəyişiklik hadisəsində, LACP cəld olaraq yeni quruluşa uyğunlaşacaq.

LACP aktiv portlar üzərindən gedən çıxış trafikini keşlənmiş protokol başlığına əsasən balanslaşdırır və giriş trafikini istənilən aktiv portdan qəbul edir. Hash (keş)-ə daxildir: Ethernet **mənbə/mənsəb** ünvanları, əgər mövcuddursa, **VLAN tag** və IPv4/IPv6 **mənbə/mənsəb** ünvanları.

## **roundrobin**

Bu rejim çıkış trafikini round-robin planlayıcısını istifadə edərək bütün aktiv portlar üzərində bölüşdürürlər və giriş trafikini istənilən aktiv port üzərindən qəbul edir. Ethernet çərçivələrinin qaydalarını pozduğuna görə bu rejimi ehtiyatlı istifadə eləmək lazımdır.

## **Quraşdırma nüsxələri**

Bu seksiya Cisco switch-in və FreeBSD serverin LACP yük balanslaşdırması üçün qurulmasını nümayiş edir. Sonra da göstərir ki, necə Ethernet şəbəkə kartlarını failover rejimdə quraşdırmaq lazımdır. Həmçinin Ethernet və wifi şəbəkə kartları arasında rəddə davamlılıq quraşdırılması nümayiş ediləcək.

### Misal 1.1: Cisco Switch vasitəsilə LACP aggregation

Bu misal FreeBSD maşında olan iki bce(4) tipli fiziki Ethernet şəbəkə kartını Cisco switch üzərində olan iki GigabitEthernet portlara tək yük bölüşdürücü və link qırılmasına davamlı kimi qoşacaq. Keçirtmə qabiliyyətinin artırılması və yükün bölüşdürülməsi üçün çoxlu şəbəkə kartları əlavə edilə bilər. Aşağıdakı misalda Cisco portlarını, fiziki Ethernet adlarını(server tərəfdə), channel group rəqəmini və misalda göstərilən IP ünvanını öz daxili strukturunuza əsasən dəyişin.

Bir interfeys üçün maksimal sürəti limitləməklə Ethernet linklərin üzərində olan çərçivələrin qaydalaşdırılması mütləqdir və iki stansiya arasında olan istənilən trafik həmişə eyni fiziki link üzərindən axır. Öturmə alqoritmi daha çox informasiya istifadə etməyə çalışır ki, müxtəlif tip trafik arasında olan fərqi təyin etsin və axını mövcud olan interfeyslər arasında balanslaşdırırsın.

Cisco switch üzərində **GigabitEthernet1/0/1** və **GigabitEthernet1/0/2**-ni **channel group 1**-ə əlavə edək:

```
Switch#configure terminal
Switch(config)#interface gigabitethernet1/0/1
Switch(config-if)#channel-group 1 mode active
Switch(config-if)#channel-protocol lacp
Switch(config-if)#
Switch(config-if)#interface gigabitethernet1/0/2
Switch(config-if)#channel-group 1 mode active
Switch(config-if)#channel-protocol lacp
```

FreeBSD maşında isə **bce0** və **bcel** fiziki şəbəkə kartları istifadə edərək **lagg(4)** interfeys yaradın və IP ünvanı **10.0.0.3/24** təyin edin:

```
root@BridgeL:~ # ifconfig bce0 up
root@BridgeL:~ # ifconfig bcel up
root@BridgeL:~ # ifconfig lagg0 create
root@BridgeL:~ # ifconfig lagg0 up laggproto lacp laggport bce0 laggport
bcel 10.0.0.3/24
```

Sonra virtual şəbəkə kartının statusunu yoxlayın:

```
root@BridgeL:~ # ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=c01bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,TSO4,VLAN_HWTSO,LINKSTATE>
ether 68:b5:99:6d:b9:5c
inet 10.0.0.3 netmask 0xffffffff broadcast 10.0.0.255
media: Ethernet autoselect
status: active
laggproto lacp lagghash 12,13,14
laggport: bcel flags=lc<ACTIVE,COLLECTING,DISTRIBUTING>
laggport: bce0 flags=lc<ACTIVE,COLLECTING,DISTRIBUTING>
```

**ACTIVE** nişanla qeydə alınan portlar LAG-ın hissəsidir, hansı ki, uzaq switch-lə razılışmaya gelmişdir. Trafik bu aktiv portlar üzərində ötürüləcək və qəbul ediləcək. LAG identifikasiatorlarına baxmaq üçün isə öncəki əmrə **-v** opsiyasını artırmaq lazımdır. Aşağıdakı kimi:

```
root@BridgeL:~ # ifconfig -v lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=c01bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_
HWCSUM,TSO4,VLAN_HWTSO,LINKSTATE>
```

```

ether 68:b5:99:6d:b9:5c
inet 10.0.0.3 netmask 0xffffffff broadcast 10.0.0.255
media: Ethernet autoselect
status: active
groups: lagg
lagproto lacp lagghash 12,13,14
lag id: [(8000,68-B5-99-6D-B9-5C,010B,0000,0000),
           (8000,E8-ED-F3-8D-AA-80,0001,0000,0000)]
laggport: bcel flags=lc<ACTIVE,COLLECTING,DISTRIBUTING> state=3D
           [(8000,68-B5-99-6D-B9-5C,010B,8000,0002),
            (8000,E8-ED-F3-8D-AA-80,0001,8000,0103)]
laggport: bce0 flags=lc<ACTIVE,COLLECTING,DISTRIBUTING> state=3D
           [(8000,68-B5-99-6D-B9-5C,010B,8000,0001),
            (8000,E8-ED-F3-8D-AA-80,0001,8000,0102)]

```

Cisco switch tərəfdə portun statusuna baxmaq üçün isə aşağıdakı əmrdən istifadə edə bilərsiniz:

**Switch#show lacp neighbor**

Flags: S - Device is requesting Slow LACPDU  
          F - Device is requesting Fast LACPDU  
          A - Device is in Active mode         P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:

	LACP port				Admin	Oper	Port	Port
Port	Flags	Priority	Dev ID	Age	key	Key	Number	State
Gil/0/1	SA	32768	68b5.996d.b95c	22s	0x0	0x10B	0x1	0x3D
Gil/0/2	SA	32768	68b5.996d.b95c	22s	0x0	0x10B	0x2	0x3D

Detallı məlumat əldə etmək üçün isə **show lacp neighbor detail** əmrini daxil etmək lazımdır. Bu quraşdırılmalarımızın sistem yenidənyüklənməsindən sonra işləməsini istəsək, aşağıdakı sətirləri **/etc/rc.conf** faylinə əlavə etmək lazımdır:

```

ifconfig_bce0="up"
ifconfig_bcel ="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="lagproto lacp laggport bce0 laggport bcel 10.0.0.3/24"

```

### Misal 1.1: Failover rejimi

Əgər master şəbəkə kartında problem yaranarsa, avtomatik olaraq linki ikinci dərəcəli şəbəkə kartının üstünə keçirmək tələbi yaranarsa, failover rejimi bu tələbi qarşılıyır. Failover şəbəkə kartlarını quraşdırmaq üçün əmin olun ki, fiziki şəbəkə kartları işlək vəziyyətdədir, sonra da **lagg(4)** şəbəkə kartını yaradın. Bu misalımızda **bce0** şəbəkə kartımız **master** (əsas), **bce1** isə ikinci dərəcəlidir. Virtual şəbəkə kartımıza isə **10.0.0.15/24** IP ünvanı təyin edilmişdir (Bu halda Cisco tərəfdə heç bir quraşdırma edilmir, sadəcə serverimizə aid olan hər iki portun eyni VLAN-da olması kifayətdir):

```
root@Bridgel:~ # ifconfig bce0 up
root@Bridgel:~ # ifconfig bce1 up
root@Bridgel:~ # ifconfig lagg0 create
root@Bridgel:~ # ifconfig lagg0 up laggproto failover laggport bce0 laggport
bce1 10.0.0.15/24
```

Virtual şəbəkə kartı çıxışda aşağıdakı məlumatı çap etməlidir:

```
root@Bridgel:~ # ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=c01bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_
HWCSUM,TSO4,VLAN_HWTSO,LINKSTATE>
    ether 68:b5:99:6d:b9:5c
    inet 10.0.0.15 netmask 0xffffffff broadcast 10.0.0.255
        media: Ethernet autoslect
        status: active
    laggproto failover lagghash 12,13,14
    laggport: bce1 flags=0<>
    laggport: bce0 flags=5<MASTER,ACTIVE>
```

Trafik **bce0** şəbəkə kartı üzərindən daxil olur və çıxır. Əgər **bce0** şəbəkə kartı sıradan çıxarsa, **bce1** avtomatik olaraq, **ACTIVE** link statusuna keçəcək və işə düşəcək. Əgər master şəbəkə kartı yenidən bərpa edilərsə, o, yenidən aktiv linkə çevriləcək. Aşağıdakı kimi:

```
root@Bridgel:~ # ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=c01bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_
HWCSUM,TSO4,VLAN_HWTSO,LINKSTATE>
    ether 68:b5:99:6d:b9:5c
    inet 10.0.0.15 netmask 0xffffffff broadcast 10.0.0.255
        media: Ethernet autoslect
        status: active
```

```
laggproto failover lagghash 12,13,14  
laggport: bce1 flags=4<ACTIVE>  
laggport: bce0 flags=1<MASTER>
```

Etdiyimiz quraşdırımların sistem yenidənyüklənməsindən sonra avtomatik işə düşməsini istəyiriksə, aşağıdakı sətirləri **/etc/rc.conf** faylına əlavə etmək lazımdır:

```
ifconfig_bce0="up"  
ifconfig_bce1 ="up"  
cloned_interfaces="lagg0"  
ifconfig_lagg0="laggproto failover laggport bce0 laggport bce1 10.0.0.15/24"
```

#### Misal 1.3: Ethernet və Wireless şəbəkə kartları arasında failover rejimi

Adətən laptop istifadəçiləri üçün istənilən odur ki, wifi şəbəkə kartlarını ikinci dərəcəli kimi təyin etsinlər və yalnız Ethernet şəbəkə kartında problem olarsa, o, avtomatik olaraq işə düşsün. **lagg(4)** vasitəsilə bütün trafiki təhlükəsizlik/üstünlük baxımından Ethernet üzərindən ötürmək və rezerv üçün wireless qoşulmasını istifadə etmək olur. Biz bu işi Ethernet şəbəkə kartının MAC ünvanını Wifi şəbəkə kartına təyin etməklə edə bilərik.

Bu misalda Ethernet şəbəkə kartımızın adı **bge0**-dir, hansı ki, master-dir və wireless şəbəkə kartı **wlan0** işə failover (rəddədəvamlı) olandır. **wlan0** avadanlığı **iwn0** wireless şəbəkə kartından yaradılmışdır, hansı ki, Ethernet kartının şəbəkə kartının MAC ünvanı ilə quraşdırılacaq. İlk olaraq, Ethernet şəbəkə kartının MAC ünvanını təyin edək:

```
root@Bridgel:~ # ifconfig bge0  
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500  
    options=19b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWSUM,TSO4>  
ether 00:21:70:da:ae:37  
inet6 fe80::221:70ff:feda:ae37%bge0 prefixlen 64 scopeid 0x2  
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>  
    media: Ethernet autoselect (1000baseT <full-duplex>)  
    status: active
```

**bge0** adını öz sisteminizdə olanla dəyişin. **ether** olan sətirdə **MAC** ünvan görünür. İndi işə həmin MAC ünvanı aşağıdakı wifi şəbəkə kartına təyin edək:

```
root@Bridgel:~ # ifconfig iwn0 ether 00:21:70:da:ae:37
```

Wireless şəbəkə kartını işə salaq, amma IP ünvanı təyin etməyək:

```
root@Bridgel:~ # ifconfig wlan0 create wlandev iwn0 ssid my_router up
```

Əmin olun ki, **bge0** şəbəkə kartı qalxmışdır və sonra **lagg(4)** şəbəkə kartı yaradın, hansı ki, **bge0** olacaq, master **wlan0** işə failover:

```
root@Bridgel:~ # ifconfig bge0 up
root@Bridgel:~ # ifconfig lagg0 create
root@Bridgel:~ # ifconfig lagg0 up laggproto failover laggport bge0 laggport
wlan0
```

Virtual şəbəkə kartının görünüşü aşağıdakı kimi olacaq:

```
root@Bridgel:~ # ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=8<VLAN_MTU>
    ether 00:21:70:da:ae:37
    media: Ethernet autoselect
    status: active
    laggproto failover
    laggport: wlan0 flags=0<>
    laggport: bge0 flags=5<MASTER,ACTIVE>
```

Sonra DHCP client-i işə salın ki, **lagg0** şəbəkə kartı IP ünvanı əlsin:

```
root@Bridgel:~ # dhclient lagg0
```

Bu quraşdırmalarımızın sistem yenidənyüklənməsindən sonra avtomatik işə düşməsini istəsək, aşağıdakı sətirləri **/etc/rc.conf** StartUP faylımiza əlavə etmək lazımdır:

```
ifconfig_bge0="up"
ifconfig_iwn0="ether 00:21:70:da:ae:37"
wlans_iwn0="wlan0"
ifconfig_wlan0="WPA"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto failover laggport bge0 laggport wlan0 DHCP"
```

# Common Address Redundancy Protocol (CARP)

Common Address Redundancy Protocol (**CARP**) çoxlu maşına imkan verir ki, bir və ya bir neçə xidmet üçün eyni IP ünvanı və Virtual Host ID (**VHİD**)-ni paylaşaraq dayanıqlıq yaratsın. Bu, o deməkdir ki, bir və ya bir neçə ünvan sıradan çıxa bilər, ancaq digər hostlar bütün servisləri özlərindən elə cavablandıracaqlar ki, istifadəçilər bunu hiss etməsin.

Paylaşılmış IP ünvandan başqa hər bir hostun özünün IP ünvanı olur, hansı ki, idarəetmə və quraşdırma üçün lazımdır. Bütün IP paylaşan maşınların eyni VHİD-i olur. Hər bir Virtual IP ünvan üçün şəbəkə kartının broadcast domain əhatəsində VHİD unikal olmalıdır.

FreeBSD yüksək davamlılıq üçün CARP istifadə edir. Ancaq onun qurulması FreeBSD versiyasında fərqli olsa da, bu başlığımızda istənilən versiya üçün quraşdılmalar açıqlanacaq.

Bu misalımızda dayanıqlılıq üçün 3 ədəd hostdan istifadə edirik. Hər birinin unikal IP ünvanı mövcuddur, ancaq eyni web kontenta malikdirlər. İki fərqli master mövcuddur, biri **hosta.example.org** və **hostb.example.org**, hansı ki, paylaşılmış adları **hostc.example.org** olacaq.

Bu maşınların yük bölgüleri DNS-in Round Robin quraşdırması ilə təmin edilmişdir. Master və rezerv nüsxə maşınları idarəetmə IP ünvanları və host adlarından başqa identik quraşdırılmışdır. Bu serverlərdə eyni quraşdırma olmalıdır və eyni xidmətlər işləməlidir. Əgər failover (rəddə davamlılıq) baş verərsə və rezerv serverdə də eyni informasiya kontentinə yetki varsa, yalnız paylaşılmış IP ünvana gələn müraciətlər düzgün cavablandırıla bilər. Rezerv maşının iki əlavə CARP interfeysi mövcuddur, master kontent daşıyan server IP ünvanlarının hər biri üçün bir ədəd. Nasazlıq baş verən kimi, rezerv serveri sıradan çıxan master maşından IP ünvanı alacaq.

## FreeBSD10, ya da daha yeni versiyada CARP-in istifadə edilməsi

Sistemin işə düşməsi zamanı CARP-i işə salmaq istəsəniz, kernel modulu olan **carp.ko**-nu **/boot/loader.conf** faylına əlavə etmək lazımdır:

```
carp_load="YES"
```

Yenidənyüklənmə etmədən modulu yükleyirik:

```
# kldload carp
```

Əgər kernel-in içinə əlavə etmək istəsəniz, aşağıdakı sətri kernel faylına əlavə edib kompilyasiya etmək lazımdır:

```
device carp
```

Host adı, idarəetmə İP ünvanı və şəbəkə maskası, paylaşılmış İP ünvanı və VHİD quraşdırılmalarını **/etc/rc.conf** faylına əlavə edirik. Maşın master-dir. Aşağıdakı nüsxələr **hosta.example.com** üçündür:

```
hostname="hosta.example.com"
ifconfig_em0="inet 192.168.121.136/24"
ifconfig_em0_alias0="vhid 1 pass freebsd alias 192.168.121.200/32"
```

Sonra isə **hostb.example.com** üçün verilənləri əlavə edək. Bu maşın başqa **masteri** təmsil etdiyinə görə fərqli paylaşılmış İP ünvanı və VHİD istifadə edir. Ancaq təyin etdiyiniz şifrə ilə diqqətli olun ki, eyni VHİD üçün backup maşında da eyni olsun, çünki CARP xəbərdarlığı şifrə doğru olduğu halda yollayır.

```
hostname="hostb.example.com"
ifconfig_em0="inet 192.168.121.137/24"
ifconfig_em0_alias0="vhid 2 pass cavid123 alias 192.168.121.201/32"
```

Üçüncü maşın isə, **hostc.example.com**-dur, hansı ki, hər iki maşın üçün rəddə davamlı olan server işini görür. Bu server 2 ədəd VHİD ilə quraşdırılmışdır ki, hər bir master maşını ayrıca Virtual İP ünvan ilə emal eləsin. CARP avadanlığı **skew** və **advskew** təyin edir ki, əmin olsun rezerv maşın xəbəri master olandan gec ötürür. Ancaq çoxlu rezerv serverlər olarsa, advskew öncəliyi kontrol edir.

```
hostname="hostc.example.com"
ifconfig_em0="inet 192.168.121.138/24"
ifconfig_em0_alias0="vhid 1 advskew 100 pass freebsd alias 192.168.121.200/32"
ifconfig_em0_alias1="vhid 2 advskew 100 pass cavid123 alias 192.168.121.201/32"
```

İki ədəd CARP VHİD-in olması o deməkdir ki, **hostc.example.com** maşını iki ədəd master olan maşının hər birini ayrılıqla əvəz edə bilir. Əgər master serverlərdən biri dayanarsa, avtomatik olaraq bütün **hostc.example.com** maşını master statusunu özünə götürəcək. Əgər master serveri rezervdən önce özünü elan etdikdə problem yaranarsa, yenidən master serveri işlək vəziyyətə gələnədək rezerv serveri Virtual İP-ni özünə götürəcək.

**Qeyd:** Üstünlük susmaya görə sönülu olur. Əgər **preemption** (üstünlük) işe salınarsa, **hostc.example.com** maşını virtual İP ünvanı azad edib orijinal maşına geri qaytara bilməyəcək. İnzibatçı sərt olaraq, master maşına əmr ötürərək deyə biler ki, İP ünvanını digər maşına qaytarınsın:

```
# ifconfig em0 vhid 1 state backup
```

Quraşdırmanız bitdikdə hər bir serverə, ya da şəbəkəyə yenidənyüklənmə əmri ötürün.

CARP funksionallığı çoxlu **sysctl(8)** dəyişənləri vasitəsilə **carp(4)** man səhifələrindən oxunulub idarə edilə bilər. CARP xəbərdarlıqlarından digər məlumatları **devd(8)** istifadə edərək almaq olar.

## CARP-in FreeBSD9 və daha köhnə versiyalarda istifadə edilməsi

Bu versiya FreeBSD serverlər üçün də məntiq eynidir və öncəki seviyədə məntiqi açıqlanmışdır. Yalnız burada siz aşağıda göstərilən quraşdirmalara əsasən öncə carp aləti yaratmalısınız.

Sistem yenidənyüklənməsindən sonra işə düşməsini kernel modulu **if\_carp.ko** çağırı bilərsiniz. Bunun üçün **/boot/loader.conf** faylına aşağıdakı sətri əlavə etmək lazımdır:

```
if_carp_load="YES"
```

Yenidənyüklənmə etmədən modulu yüklemək üçün isə aşağıdakı əmri CLI-də daxil etməniz kifayətdir:  
**# kldload carp**

Əgər modulu kernel-in daxilində istifadə etmək istəyirsinizsə, aşağıdakı sətri seçdiyiniz kernel quraşdırma faylına əlavə edib, yenidən kompilyasiya etməyiniz kifayətdir:

```
device carp
```

Sonra hər bir maşında CARP alətini yaradın:

```
# ifconfig carp0 create
```

Tələb edilən sətirləri **/etc/rc.conf** faylına əlavə etmək lazımdır. Yəni host adı, idarə etmə İP ünvanı, paylaşılmış İP ünvan və **VHİD**. Ancaq burada alias əvəzinə CARP aləti Virtual İP istifadə

edəcək və bu səbəbdən şəbəkə aralığı /32 yox, /24 olacaq. Aşağıda `hosta.example.com` üçün verilənlər təqdim edilir:

```
hostname="hosta.example.org"
ifconfig_bce0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass freebsd 192.168.1.50/24"
```

`hostb.example.org`-da isə aşağıdakı sətirləri `/etc/rc.conf` faylına əlavə edirik:

```
hostname="hostb.example.org"
ifconfig_bce0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass cavid123 192.168.1.51/24"
```

Üçüncü maşın `hostc.example.com` isə quraşdırılır ki, master maşınların qəza keçidində avtomatik olaraq bu maşın işə düşsün:

```
hostname="hostc.example.org"
ifconfig_bce0="inet 192.168.1.5 netmask 255.255.255.0"
cloned_interfaces="carp0 carp1"
ifconfig_carp0="vhid 1 advskew 100 pass testpass 192.168.1.50/24"
ifconfig_carp1="vhid 2 advskew 100 pass cavid123 192.168.1.51/24"
```

**Qeyd:** **Preemption** (üstünlük) susmaya görə olan FreeBSD GENERIC kernel-də söndürülümüşdür. Əgər öz seçdiyiniz kernel-də preemption işə salınarsa, `hostc.example.com` maşını IP ünvanı geri orijinal server üçün azad edə bilməyəcək. İnzibatçı özü sərt olaraq əmrlə təyin edə bilər ki, maşın rezerv(backup) statusa keçsin. Bu, aşağıdakı əmrlə edilir:

```
# ifconfig carp0 down && ifconfig carp0 up
```

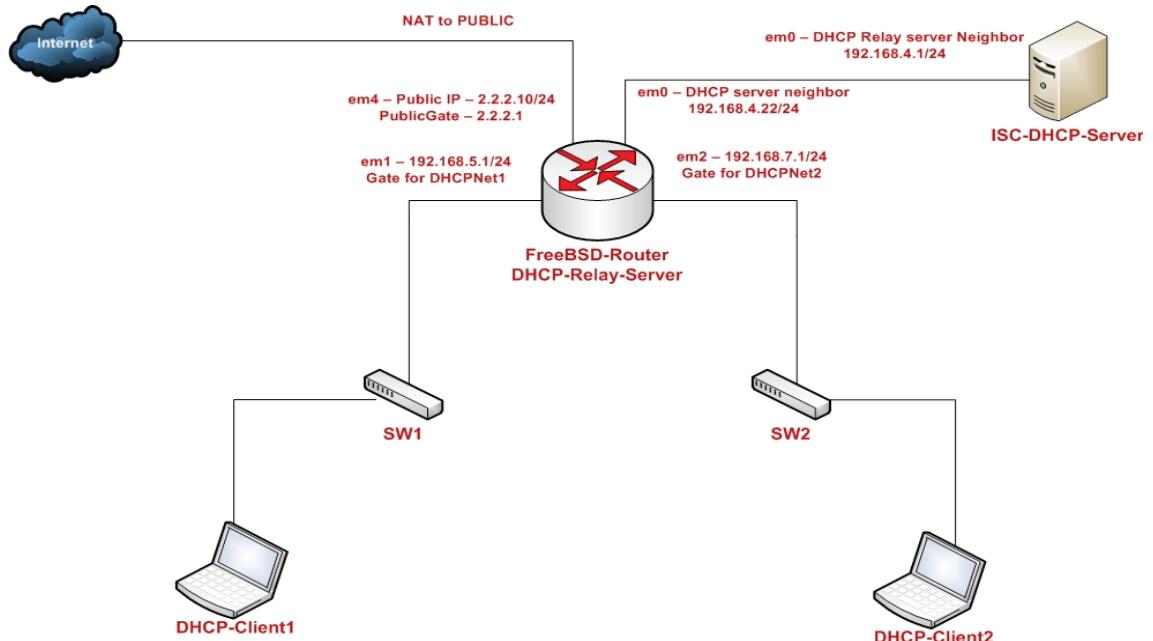
Bu tələbə uyğun olan maşının düzgün adlı CARP alətində edilməlidir.

Quraşdırmanız bitdikdən sonra şəbəkə servislərinizi, ya da hər bir sistemi yenidənyüklənmə edin. Yüksək davamiyyət artıq aktivdir.

## FreeBSD DHCP Relay quraşdırılması

İşimizin görülməsi üçün şəbəkəmizdə bir ədəd FreeBSD9.2 x64 (**Gateway** və **DHCP relay**) server, bir ədəd **DHCP server** və iki müxtəlif şəbəkədə yerləşən istifadəçilər var. Lazımı ardıcılıqla işimizin görülməsi üçün önce DHCP relay serveri hazırlayaq.

### 1. DHCP Relay serverimiz aşağıdakı şəbəkə quruluşunda işləyəcək.



Relay serverimzdə NAT-ın işleməsi üçün kernel-i aşağıdakı imkanlarla kompilyasiya edin.

```
options      IPDIVERT
options      IPFIREWALL
options      IPFIREWALL_NAT
options      IPFIREWALL_FORWARD
options      IPFIREWALL_VERBOSE
options      IPFIREWALL_VERBOSE_LIMIT=3
```

Relay serverimzdə lazımi paketi yükləyək və onu StartUP-a əlavə edək.

```
root@dhcprelay:~ # cd /usr/ports/net/dhcprelay/
- PORT ünvanına daxil oluruq.
root@dhcp:/usr/ports/net/dhcprelay # make install
- Yükləyirik.
```

StartUP quraşdırma faylımız aşağıdakı kimi olacaq(yəni /etc/rc.conf faylı):

```
sshd_enable="YES"
dumpdev="NO"
ifconfig_em0="inet 192.168.4.22 netmask 255.255.255.0"
ifconfig_em1="inet 192.168.5.1 netmask 255.255.255.0"
ifconfig_em2="inet 192.168.7.1 netmask 255.255.255.0"
ifconfig_em4="inet 2.2.2.10 netmask 255.255.255.0"
defaultrouter="2.2.2.1"
hostname="dhcprelay"
firewall_enable="YES"
firewall_type="OPEN"
natd_enable="YES"
natd_interface="em4"
gateway_enable="YES"
dhcprelya_enable="YES"
dhcprelya_servers="192.168.4.1"          # DHCP Server IP
dhcprelya_ifaces="em1 em2"
```

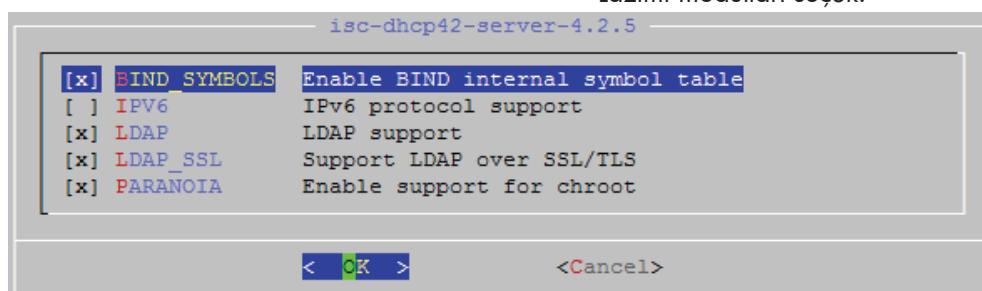
Sonda **DHCP Relay** daemon-u işə salaq.

```
root@dhcp:~ # /usr/local/etc/rc.d/dhcprelya start
```

## 2. Artıq DHCP Serverimizin özünü hazırlayaq

Öncə lazımi paketləri yükləyək.

```
root@dhcpserver:~ # cd /usr/ports/net/isc-dhcp42-server/
                    - Port ünvanına daxil olaq.
root@dhcpserver:/usr/ports/net/isc-dhcp42-server # make config
                    - Lazımi modulları seçək.
```



```
root@dhcpserver:/usr/ports/net/isc-dhcp42-server # make install
                    - Yükləyək.
```

DHCP serverimizdə StartUP quraşdırma faylı aşağıdakı kimi olacaq (yəni `/etc/rc.conf` faylı):

```
sshd_enable="YES"
dumpdev="NO"
ifconfig_em0="inet 192.168.4.1 netmask 255.255.255.0"
hostname="dhcpserver"
gateway_enable="YES"
dhcpd_enable="YES"
dhcpd_ifaces="em0"
dhcpd_flags="-q"
dhcpd_conf="/usr/local/etc/dhcpd.conf"

# DHCP Relay serverə qayıdış üçün routing yazırıq.
static_routes="dhcpnet1 dhcpnet2"
route_dhcpnet1="-net 192.168.5.0/24 192.168.4.22"
route_dhcpnet2="-net 192.168.7.0/24 192.168.4.22"
```

Ardınca DHCP serverimizin quraşdırma faylini yaradaq. (`/usr/local/etc/dhcpd.conf`):

```
option domain-name "bsdrp.net";
option domain-name-servers 8.8.8.8, 8.8.4.4;
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
subnet 192.168.4.0 netmask 255.255.255.0 {
}

subnet 192.168.5.0 netmask 255.255.255.0 {
    range 192.168.5.100 192.168.5.120;
    option routers 192.168.5.1;
}

subnet 192.168.7.0 netmask 255.255.255.0 {
    range 192.168.7.100 192.168.7.120;
    option routers 192.168.7.1;
}
```

root@dhcpserver:/ # `/usr/local/etc/rc.d/isc-dhcpd start`

- Sonda DHCP serveri işə salırıq və istifadəçilərdə yoxlayırıq.



# BÖLÜM 15

## FreeBSD inzibatçı üçün vacib olanlar

- / İstifadəçilər və qrupların idarə edilməsi.
- / Yetki hüquqlarının idarə edilməsi
- / Access Control List
- / Spesifik flaglar

Bu başlıq inzibatçının gündəmdə tələb edilən ən vacib məqamlarını açıqlayır. İstifadəçi və qrupların tam detallı işləmə prinsipləri, istifadəçilər üçün resurslara təyin edilən məhdudiyyət, istifadəçilərin hesabatlarının əldə edilməsi, fayl və qovluqlarda olan yetkilərin idarəedilməsinin detallı açıqlanması, fayl sistemdə olan fayllara yetki hüquqlarının kontrolünün qurulması, spesifik flagların təyin edilməsi (bu, təhlükəsizlik üçün çox yararlıdır) bu başlıqda izah olunacaq.

# İstifadəçilər və qrupların idarə edilməsi

İstifadəçi və faylların özlerinə aid olan hüquqlar elə məhz OS UNIX konsepsiyasını formalasdırır. FreeBSD OS çox istifadəcili olduğuna və ümumi UNIX sinfinə daxil edildiyinə görə istifadəçilərin düzgün idarə edilməsində çox önemlidir.

İstifadəçi və qrupların qeydiyyat yazıları iki faylda saxlanılır:

- **/etc/master.passwd** - bu faylda istifadəçilərin qeydiyyat verilənləri və kodlaşdırılmış şəkildə şifrələri saxlanılır.
- **/etc/group** - qruplara cavabdeh fayl.

FreeBSD-də kölgəli parollar texnologiyası istifadə edilir – bu, o halda olur ki, istifadəçilərin sistem verilənləri iki yerə bölünür:

1. **/etc/master.passwd** faylı, hansında ki, şifrələr kodlaşdırılmış şəkildə qalır və ancaq root istifadəçisinin bu faylı oxuma və yazma yetkisi olur.
2. **/etc/passwd** faylı, hansı ki, **pwd\_mkdb(8)** (**sifrələrlə olan bazanın generasiyası**) əmri ilə **/etc/mastwer.passwd** faylından yaradılmışdır. Bu faylda bütün qruplarda olan istifadəçilər üçün oxuma yetkisi var və şifrələr \* simvolu ilə təyin edilmişdir. Həmçinin **pwd\_mkdb(8)** əmrinin köməkliyi ilə **/etc/master.passwd** faylından iki ədəd yeni fayl yaradılır - **/etc/pwd.db** və **/etc/spwd.db** (verilənlər bazasının indeksləşdirilməsi). Əgər sisteminizdə həddən artıq istifadəçi yaradılsara, onların axtarışının sürətli olmasını məhz bu iki fayl edir. **/etc/spwd.db** faylı elə **/etc/master.passwd** faylı kimi gizlidir, həmçinin eyni yetki və sahibə malikdir.

## /etc/master.passwd faylinin sintaksisine baxaq

```
root:$6$dNi.44AaLBdCyitt$p9W4qCGQZTwX1EwtoleAbHs/t7VV76WJ4tAYGF.  
dIoREEK.2UvKxkhFImfP.uvg5kKNb6iMN2uZrOKLwE/Acb1:0:0::0:Charlie &:/bin/csh  
toor:*:0:0::0:0:Bourne-again Superuser:/root:  
daemon:*:1:1::0:0:Owner of many system processes:/root:/usr/sbin/nologin  
operator:*:2:5::0:0:System &:/usr/sbin/nologin  
bin:*:3:7::0:0:Binaries Commands and Source:/:/usr/sbin/nologin  
tty:*:4:65533::0:0:Tty Sandbox:/:/usr/sbin/nologin  
kmem:*:5:65533::0:0:KMem Sandbox:/:/usr/sbin/nologin  
games:*:7:13::0:0:Games pseudo-user:/usr/games:/usr/sbin/nologin
```

Faylda olan hər bir sətr ayrılıqda bir istifadəçini açıqlayır və özündə sütunları ayırmak üçün ":" simvolu istifadə edir.

### Qayda ilə sütunları açıqlayaq:

1. **name** - İstifadəçi logini, hansı ki, sistemə daxil olduqda istifadə edilir.
2. **password** - **/etc/master.passwd** faylında kodlaşdırılmış şifrə və \* ilə **/etc/passwd** faylında.
3. **uid** - İstifadəçinin unikal identifikatoru.
4. **gid** - Qrupun unikal identifikatoru.
5. **class** - **/etc/login.conf** faylından götürülen, quraşdırılma və yüklənmə class-dır.
6. **Change** – Şifrənin yaşama müddəti. Bu müddətdən sonra şifrə dəyişməlidir. Vaxt saniyelərlə olur və 1 yanvar 1970-ci ildən bəri hesablanır. Vaxtı təyin etmək üçün isə, bu sütunda yerləşən rəqəmi **date** əmrində istifadə etmək olar: **date -r seconds**, seconds olan yer elə sütunda olan rəqəmdir.
7. **expire** - İstifadəçi adının yaşama müddəti, bu müddət bitdikdən sonra istifadəçi adı bağlanacaq. Vaxt 1 yanvar 1970-ci ildən saniyelərlə hesablanır. Bu sütunda olan saniyelərin hansı tarixə göstərildiyinə bu əmrlə baxa bilərik: **date -r seconds**, seconds olan yer elə sütunda olan rəqəmdir.
8. **Gecos** - İstifadəçi haqqında ümumi məlumat.
9. **homedir** - İstifadəçinin ev qovluğu.
10. **shell** - mühit, hansı ki, istifadəçi tərəfindən istifadə ediləcək.

**/etc/passwd** faylinin yaradılmasında **/etc/master.passwd** faylından **class,change,expire** sütunları silinir və şifrənin yerinə \* simvolu əlavə edilir.

**Login** (**name**) sütunu (-) tire simvolu ile başlaya bilməz, həmçinin gündəlikdə istifadə edilməyən simvollar və loginin(.) nöqtə simvolu ilə ayrılması məsləhət deyil (Bu maille işləyəndə problem yarada bilər). **/etc/master.passwd** faylında **password** sütunu şifrələnmiş olur, əgər sütun yoxdursa, onda şifrənin əvəzinə \* simvolu durur və bu istifadəçi maşına yetki ala bilməyəcək. **master.passwd** faylinin tez redaktə edilməsi üçün **pwd\_mkdb(8)** deyil, **vipw(8)** əmri istifadə edilir. Bu, elə **vi(8)** redaktorunun özüdür. (Ona görə də **vipw** əmrinin istifadə edilməsindən önce **vi** əmri ilə tanış olun)

Məsələn:

```
root@mercuri:~ # vipw
```

```
root:$6$dNi.44AaLBdCyitt$p9W4qCGQZTwX1EwtoleAbHs/t7VV76WJ4tAYGF.  
dIoREEK.2UvKxkhFImfP.uvg5kKNb6iMN2uZrOKLwE/Acb1:0:0::0:Charlie &:/root:/bin/csh  
toor:*:0:0::0:0:Bourne-again Superuser:/root:  
daemon:*:1:1::0:0:Owner of many system processes:/root:/usr/sbin/nologin  
operator:*:2:5::0:0:System &:/usr/sbin/nologin  
bin:*:3:7::0:0:Binaries Commands and Source:/usr/sbin/nologin  
tty:*:4:65533::0:0:Tty Sandbox:/usr/sbin/nologin  
kmem:*:5:65533::0:0:KMem Sandbox:/usr/sbin/nologin  
games:*:7:13::0:0:Games pseudo-user:/usr/games:/usr/sbin/nologin
```

**/etc/master.passwd** faylında **password** sütununda \* simvolunun istifadə edilməsi nəticəsində istifadəçi sistemə giriş edə bilməyəcək, çünki şifrə kodlaşdırılmış şəkildə ola bilməz. Məsələn, istifadəçinin müvəqqəti bloklanması üçün **/etc/master.passwd** faylinin **password** sütununda **\*LOCKED\*** kombinasiyası qoymaq, ya da parol mövcuddursa, onun önünə əlavə etmək lazımdır. Yenidən istifadəçini aktivləşdirmək üçün isə sadəcə bunu silmək kifayət edər (Bunu **vipw** əmri ilə etmək lazımdır). İstifadəçini CLI-dən birbaşa block etmək üçün **pw(8)** manualını oxumaq lazımdır.

İstifadəçi haqda ümumi məlumatı təşkil edən gecos sütunu vergül ilə ayrılan aşağıdakı sütunlardan ibarətdir:

- **name** - İstifadəçinin tam adı;
- **office** - Ofisin nömrəsi;
- **wphone** - İş telefonu;
- **hphone** - Ev telefonu;
- **home\_dir** - sütunu sahibi olan istifadəçinin ev qovluğu ünvanını təyin edir.
- **shell** sütunu istifadəçinin mühitini təyin edir. İstifadəçi üçün işləmə mühitinin siyahısını **/etc/shells** ünvanından eldə edə bilərsiniz. **root** istifadəçisi üçün shell mühitini dəyişmək məsləhət deyil, çünki fayl sistemin qəza vəziyyətində **/usr** mount edilməmiş ola bilər, bu

halda root istifadəçi sistemə yetki ala bilməyəcək. Əgər siz istifadəçinin sistemə girişini qadağan etmək istəyirsinizsə, onda onun shell mühitini **/sbin/nologin** edin. Bu programlar digərlərindən daha düzgün işləyir (Məsələn: **/dev/null** istifadəçinin giriş cəhdini emal edir).

Yeni istifadəçinin əlavə edilməsində unikal ad seçmək lazımdır ki, **/etc/passwd** və "**/etc/mail/aliases**" faylında olmasın. Həmçinin login name-də **-{tire}**, **.(nöqtə)** simvollarından və yazıda istifadə etdiyiniz simvollardan burada istifadə etməyin, bu, mail-lərlə bağlı ciddi problemlərə səbəb ola bilər. Yeni istifadəçi unikal ID alır - UID və qrupun üzvü olur və qrupun adı istifadəçi adı ilə üst-üstə düşür. Qrupun bu cür adlandırılması və siyaseti daha təhlükəsizdir, ona görə ki, yetkilərin idarə edilməsində rahat imkanlar yaradır. Sistemdə olan UID və istifadəçi adı unikaldır və fayl sistem üzərində yetkilərin verilməsində istifadə ediləcək. İstifadəçinin sistemə əlavə edilməsindən sonra onun ev qovluğuna fayllar nüsxələnir. **.profile** (istifadəçi sistemə daxil olan kimi yerinə yetirilir), əgər **/bin/sh** və ya **/bin/csh** mühiti istifadə edilirsə. **.cshrc** (mühitin işə düşməsində) və **.login** (istifadəçi sistemə login olduqda) işə düşür. Bütün bu fayllar **/usr/share/skel** qovluğundan nüsxələnir.

**/etc/group** faylında sistemin daxili qrupları saxlanılır. Bu fayl istənilən mətn redaktoru ilə dəyişdirilə bilər. Yeni qrupu əlavə etmək üçün sadəcə yuxarıda göstərilən fayla lazımi standartla qrupun sətrini əlavə etmək kifayətdir.

Fayl ayrılmış sətirlərdən ibarətdir. Bu sətirlər işə öz növbəsində ":" simvolu ilə sütunları ayırrı. Sətirdə aşağıdakı sütunlar olur:

- **group** - qrupun adı;
- **password** - qrup üçün kodlaşdırılmış şifrə;
- **gid** - qrupun unikal nömrəsi;
- **member** - bu qrupun üzvləri.

Bu faylda #(diyez)simvolu ilə başlayan hər sətir şərh deməkdir.

**group** sütunu qrupun adıdır, bu qrupun üzvlüyündə olan istifadəçilərin fayllara olan yekisini təyin edir. **group** sütunu **gid** sütunu ilə uyğunlaşdırılır və bu, təyin edilmiş qrupun unikal identifikasiatorunu göstərir. Bu iki sütun ayrılmaz olaraq əlaqəlidir, eyni ilə istifadəçi adı və UID-də olduğu kimi, **.password** sütunu mütləq deyil, o, bəzən istifadə edilir və \* simvolu kodlaşdırılmış şifrədən daha yaxşıdır.

**member** sütunu qrupun üzvlərini özündə təşkil edir və istifadəçilər vergul (,) simvolu ilə ayrılır. Bir qrupda 200-dən artıq istifadəçi ola bilməz. **/etc/group** faylında olan sətin maksimal uzunluğu 1024 simvoldan yuxarı ola bilməz.

## Istifadəçi resurslarının istifadəsi və məhdudiyyət

Istifadəçi resurslarının idarə etməsi **class**-lar vasitəsilə yerinə yetirilir, hansı ki, **/etc/login.conf** faylında təyin edilir və həmçinin istifadəçi yaradılmasında əlavə edilə bilir. Əger istifadəçi üçün hansısa bir class müəyyən edilmirsə, onda ona **default** class-ı mənimsədir. Hər bir class-in xüsusiyyət yığımı olur, hansı ki, **ad=mənası** kimi yazılır. Verilənlərə sürətli yetki almaq üçün sistem **/etc/login.conf** faylıni birbaşa oxumaq əvəzinə **/etc/login.conf.db** faylıni oxuyur, hansı ki, **cap\_mkdb(1)** əmri ilə yaradılır.

### **cap\_mkdb /etc/login.conf**

Ona görə də **/etc/login.conf** faylında edilən hər bir dəyişiklikdən sonra **cap\_mkdb(1)** əmrini daxil etməyi unutmayın. İstifadəçi üçün class-in əlavə edilməsini və ya dəyişdirilməsini etmək istəsəniz, **/etc/master.passwd** faylında class sütununu dəyişmək lazımdır. Bunun haqqda yuxarıda danışmışdıq. **UID=0** olan istifadəçi sistemin administratorunda(root) heç bir işləyən class olmur və ona **/etc/login.conf** faylında **root** class-ı, ya da root class-ı mövcud deyilsə, **default class** mənimsədir. İstifadəçi özü üçün şəxsi resurs quraşdırımıları ilə öz ev qovluğunda ayrıca fayl yarada bilər. Yəni öz ev qovluğunda **~/login.conf** adlı fayl yaradacaq və sintaksisi elə **/etc/login.conf**-da olduğu kimi olacaq, ancaq burada qeydiyyat ID-sinin adı "me" olacaq. Bu fayl ilə istifadəçi yalnız özünə aid edilmiş resursları kiçildə bilər, artırı yox.

**/etc/login.conf** faylında sütunların ayırcısı kimi ":"(iki nöqtə) simvolu istifadə edilir. İlk sütun gələcəkdə hansısa istifadəçiye mənimsediləcək class-in adını təyin edir.

**/etc/login.conf** faylında olan hər bir sütun aşağıdakı mənaları qəbul edə bilər:

- **bool** - əgər parametr bool-dursa, onda o, göstərilən mənaları mənimseyə bilər – **true**, ya da **false**; Sadəcə **/etc/login.conf** faylında olan sütun açıq şəkildə təyin edilməyib, demək, - **true**. **false** qeyd etmək üçün isə özümüz yazmalıyıq.
- **file** - opsiya faylin ünvanı görünüşündə məna qəbul edir;
- **program** - opsiya yerinə yetirilən fayl və ya programın ünvanı mənasını qəbul edir;
- **list** - opsiya vergül və boşluqla ayrılan siyahı mənasını qəbul edir;
- **path** - opsiya ünvan mənasını qəbul edir, hansı ki, boşluqla ayrılır.  
Tilda (~) istifadəçinin ev qovluğu mənasını verir.
- **number** - 8-lükdə, 10-luqda, 16-liqda olan rəqəmsal məna.
- **string** - sətir tipli;

- **size** - həcm. Susmaya görə baytlarla qəbul edilir. Aşağıdakı suffiksləri qəbul edə bilir ki, həcmi təyin edə bilsin:
  - b** - bayt
  - k** - kilobayt
  - m** - megabayt
  - g** - qıqqabayt
  - t** - terabayt

Həmçinin mümkündür ki, uyğun olan bir neçə suffuksin birləşməsi istifadə edilə bilsin: **1m30k**

- **time** - vaxt aralığı, susmaya görə saniyelərlə təyin edilir. Suffiks olaraq aşağıdakı mənalar mənim sədilə bilər:
  - y** - il
  - w** - həftə
  - d** - gün
  - h** - saat
  - m** - dəqiqə
  - s** - saniyə

Uyğun suffikslərin birləşməsi mümkündür: **2H30m**

- **unlimited** - Məhdudiyyət yoxdur.

### Resursların məhdudlaşdırılması

#### Cədvəl 1.1

Opsiyonun adı	Təyinat tipi	Açıqlanması
<b>coredumpsize</b>	size	Coredump faylinin həcmini məhdudlaşdırır.
<b>Cputime</b>	time	Prosesorun işləmə vaxtını məhdudlaşdırır.
<b>datasize</b>	size	Verilənlərin maksimal həcmi
<b>Filesize</b>	size	Faylin maksimal həcmi. Sayca göstərilən rəqəmdən çox fayl yaratmağa qadağası qoyur.
<b>Maxproc</b>	number	Proseslərin maksimal sayı, hansı ki, istifadəçi yarada bilər.
<b>memorylocked</b>	size	Core memory-nin maksimal həcmi, hansı ki, proses block edə bilər.
<b>memoryuse</b>	size	Proses istifadə edə biləcək memory-nin maksimal həcmi.
<b>openfiles</b>	number	Hər prosesin açı biləcəyi maksimal faylin sayı
<b>sbsize</b>	size	SocketBuffer-in maksimal icazə verilən həcmi
<b>vmemoryuse</b>	size	Hər proses üçün maksimal izin verilən virtual yaddaşın həcmi
<b>stacksize</b>	size	Stack-in maksimal həcmi

Resurslar yumşaq məhdudiyyətdə olduğu kimi sərt məhdudiyyətdə də təyin edilə bilər. Aralarında fərq ondan ibarətdir ki, sərt təyin edilmiş məhdudiyyəti istifadəçi sonradan artırı bilməz. Ancaq yumşaqları o şərtlə artırı bilər ki, yenə də sərt təyin edilən məhdudiyyətdən çox olmasın. Yumşaq və sərt məhdudiyyətlərin təyin edilməsi üçün spesifik suffix olan **-max** və **-cur** istifadə edilir. Məs: **filesize-max**

#### İstifadəçi əhatəsi:

#### Cədvəl 1.2

Opsiyanın adı	Təyinat tipi	Susmaya görə	Açıqlanması
<b>charset</b>	string		Dəyişən mühitin mənasını təyin edir <b>\$MM_CHARSET</b> . Məs: <b>KOI8-R</b>
<b>hushlogin</b>	bool	false	Izin verir ( <b>false</b> ) ki, yüklənmə vaxtı <b>/etc/motd</b> görünən, ya da qadağa qoyur( <b>true</b> ). <b>~/.hushlogin</b> faylinin ev qovluğunda olması da eyni işi görəcək.
<b>ftp-chroot</b>	bool	false	<b>chroot(2)</b> - istifadəçini FTP login vaxtı öz ev qovluğunda root kimi göstərir. Yalnız standart daemon <b>ftp(8)</b> -e mənimsədilə bilər.
<b>ignorenologin</b>	bool	false	Login edilmə nologin-lə block edilmir.
<b>label</b>	string		İstifadəçiə mənimşədiləcək MAC( <b>maclabel(7)</b> ) siyaseti
<b>lang</b>	string		Dəyişən mühiti <b>\$LANG</b> -in mənasını təyin edir. Məs: <b>AZ.UTF8</b>
<b>manpath</b>	path		Man səhifəsinin axtarış ünvanını təyin edir.
<b>nocheckmail</b>	bool	false	İstifadəçi sistemə daxil olduqda mail yesiyinin vəziyyətini göstərir.
<b>nologin</b>	file		Əgər bu fayl mövcuddursa, onda bu faylin tərkibi sistemə daxil olduqda göstərilir və sessiya bağlanır. Bu opsiyanı istifadəçi üçün nəzərdə tutulan class-də təyin etmək və onun sistemə girişini block etmək olar. Hətta onda <b>/etc/master.passwd</b> faylında doğru shell mühiti təyin edilsə də, block ediləcək.

<b>path</b>	path		Yerinə yetirilən fayl və ya programın axtarış ünvanını təyin edir.
<b>priority</b>	number		İstifadəçinin ilkin prioritetini təyin edir ( <b>nice(1)</b> ).
<b>requirehome</b>	bool	false	İstifadəçi ev qovluğu lazımdırı, əgər deyilsə, onda o sistemə giriş edə bilməyəcək.
<b>setenv</b>	list		Vergül(,) ilə ayrılan dəyişən mühitini <b>dəyişən=mənası</b> formasında təyin edir.
<b>shell</b>	prog		İstifadəsi shell mühiti <b>/etc/master.passwd</b> faylında təyin edilən shell mühitinin prioritetini üstün sayıır.
<b>term</b>	string		Terminal tipini təyin edir.
<b>timezone</b>	string		<b>\$TZ</b> dəyişəninin mənasını təyin edir. Zonalar <b>/usr/share/zoneinfo</b> ünvanında yerləşir.
<b>umask</b>	number	022	Yaradılacaq fayllar üçün hüquqları təyin edir. Yetkilər 666-dan çıxmazla, qovluq üçün isə 777-dən çıxmazla təyin edilir.
<b>welcome</b>	file	./etc/motd	Salamlaşma faylı, hansı ki, istifadəçi sistemə daxil olduqda göstərilir.

## Istifadəçinin giriş

### Cədvəl 1.3

Opsiyanın adı	Təyinat tipi	Susmaya görə	Açıqlanması
<b>copyright</b>	file		Copyright haqda informasiya saxlayan əlavə fayl
<b>host.allow</b>	list		Uzaq məşinlərin siyahısı, bu ünvanlardan gələn istifadəçilər sistemə daxil ola bilərlər.
<b>host.deny</b>	list		Uzaq məşinlərin siyahısı, bu ünvanlardan gələn istifadəçilər sistemə daxil ola bilməzlər.
<b>login_prompt</b>	string		<b>login(1)</b> müraciətinin gəlməsi nəticəsində çıxan sətir.
<b>login-backoff</b>	number		Səhv olan daxil olmalar arasında olan gecikmə vaxtı 5 saniyəyə vurulmuş formada. Giriş cəhdlərinin sayının bitməsi sonrakı parametrdə təyin edilir. Uzaq olmayan terminala mənimşədilə bilər.
<b>login-retries</b>	number	10	Daxil olmaq cəhdini uğursuz qeydə alınanadək giriş üçün icazə verilən səhv cəhdlərin sayı.
<b>password_format</b>	string		Yeni parolun şifrələnmə formatı. Mənasını ' <b>md5</b> ', ' <b>blf</b> ' və ' <b>des</b> '-də təyin edə bilərsiniz. Susmaya görə ' <b>blf</b> ' formatından istifadə edilməsi daha məqsədə uyğundur, ona görə ki, şifrələnməyə daha dayanıqlı alqoritmdir.
<b>password_prompt</b>	string		Şifrə üçün salamlaşma
<b>times.allow</b>	list		Vaxt aralığıdır, hansı ki, bu aralıqda sistemə girişə izin var.
<b>times.deny</b>	list		Vaxt aralığıdır, hansı ki, bu aralıqda sistemə giriş qadağandır.
<b>ttys.allow</b>	list		Terminal qrupu və ya siyahısı, hansı ki, bu class ilə olan istifadəyə izin var. Terminallar qrupuna <b>/etc/ttys(5)</b> faylında baxa bilərsiniz.

<b>ttys.deny</b>	list	Terminal qrupu və ya siyahısı, hansı ki, bu class ilə olan istifadəyə izin yoxdur. Terminallar qrupuna <b>/etc/ttys(5)</b> faylında baxa bilərsiniz.
<b>warnexpire</b>	time	Vaxt aralığıdır, hansı ki, bu aralıqda istifadəçinin vaxtının bitməsi haqqında ona xəbərdarlıq göndərilir.
<b>warnpassword</b>	time	Vaxt aralığıdır, hansı ki, bu aralıqda istifadəçinin şifrəsinin vaxtının bitməsi haqqında ona xəbərdarlıq göndərilir.

**/etc/hosts.allow** və **/etc/hosts.deny** fayllarında hostların ayırıcısı kimi vergül istifadə edilir.

**times.allow** və **times.deny** opsiyalarında yazılar vergül ilə ayrılır. Vaxt aralığı 24-saatlıq formatda yazılır və bir-birilə defislə ayrılır. Məsələn: **MoThSa0200-1300** Bu yazı aşağıdakı qaydada açıqlanır: İstifadəçiye 1-ci, 4-cü və 5-ci günlər gecə saat 2-dən gündüz saat 1-dək izin verilir. Əgər hər iki opsiya class-da olmazsa, yetki istənilən vaxtda olacaq. Əgər **times.allow** opsiyasında olan izin vaxtı aralığı eyni vaxtda **times.deny** faylında da qadağan edilirsə, onda üstünlük **times.deny** faylinə verilir.

**ttys.allow** və **ttys.deny** opsiyalarında vergüllə (**/dev/** prefaksi olmadan) ayrılan **tty** alətlərin yazıları və **ttygroups** (həmçinin **getttyent(3)** və **tty(5)**-ə baxın)siyahısı olur, hansı ki, bu class istifadəçisinin yetkisi olur, ya da olmur. Əgər opsiyada heç bir yazı yoxdursa, onda istifadəcidə məhdudiyyətsiz yetki olacaq.

Şifrə parametrləri necə ki, minimal uzunluq (**minpasswordlen**) və xəbərdar edici parametrlərdə, əgər istifadəçi öz şifrəsini yalnız kiçik simvollarla daxil edirsə (**minpasswordcase**), dəstəklənməyəcək. Bu məhdudiyyətlər üçün isə **pam\_passwdqc(8)** pam modulu mənimsədirilir.

Sistem istifadəçiləri üçün class-ların təyin edilməsi, seçilmiş istifadəçinin məhdudlaşdırılması üçün yaxşı üsuldur, ancaq bu imkanı tam başa düşərək və dəqiq istifadə etmək lazımdır.

İstifadəçi və qrupların idarə edilməsi üçün aşağıdakı əmrlər effektivdir

- **pw(8)** - İstifadəçi və qrupları yaratmaq, silmək, dəyişmək, göstərmək üçün;
- **adduser(8)** - Yeni istifadəçinin interaktiv əlavə edilməsi;
- **rmuser(8)** - İstifadəçinin sistemdən silinməsi;

- **id(1)** - İstifadəçi adını, UID-ni və yerləşdiyi qrupları GID-ləri ilə birləşdə göstərir;
- **finger(1)** - Sistem istifadəçisi haqqında informasiyani göstərir;
- **users(1)** - Hal-hazırkı istifadəçilərin siyahısını çap edir;
- **who(1)** – Sistemdə kimlərin olduğunu göstərir;
- **whoami(1)** - Hal-hazırkı istifadəçinin effektiv ID-sini göstərir;
- **last(1)** - İstifadəçilərin terminaldan istifadə vaxtını göstərir;
- **lastlogin(8)** - Terminalın son istifadəsi haqda məlumatı göstərir;
- **lastcomm(1)** - İstifadəçilər tərəfindən bütün son yerinə yetirilən əmrləri göstərir;
- **ac(8)** - İstifadəçinin sistemdə nə qədər vaxt olduğunu göstərir;
- **sa(8)** - İstifadəçilərə görə statistikani göstərir;
- **passwd(8)** - İstifadəçi parolunun dəyişdirilməsi;
- **chpass(1)** - İstifadəçi parolunun dəyişdirilməsi;
- **chfn(1)** - İstifadəçi verilənlərinin dəyişdirilməsi;
- **groups(1)** - Kimin hansı qrupda olduğunu göstərir;
- **chgrp(1)** - Qrupun dəyişdirilməsi;
- **chkgrp(8)** - **/etc/group** faylinin sintaksisinin düzgünlünü yoxlayır;
- **vipw(8)** - **/etc/master.passwd** faylinin **pwd\_mkdb(8)** əmrini daxil etmədən redaktə edilməsi.

Ancaq **lastcomm(1)**, **sa(8)** əmrlərinin işləməsi üçün siz istifadə olunan resursların qeydiyyatını aktivləşdirmelisiniz. Proseslər tərəfindən istifadə edilən resursların qeydiyyatı bir tərəfdən müdafiə metodikasıdır. İnzibati sistem resurslarının istifadəsini və bölməsini bu üsulla izləyə və lazımlıda onun bölgüsünü düzgün edə bilir.

Sistem resurslarının qeydiyyatının aparılmasını işə salmaq üçün siz bir neçə addım iş görməlisiniz.

1. **mkdir /var/account**
2. **touch /var/account/acct**
3. **accton /var/account/acct**
4. **echo 'accounting\_enable=" YES"' >> /etc/rc.conf**
5. **/etc/rc.d/accounting start**

Bu 5 addımı yerinə yetirməklə siz artıq **lastcomm(1)**, **sa(8)** əmrlərini istifadə edə bilərsiniz. **accton(8)** əmri istifadə edilən resursların qeydiyyatını aktivləşdirir.

Praktika/təcrübə/misal:

(Əmrlərin daha detallı istifadəsini öyrənmək üçün mütləq man-ları oxuyun.)

Resursların istifadəsi hesabatı, əmr **lastcomm(1)** və **sa(8)**:

```
root@mercuri:~ # lastcomm root
```

cron-F	root	—	0.001	secs	Wed Feb 19 09:55
cron	-F	root	---	0.001	secs Wed Feb 19 09:55
atrun	-	root	---	0.007	secs Wed Feb 19 09:55
cron	-F	root	---	0.001	secs Wed Feb 19 09:50
atrun	-	root	---	0.005	secs Wed Feb 19 09:50
sh	-	root	pts/0	0.010	secs Wed Feb 19 09:39
more	-	root	pts/0	0.013	secs Wed Feb 19 09:39
sh	-	root	---	0.002	secs Wed Feb 19 09:47
rm	-	root	---	0.000	secs Wed Feb 19 09:47
sh	-	root	---	0.002	secs Wed Feb 19 09:47
cat	-	root	---	0.000	secs Wed Feb 19 09:47
cat	-	root	---	0.001	secs Wed Feb 19 09:47
rm	-	root	---	0.000	secs Wed Feb 19 09:47
route	-	root	---	0.001	secs Wed Feb 19 09:47
hostname	-	root	---	0.000	secs Wed Feb 19 09:47
cron	-F	root	---	0.000	secs Wed Feb 19 09:45
atrun	-	root	---	0.002	secs Wed Feb 19 09:45
cron	-F	root	---	0.000	secs Wed Feb 19 09:44

```
root@mercuri:~ # sa
```

74	17.275re	0.00cp	Oavio	91733k	
14	8.635re	0.00cp	Oavio	93659k	***other
7	8.635re	0.00cp	Zavio	212192k	sh
5	0.001re	0.00cp	Oavio	0k	atrun
21	0.000re	0.00cp	Oavio	0k	mv
2	0.000re	0.00cp	Oavio	0k	zcat
9	0.002re	0.00cp	Oavio	0k	cron*
3	0.000re	0.00cp	Oavio	0k	dd
2	0.001re	0.00cp	Oavio	0k	lastcomm
3	0.000re	0.00cp	Oavio	0k	unlink
2	0.001re	0.00cp	Oavio	0k	accton
2	0.000re	0.00cp	Oavio	0k	cat
2	0.000re	0.00cp	Oavio	0k	sysctl
2	0.000re	0.00cp	Oavio	0k	rm

Yeni istifadəçinin sistemə əlavə edilməsi, [pw\(8\)](#) əmrindən istifadə edəcəyik

```
#pw useradd test -s /bin/sh -c "Test user" -m -b /home/ -e 03-06-2014 -p 02-6-2014
```

Bunu addımlarla açıqlayaq:

- s - Göstərir ki, hansı shell istifadə ediləcək. shell sütunu
- c - Yaradılan istifadəçi üçün şərh, gecos sütunu
- e - İstifadəçi adının yaşama müddəti, expire sütunu. Sütun formatı  
‘-p’ opsiyası ilə eynidir.
- p - Şifrənin yaşama müddəti, change sütunu. Tarix və vaxtın formatı aşağıdakı kimidir:  
**dd-mm-yy[yy]**, dd - gün, mm - ay, yy[yy] - il. Ya da aşağıdakı format istifadə edilir:  
**+0mhdwoy**, m - dəqiqə, h - saat, d - gün, w - həftə, o - ay, y - il.
- m - Məcbur edir ki, istifadəçinin ev qovluğu yaransın və ora **/usr/share/skel** qovluğundan standart faylları nüsxələsin.
- b - Baza qovluğudur, hansında ki, istifadəçinin ev qovluğu yaranacaq **home\_dir**.
- L - **login.conf** faylından istifadəçi üçün class-ı təyin edir, **class** Sütunu.

İşimizin nəticəsində **/etc/master.passwd** faylı aşağıdakı kimi olacaq:

```
test:*:1002:1002::1401652800:1401739200:Test user:/home/test:/bin/sh
```

Daha interaktiv əmr [adduser\(8\)](#)-dir, onun quraşdırma faylı var, hansı ki, ötürülən verilənləri orada təyin edə bilərsiniz.

Gördüyüümüz kimi, yaratdığımız istifadəçinin şifrəsi yoxdur və buna görə də sistemə daxil ola bilməyəcək. Şifrəni təyin etmək üçün isə [passwd\(8\)](#) əmrindən istifadə edirik:

```
root@mercuri:~ # passwd test
Changing local password for test
New Password:
Retype New Password:
```

İstifadəçinin sistemdən daha düzgün silinməsi üçün [rmuser\(8\)](#) əmrindən istifadə edin. Bu əmr sadəcə istifadəçinin **/etc/master.passwd** faylından silmir, həm də ona aid olan verilənləri sistemdən silir:

1. İstifadəçi [crontab\(1\)](#)-ni silir, əgər mövcuddursa.
2. İstifadəçinin [at\(1\)](#) əmri ilə yaratdığı bütün tapşırıqları silir.
3. Bu istifadəçi adından işləyən bütün proseslərə **SIGKILL** siqnalını ötürür.

4. İstifadəçini **/etc/passwd** faylından silir.
5. İstifadəçinin ev qovluğununu və ev qovluğununa aid olan bütün simvolik linkləri silir.
6. **/var/mail** qovluğundan bütün daxil olan mail-ləri silir.
7. Sahibi bu istifadəçi olan **/tmp**, **/var/tmp** və **/var/tmp/vi.recover** qovluqlarında olan bütün faylları silir.
8. Bu istifadəçini bütün olduğu qruplardan silir (**/etc/groups** faylı).
9. Bütün növbə mesajlarını silir, bütün ayrılmış yaddaşı, sahibi bu istifadəçi olan bütün semaforları silir. (Açıqlama üçün: Bütün bu interfeyslər prosessorlar arası qarşılıqlı hərəkətdir.) **rmuser(8)** utilitini **'-y'** parametri ilə istifadə edin ki, o, silinmə haqqında təsdiq gözləməsin.

**'-y'** opsiyası olmadan silmənin və **'y'** opsiyası olaraq silmənin misalları:

```
root@mercuri:~ # rmuser test
Matching password entry:

test:*:1002:1002::0:1401739200:Test user:/home/test:/bin/sh
```

```
Is this the entry you wish to remove? y
Remove user's home directory [/home/test]? y
Removing user (test): mailspool home passwd.
```

```
root@mercuri:~ # pw useradd test -s /bin/sh -c "Test user" -m -b /home/ -e 03-06-2014 -p 02-6-2014
root@mercuri:~ # rmuser -y test
Removing user (test): mailspool home passwd.
```

Artıq mövcud olan istifadəçilər haqqında məlumatə baxaq, bunun üçün id(1), finger(1) əmrlərindən istifadə edəcəyik.

```
root@mercuri:~ # id root
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
```

Gördüyüümüz kimi, utilit göstərir ki, bu istifadəçi hansı qruplarda mövcuddur. Bu əmrin həddən artıq yararlı opsiyaları mövcuddur, ona görə də man-nını oxuyun.

```
root@mercuri:~ # id -P root
root:$6$dnI.44AaLBdCyitt$p9W4qCGQZTwX1EwtoleAbHs/t7VV76WJ4tAYGF.
```

```
dIoREEK.2UvKxkhFImfP.uvg5kKNb6iMN2uZrOKLwE/Acb1:0:0::0:0:Charlie &:/root:/bin/csh
```

```
root@mercuri:~ # finger root
Login: root                               Name: Charlie Root
Directory: /root                            Shell: /bin/csh
On since Wed Feb 19 21:51 (AZT) on ttv0, idle 0:28 (messages off)
On since Wed Feb 19 21:51 (AZT) on pts/0 from 192.168.68.1
No Mail.
No Plan.
```

### Yeni qrup yaradaq

```
root@mercuri:~ # pw useradd atrium -s /bin/sh -c "Test user" -m -b /home/ -e 03-
06-2014 -p 02-6-2014
```

```
root@mercuri:~ # echo 'test:*:200:atrium'>> /etc/group
root@mercuri:~ # id atrium
uid=1002(atrium) gid=1002(atrium) groups=1002(atrium),200(test)
```

Biz yeni '**test**' adlı qrup yaratdıq və həmin anda da ona '**atrium**' adlı istifadəçini əlavə elədik.

```
root@mercuri:~ # pw groupadd list -M atrium,root
root@mercuri:~ # pw groupshow list
list:*:1003:atrium,root
```

**pw(8)** əmrinin sayesində biz '**list**' adlı qrup yaratdıq və ona **atrium** ilə **root** istifadəçisini əlavə etdik. Həmçinin yaratdığımız qrupa baxdıq.

# Yetki hüquqlarının idarə edilməsi

OS konsepsiyasının formallaşmasında yetki hüquqları çox önemli rol oynayır. FreeBSD-də yetki hüquqları Windows(nəsil daşınma hüquqları var) OS-dan fərqli olaraq çox sadədir. Düzdür, Windows-da olan bu imkan bizim işimizi rahatlaşdırır, ancaq həddən artıq mürəkkəb yetki hüquqlarının verilməsi çox böyük səhvə gətirib çıxara bilər.

FreeBSD əməliyyat sistemində cəmi 3 standart flag mövcuddur(həmçinin spesifik flaglar da mövcuddur [cədvəl 1.5]). Onlar bu və ya digər yetki hüquqlarını təyin edir:

- u** - oxumaq
- w** - yazmaq
- x** - yerinə yetirmək

Bu flaglar aşağıdakı 3 kateqoriyaya mənimsədirilir:

- u** - İstifadəçi (owner)
- g** - Qrup (group)
- o** - Digər (other)

Sahibi asılı olmadan istifadəçi təyin edir, hansı ki, bu və ya digər resursa malikdir (fayl, alət, qovluq). O, fərqli yetki flag-larını həm təyin edə, həm də yığışdırıa bilər. Resursa birgə yetki hüququ olan istifadəçilərin siyahısını **qrup** təyin edir.

Digər - Bu, o istifadəçilərdir ki, heç bir qrupa daxil deyillər, həmçinin fayl və ya qovluğa heç bir sahiblikləri yoxdur. Digərləri həmişə istənilməyən istifadəçilər olub və əksər hallarda minimal yetkiyə sahib olurlar və ya heç yetkiyə sahib olmurlar.

Hansısa resursun təyin olunmuş yetki hüquqlarına, həmçinin sahibi və qrupuna baxmaq üçün **ls(1)** əmri **-l** opsiyası ilə istifadə edilir:

```
root@mercuri:~ # ls -l /root/tests  
-rwxr-x--- 1 atrium wheel 0 Feb 19 22:58 /root/tests
```

Gördüyüümüz kimi, **/root/test** faylinə **atrium** istifadəcisinin oxuma/yazma/yerinə yetirmə yetkiləri, **wheel** qrupunun oxuma/yerinə yetirmə və digərlərinin isə heç bir yetkisi yoxdur.

Yetki hüquqlarının bütün xirdalıqlarını anlamaq üçün isə vacibdir ki, yoxlanış qaydasını açıqcasıınız

1. İlk olaraq yoxlanılır ki, istifadəçi faylin sahibidir, ya yox. Əgər o, sahibidirsə, onun yetkisi təyin edilir və ardınca yetki hüquqları qəbul edilmir, ona görə ki, hətta istifadəçi yetkisi sahibindən az olan qrupda olsa belə, qrupun yetkisi yoxlanılmayacaq. Əgər istifadəçi resursun sahibidirsə və onun üçün hansısa bir yetki təyin edilməyib, onda yetki qadağan edilir.

Məsələn:

**atrium** istifadəcisi təyin edilmiş hansısa bir faylin sahibidir və **wheel** qrupundadır. Fayla olan yetki **rwx-r-x---** formasında təyin edilmişdir. Bu halda faylin sahibi yaza/oxuya/yerinə yetirə, qrupun üzvləri oxuya/yerinə yetirə bilər.

```
root@mercuri:~ # id atrium  
uid=1002(atrium) gid=1002(atrium) groups=1002(atrium),200(test),1003(list)
```

```
root@mercuri:~ # ls -l /root/tests  
-rwxr-x--- 1 atrium wheel 0 Feb 19 22:58 /root/tests
```

```
root@mercuri:~ # su - atrium  
$ echo "test message" >> /root/tests
```

Gördüyüümüz kimi, fayla yazmaq uğurla yerinə yetirildi.

2. Əgər istifadəçi faylin sahibi deyilsə, yoxlanılır ki, o, resursa yetkisi olan qrupa daxildirsə, ona qrupda olan yetkiləri mənimsədir. Əgər istifadəçi bu qrupdadırsa və qrupun yetkisi yoxdursa, eyni ilə istifadəçi üçün də yetki qadağan olacaq.
3. Əgər istifadəçi faylin sahibi deyilsə və resursa yetkisi olan qrupda deyilsə, onda digərlərinə aid olan yetki mənimsədir.

4. Əsasən, root istifadəcisinə aid olan yetkilər yoxlanılır. Əgər sistem görsə ki, istifadəçi də UID=0-dır, hansı ki, root deməkdir, onda login\_name-dən asılı olmayıaraq, sistem heç bir yoxlanış etməyəcək, çünki superuserin hər yerə tam yetkisi olur.

#### Yetki hüquqlarının təqdimat növləri

Yetki hüquqlarını rəqəmsal görünüşdə yazdığımız kimi, eynilə də simvol görünüşündə yazmaq olar.

- **Rəqəmsal variant**

Yetki hüquqları 3 kateqoriyada təyin edildiyinə görə və 3 tip yetki hüququ olduğuna görə cəmi 9 flag təyin edilir:

(111 - 111 - 111) – Bu, 777 yetki hüququna uyğun gəlir.

Yetki hüquqları görünüşü 2-li say sistemində göstərilir. Hər bir kateqoriyaya 3 flag təyin eləmək olar - oxumaq(r), yazmaq(w), yerinə yetirmək(x) və bu (111) deməkdir. 1 - bu və ya digər flag-in olması deməkdir, 0 - uyğun olaraq, bu və ya digər flag-in olmaması deməkdir.

İkili say sisteminin 10-luğa çevrilməsi üçün aşağıdakı cədvəldən istifadə edilir. O, sağdan sola oxunur:

**128 64 32 16 8 4 2 1**  
1 1 1 1 1 1 1 1

Öz yetkilərimizi 10-luq sistemə necə köçürməliyik? Gəlin məşq edək. test.txt faylına aşağıdakı yetkiləri təyin edək:

	<b>u(user)</b>	<b>g(group)</b>	<b>o(other)</b>
test.txt	111	101	100
	rwx	r-x	r--

İndi alınan nəticələri rəqəmlərə çevirək:

- **u(user)** - 111 - öz cədvəlimizə sağdan sola doğru baxaq. Bu ardıcılıqla rəqəmlər - 124 olacaq. Alınan rəqəmləri isə hesablayırıq və alırıq -  $1+2+4=7$
- **g(group)** - 101 - eyni işi görürük -  $1+4=5$
- **o(other)** - 100 - eyni işi görürük -  $4 = 4$

Nəticədə bizim yetki hüquqlarımız fayl üzərində **754** kimi olacaq və bunu **chmod(1)** əmri ilə təyin etmək olar.

Rəqəmsal mənənin ikili saya çevrilməsi üçün 2-yə bölmək lazımdır, bölmə vaxtı qalıq **1**-ə uyğundur, əgər qalıq **0** deyilsə. Sağdan sola yazılır.

Məsələn:

```
root@mercuri:~ # touch test.txt
root@mercuri:~ # chmod 754 test.txt
root@mercuri:~ # ls -l test.txt
-rw-r--r-- 1 root wheel 0 Feb 20 12:23 test.txt
```

Həmçinin yadda qalmanın rahat olması üçün rəqəmsal görünüşdə yetki hüquqlarının cədvəlini yaradaq:

#### Cədvəl 1.4

Kateqoriyası	Rəqəmsal	İkili	Simvol görünüşlü	Mənası
Sahibi	0400	100-000-000	r--- ---	Sahibi üçün oxuma yetkisi
	0200	010-000-000	-w- --- ---	Sahibi üçün yazma haqqı
	0100	001-000-000	--- -r- ---	Fayllar üçün yerinə yetirilmə, ancaq kataloqlar üçün axtarışa izin verir.
Qrup	0040	000-100-000	--- -r- ---	Qrup üçün oxuma
	0020	000-010-000	--- -w- ---	Qrup üçün yazma
	0010	000-001-000	--- --x ---	Fayllar üçün yerinə yetirilmə, qovluqlar üçün axtarış etmə imkanı
Digerleri	0004	000-000-100	--- --- r--	Digər istifadəçilər üçün oxuma yetkisi
	0002	000-000-010	--- --- -w-	Digər istifadəçilər üçün yazma
	0001	000-000-001	--- --- --x	Digər istifadəçilər üçün faylin yerinə yetirilməsi və qovluqlarda axtarış etmək imkanı

Bu cədvəldən göründüyü kimi, resursa müxtəlif kateqoriyalı yetki hüququ təyin etmək üçün sadəcə verilmiş qovluğun yetki hüququnu öz aralarında təyin etmək lazımdır. Məsələn, aşağıdakı yetkiləri fayla təyin etmək lazımdır:

1. Sahib üçün - oxumaq, yazmaq, yerinə yetirmək;
2. Qrup üçün - oxumaq, yazmaq;
3. Digərləri üçün – oxumaq.

Sahibi: 0400 + 0200 + 0100 = 0700

Qrup: 0040 + 0020 = 0060

Digərləri: 0004 = 0004

Nəticədə, fayl üçün yetki hüququ - **764** alırıq, bu yetkini **chmod(1)** əmri ilə təyin etmək olar.

- **Simvollu üsul**

Yetki hüquqlarının simvol ilə təqdim olunma üsulunu yadda saxlamaq həddən artıq sadədir. Bu üsulla yetki hüququnun təyinatı 4 kateqoriyaya bölünür:

- a** - Hər şey (all)(ugo)
- u** - Sahibi (owner)
- g** - Qrup (group)
- o** - Digər (other)

Bu 4 kateqoriya üçün aşağıdakı yetki hüquqları təyin edilir:

- r** - Oxuma yetkisi[Cədvəl 1.4-ə baxın]
- s** - istifadəçi üçün(owner) - setuid bit, g(group) üçün - setgid bit[Cədvəl 1.5-ə baxın]
- t** - sticky bit[Cədvəl 1.5-ə baxın]
- w** - Yazma yetkisi[Cədvəl 1.4-ə baxın]
- x** - Yerinə yetirilmə yetkisi[Cədvəl 1.4-ə baxın]

Yuxarıda göstərdiyimiz yetkiləri aşağıdakı simvollarla idarə etmək mümkündür:

- +** - Fayl və ya qovluq üçün göstərilmiş yetkiləri əlavə edir.
- - Göstərilən yetkiləri fayl və ya qovluq üçün silir.
- =** - Göstərilən yetki hüquqlarını təyin edir, həmçinin öncə olan yetkiləri bu qovluq üçün silir.

Simvol variantında yetki hüququnun təyinatında kateqoriyalar vergül simvolu ilə ayrılır (,). Misal olaraq, faylin bu yetki hüququ var (rw-r-rw-) və biz ona bu yetki (**rwx ---w-**) hüququnu veririk:

```
root@mercuri:~ # ls -l file.txt  
-rwx-w--- 1 atrium wheel 0 Feb 20 14:45 file.txt
```

```
root@mercuri:~ # chmod u+x,g=x,o-r file.txt  
root@mercuri:~ # ls -l file.txt  
-rwx--x--- 1 atrium wheel 0 Feb 20 14:45 file.txt
```

Ya da başqa yolla etmək olar.

```
root@mercuri:~ # ls -l file.txt  
-rwx--x--- 1 atrium wheel 0 Feb 20 14:45 file.txt
```

```
root@mercuri:~ # chmod u+x,g=r+x,o-r file.txt
```

```
root@mercuri:~ # ls -l file.txt  
-rwx--x--- 1 atrium wheel 0 Feb 20 14:45 file.txt
```

Standart yetki hüquqlarından başqa oxumaq, yazmaq, yerinə yetirmək üçün spesifik yetki hüquqları tələb edilir. Onları həm simvol, həm də rəqəmsal variantda göstərmək olar:

## Cədvəl 1.5

Rəqəmsal	Simvol tipli	Mənəsi
4000	--s --- ---	Setuid bit. Faylin bu bit ilə yerinə yetirilməsində effektiv uid istifadəçinin uid-nə mənimşədir. Digər sözlə, proses faylin sahibinin adından işə düşür (məsələn, <b>passwd(1)</b> programı). Kataloq üçün isə bu bit təyin edir ki, faylin sahibi onu yaradan istifadəçi yox, kataloqun sahibi olacaq. Əgər, doğrudan da, kataloq yerləşən fayl sistem <b>suiddir</b> opsiyasını dəstəkləyirsə, <b>mount(8)</b> -ə baxın.
2000	--- --s ---	<b>setgid bit</b> . Bu bit ilə faylin yerinə yetirilməsində effektiv gid qrupun gid-nə mənimşədir.
1000	--- --- --t	<b>sticky bit</b> . Bu bit kataloqlar üçün effektiv mənimşədir. Əgər o, kataloqa təyin edilibsə, onda olan konkret faylı kataloqda olan yetkilərdən asılı olmayıaraq, yalnız onun sahibi sile bilər. Qruplu yetkidə çox effektividir.

## Vacibdir!

Öz işindən asılı olaraq, kataloqa təyin edilən hüquqlar fayla təyin edilmiş hüquqlardan fərqlənir:

Qrup, sahib və ya digərləri üçün kataloqa təyin edilmiş yazma hüququ izin verir ki, onda olan sahibdən və təyin edilmiş yetkisindən asılı olmayıaraq, orada istənilən faylı silə bilsin. Ona görə də kataloqa təyin edilən yazma yetkisindən ehtiyatlı olun. Bu problemin həlli kimi spesifik flag olan **skick** bit-dən istifadə etmək mümkündür - Stick bit(1000), cədvəl 1.5-dən baxa bilərsiniz.

1. Kataloqa təyin edilmiş yerinə yetirilmə yetkisi izin verir ki, orada **ls(1)** əmrini '**-I**'açarı ilə, **cd(1)** əmrini istifadə edə biləsiniz, ona görə ki, faylların meta verilənlərinə müraciət gedir.
2. Kataloq üçün təyin edilmiş oxuma yetkisi izin verir ki, faylların siyahısını onlar haqda ətraflı məlumat əldə etmədən öyrənmək olsun.
3. Kataloq üçün oxuma və yerinə yetirilmə yetkisi isə bir-birindən asılı olmadan işləyir, ona görə ki, hər iki yetki hüququnun olması vacib deyil. Bu yetki hüquqlarının birləşməsi nəticəsində çox maraqlı nəticə əldə etmək olar.
  - İcra edilmə təyin edilmiş qovluq, oxuma yetkisi olmadan "gizli" qovluq yaratma yetkisi verir. Bu nədir? Bu, kataloqda olan faylların siyahısını əldə etmək mümkün olmur, ancaq imkan var ki, onların adları öncədən bəlliidirsə, birbaşa müraciət etmək olsun. Misal üçün, tünd kataloq yaradıq və çalışaq ki, faylları siyahılayaq, sonra isə içində olan bir faylı oxumağa çalışaq.

```
fdesktop1# ls -l ./test
total 2
-rw-r--rw- 1 atrium wheel 8 Jun 18 00:51 file.txt
fdesktop1# ls -l
total 31
drwx--x--- 2 root   wheel   512 Jun 18 00:51 test

$ cat /root/test/file.txt
My file
$ ls /root/test
ls: test: Permission denied
```

Bu və ya digər resursa olan yetkini, həmin resursun sahibi, ya da super istifadəçi (**UID=0**) dəyişə bilər.

### Yetki hüquqlarının öncədən təyin edilməsi

Çox effektiv bir əmr var **umask(2)**, bu fayl sistemdə yeni yaradılacaq fayllara maskanı təyin edir. Bu əmri ancaq FreeBSD OS-da yox, həmçinin elə digər program təminatlarında(SAMBA) da görmək olar. Program təminatlarında o, həm də kataloqlara təyin edilə bilər, ancaq kataloqlarda olan yetkilərin oxunmasında və faylların oxunmasında fərqli üsullar mövcuddur, hansı ki, bu əmri təyin edir. Əmrin formatı çox sadədir: **3** ardıcıl rəqəm təyin edilir, əgər kataloqa təyin edilirsə **3** ardıcıl **7** rəqəmindən çıxılır, əgər fayllar üçündürsə, onda **6** rəqəmindən çıxılır. Misal üçün, FreeBSD OS üçün **240** maskası təyin edirsiniz. Bu yeni təyin edilən fayllara **426** yekisi verir, hansı ki, simvol görünüşündə **r---w-rw-** belə olacaq.

```
$ umask 240
$ touch file.test
$ ls -l
total 0
-r---w-rw- 1 atrium atrium 0 Feb 20 18:42 file.test
```

Yetki hüquqları işləmə və sahib, qrupların dəyişdirilməsində effektiv olan əmrlər aşağıdakılardır:

- **chmod(1)** - Bu və ya digər resurs üçün yetki hüququnun dəyişdirilməsi.
- **chown(8)** - Qrup və resurs sahibinin dəyişdirilməsi.
- **chgrp(1)** - Resursun qrupunun dəyişdirilməsi.
- **ls(1)** - Kataloğun oxunması və çoxlu fərqli detallı informasiyanın əldə edilməsi.
- **stat(1)** - Fayl haqqında statistik məlumatın əldə edilməsi. Çox effektiv əmdir.
- **umask(2)** - İstifadəçi üçün yaradılacaq fayllara maskanı təyin edir.

### Təcrübə

(Əmrlər haqqında daha detallı məlumat əldə etmək üçün mütləq man-larını oxuyun)

Fayl yaradaq və sahibinə oxuma/yazma yekisi, qrupu üçün oxuma və digərləri üçün isə heç bir yetki verməyək.

Rəqəmsal variantda:

```
atrium@mercuri:~ % touch file
atrium@mercuri:~ % chmod 640 ./file
atrium@mercuri:~ % ls -l
total 0
-rw-r----- 1 atrium atrium 0 Feb 21 08:51 file
```

Simvol variantda:

```
atrium@mercuri:~ % touch test.file
atrium@mercuri:~ % chmod u=rw,g=r,o-rwx test.file
atrium@mercuri:~ % ls -l
total 0
-rw-r----- 1 atrium  atrium  0 Feb 21 08:53 test.file
```

Fayl və kataloq üçün sahibi və qrupunu dəyişək.

Öncədən **atrium** istifadəçi adını **cavid** və **wheel** qrupuna əlavə edin.

```
atrium@mercuri:~ % ls -l
total 0
-rw-r----- 1 atrium  atrium  0 Feb 21 08:53 test.file
```

```
atrium@mercuri:~% chgrp cavid test.file
atrium@mercuri:~ % ls -l
total 0
-rw-r----- 1 atrium  cavid   0 Feb 21 08:53 test.file
```

```
atrium@mercuri:~ % ls -l
total 0
-rw-r----- 1 atrium  cavid   0 Feb 21 08:53 test.file
```

```
atrium@mercuri:~ % /usr/sbin/chown atrium:0 test.file
atrium@mercuri:~ % ls -l
total 0
-rw-r----- 1 atrium  wheel   0 Feb 21 08:53 test.file
```

```
root@mercuri:~ # ls -l /home/atrium/test.file
-rw-r----- 1 atrium  wheel   0 Feb 21 08:53 /home/atrium/test.file
root@mercuri:~ # chown root:0 /home/atrium/test.file
root@mercuri:~ # ls -l /home/atrium/test.file
-rw-r----- 1 root    wheel   0 Feb 21 08:53 /home/atrium/test.file
```

Fayl və ya kataloq haqqında informasiyaya baxaq.

```
atrium@mercuri:~ % stat test.file
94 2651690 -rw-r----- 1 root wheel 0 0 "Feb 21 08:53:09 2014" "Feb 21 08:53:09
2014" "Feb 21 09:01:26 2014" "Feb 21 08:53:09 2014" 32768 0 0 test.file

atrium@mercuri:~ % ls -l test.file
-rw-r----- 1 root wheel 0 Feb 21 08:53 test.file

atrium@mercuri:~ % ls -l | grep '^d'
drwxr-xr-x 2 atrium atrium 512 Feb 21 09:05 rt

atrium@mercuri:~ % stat rt
94 2651689 drwxr-xr-x 2 atrium atrium 5316670 512 "Feb 21 09:05:49 2014" "Feb
21 09:05:49 2014" "Feb 21 09:05:49 2014" "Feb 21 09:05:49 2014" 32768 8 0 rt
```

# Access Control List

**ACL (access control list)** – yetkilərin idarə edilməsində siyahı. Müxtəlif resurslara yetki hüquqlarını təyin etmək üçün imkanları genişləndirir, hansı ki, sistemdə olan yetki hüquqlarının təyinatını dəfələrlə dinamik edir. ACL imkan yaradır ki, yetki hüquqlarını bir qrup və ya istifadəçidən çox ünvana təyin edə bilək. Həmçinin maksimal yetki maskaları da təyin etmək imkanı var, hansı ki, istifadəçi, qrup və digərləri ala bilərlər. Öz növbəsində yetkilərin təyin edilməsində bu dinamiklik sistemin təhlükəsizliyini artırır, ona görə ki, bu və ya digər resursa seçimli yetki təyin etmək olur.

ACL-də həmin 3 kateqoriya istifadə edilir, hansı ki, adı yetki hüquqlarının təyin edilməsində olduğu kimi, ACL-ə də təyin etmək olur:

- **u(user);**
- **g(group);**
- **o(other);**

Və bunlar üçün simvol variantında standart yetki flag-ları təyin edilir.

[[Yetki hüquqları başlığına baxın.](#)]

- **r(oxuma);**
- **w(yazma);**
- **x(yerinə yetirilmə);**

Misal olaraq, server OS FreeBSD olanda ACL o halda həddən artıq tez-tez istifadə edilir ki, o, Windows maşına SAMBA ilə integrasiya edilmişdir. Windows sistemdə siz maşına olan yetki hüquqlarını işarələrlə təyin edə bilərsiniz.

ACL-lerin işlemesine başlamaq için kernel-i aşağıdaki opsiya ile kompilyasiya etmek lazımdır:

```
options UFS_ACL
```

Bu opsiya susmaya göre sistemin öz GENERIC kernel-ində olur. Önemli opsiyanı kernel-ə əlavə edib yenidən kompilyasiya etdiqdən sonra lazımdır ki, onun hansı fayl sistemdə işləməsini təyin edəsiniz. ACL-in fayl sistemdə aktivləşməsinin 2 üsulu var:

1. **/etc/fstab** faylında lazımi fayl sistemə flag-ı əlavə etmək **mount(8)**. Nəticədə, ACL elə OS-un yüklənməsi və mount edilməsi prosedurunda aktivləşir:

Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/da0p2	/	ufs	rw,acrls	1	
/dev/da0p3	none	swap	sw	0	0

Misaldan göründüyü kimi, acls flag-ı əsas özəyə / təyin edilmişdir. Sistem yenidən yüklənməsindən sonra bu fayl sistemdə ACL-i istifadə etmək olar. ACL-in hansı fayl sistemdə olmasını yoxlamaq üçün **mount(8)** əmrinin heç bir parametrsiz daxil edilməsi kifayətdir:

```
root@mercuri:~ # mount
/dev/da0p2 on / (ufs, local, journaled soft-updates, acls)
devfs on /dev (devfs, local, multilabel)
```

- Əgər istifadəçinin lazımi yetki səviyyəsi varsa, onda fayl sistemin yenidən acls flag-ı olmadan mount edilməsi mümkündür - **umount(8)**
- Ola bilər ki, **/etc/fstab** faylında acls flag-ını silib sistemin yenidənyüklənməsini edəsiniz və sonra bu ciddi təhlükəsizlik problemlərinə gətirib çıxara bilər.

2. İkinci üsul - bu flag-ın fayl sistemin başlığında işə düşməsidir. Bu üsul birinciye baxanda aşağıdakı səbəblərə görə daha istifadə ediləndir:
  - Həmişəlik flag fayl sistemin yenidən mount edilməsi ilə dəyişdirilə bilər. Bu, həm də o deməkdir ki, siz istifadə edilən kök fayl sistemin üzərində olan acls flag-ını sonra söndürə bilməzsınız.
  - Fayl sistemin superblockunda flag-ın təyin edilməsi acls flag-ının həmişəlik mountuna gətirib çıxarıır (hətta **/etc/fstab** faylında olmasa belə).

İkinci üsulun istifadə edilməsi üçün size **tunefs(8)** əmri kömək olacaq. Əgər istəsəniz ki, ACL-i tunefs əmri ilə işə salasınız, size aşağıdakılardan lazım olacaq:

1. ACL-i aktivləşdirmək istədiyiniz fayl sistemi **umount(8)** etmək lazımdır.
2. **tunefs(8)** əmrini '**-a enable**' opsiyası ilə yerine yetirmək lazımdır.
3. Yenidən fayl sistemi **mount(8)** əmri ilə mount etmək lazımdır.

**/etc/fstab** faylinin tərkibinə baxaq, o, bizə gələcəkdə lazım olacaq və fayl sistemde aydınlaşdırma gətirəcəyik. Bu bizə ACL dəsteklənməsinin işə salınması üçün lazımdır:

#	Device	Mountpoint	FStype	Options	Dump	Pass#
	/dev/da0s1b	none	swap	sw	0	0
	/dev/da0s1a	/	ufs	rw	1	1
	/dev/da0s1e	/tmp	ufs	rw	2	2
	/dev/da0s1f	/usr	ufs	rw	2	2
	/dev/da0s1d	/var	ufs	rw	2	2
	/dev/dal	/home	ufs	rw	2	2
	/dev/acd0	/cdrom	cd9660	ro,noauto	0	0

**/home** fayl sistemi üçün ACL-in dəsteklənməsini işə salaq. Əgər sizin ACL təyin etmək istədiyiniz fayl sistem istifadə ediləndirsə, onda onu umount edə bilməyəcəksiniz və tek istifadəçili rejimə keçməli olacaqsınız (**boot -s**, ya da yüklenmə menyusunda 3-cü bölmə). (Məhz bu testin edilməsi üçün FreeBSD8.4-də /home slice-ni tam ayrı disklabel-də mount etmişik ki, rahat umount və mount etmək olsun. Həmçinin /home slice-ni tamam ayrı /dev/dal diskinə mount etmişik):

```
root@:~ # umount /home  
root@:~ # tunefs -a enable /dev/dal  
tunefs: POSIX.le ACLs set
```

```
root@:~ # mount -o rw /dev/dal /home  
root@:~ # mount  
/dev/da0s1a on / (ufs, local)  
devfs on /dev (devfs, local, multilabel)  
/dev/da0s1e on /tmp (ufs, local, soft-updates)  
/dev/da0s1f on /usr (ufs, local, soft-updates)  
/dev/da0s1d on /var (ufs, local, soft-updates)  
/dev/dal on /home (ufs, local, acls)
```

Fayl sisteminizə ACL-in mənimsədilməsindən sonra sizə hansı fayllara ACL tətbiq edilməsini təyin etmək lazım olacaq. Bunu **ls(1)** əmri ilə edə bilərsiniz.

```
$ ls -l  
-rw-r--r-- 1 root  wheel 374 Jun 2 02:14 login  
-rw-r--r--+ 1 root  wheel   0 Jul 8 00:07 login.conf
```

Misalda gördüğümüz kimi, ACL mənimsədilən faylda müsbət işarəsi yetki hüquqlarının sonunda olur.

#### ACL ilə işləməyimizdə aşağıdakı əmrlər kömək edəcək

- **setfac1[1]** - ACL-in mənimsədilməsi, silinməsi;
- **getfac1[1]** – Təyin edilmiş ACL-lər haqqında informasiyanın əldə edilməsi;
- **ls[1]** - Faylların siyahısı və onlar haqqında məlumatın əldə edilməsi.

Əmrləri istifadə etməzdən önce istifadə edəcəyimiz ən lazımlı opsiyaları və onların istifadə ardıcılılığı qaydasını açıqlayaq.

#### **setfac1[1]** üçün:

- b - Bütün təyin edilmiş ACL-ləri silir.
- m <entries> - Mövcud olan ACL yazılarına əlavə və modifikasiya edir.
- M <file> - Seçilən faylin mövcud olan ACL-lərinə əlavə və modifikasiya edə bilir.
- n - ACL-in təyinatında maksimal yetkisi olan maskaları dəyişmir.
- x <entries> - Konkret ACL yazısını silir.
- X <file> - Göstərilən fayldan götürülen ACL yazılarını silir.

ACL təyinatının qaydası:

1. Maksimal yetkisi olan maskanı təyin edirik. Bu, istifadəçi və ya qrupun resursa olan maksimal hüquqlarında təyin edilir.
2. Maksimal yetkisi olan maskanı silməklə lazım olan ACL-ləri təyin və modifikasiya edirik.
3. Bütün, ya da lazım olan ACL yazılarını silirik.

#### **Önəmlidir!**

ACL heç bir halda sahibinin yetki hüquqlarına təsir etmir. Maksimal maska hüquqlarının dəyişdirilməsi susmaya görə olan qrupa təsir edir, ona görə ki, maksimal yetkisi olan maska hüquqları, həm də susmaya görə olan qrupa təyin edilir. ACL ilə **o(other)** yetki hüququ olan kateqoriyanın dəyişdirilməsi susmaya görə olan **o(other)** kateqoriyaya da təsir edir.

ACL-in əksər qruplarda təyinatında (**u,g,o**) ayırıcı kimi vergüldən(,) istifadə edilir.

ACL-i fayla təyin edək və bizim ACL-in işləməsini test edək

```
root@:/home # ls -l test.file
-rw-r--r-- 1 root  wheel  0 Feb 23 14:35 test.file

root@:/home # setfacl -m m::rw test.file
root@:/home # setfacl -n -m u:atrium:r,u:tests:rw test.file
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:atrium:r--
user:tests:rw-
group::r--
mask::rw-
other::r--
```

Sonra atrium istifadəçisi ilə daxil oluruq və təyin edilmiş hüquqları yoxlayırıq:

```
% echo "test message" >> /home/test.file
/home/test.file: Permission denied.
```

İndi isə **atrium** istifadəçisi üçün yetkiləri dəyişək və ona **rw** yetkisi verək. Sonra yenidən həmin fayla yazmağa cəhd edək.

```
root@:/home # setfacl -n -m u:atrium:rw test.file
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:atrium:rw-
user:atrium:rw-
user:tests:rw-
group::r--
mask::rw-
other::r--
```

Yenidən atrium istifadəçisi ilə daxil olub test.file-a nə isə yazmağa cəhd edək.

```
root@:/home # su - atrium
% echo "test message" >> /home/test.file
% cat /home/test.file
test message
```

ACL-in təyin edilməsində opsiyaların ardıcılılığı qaydası çox önemlidir. **-n** opsiyası **-m**, ya da **-x** opsiyasından öndə gəlməlidir.

### Önəmlidir!

Unutmayın ki, mövcud olan ACL-lərin təyinatı, modifikasiyası, ya da silinməsində mütləq **-n** opsiyası istifadə edilməlidir. Əgər bu hallarda **-n** opsiyası istifadə edilməzsə, yetki hüququndan asılı olaraq, maksimal maska yetkisi ya azalır, ya da artmağa başlayır.

### tests istifadəçisi üçün təyin edilmiş ACL-i silek

```
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:atrium:rw-
user:tests:rw-
group::r--
mask::rw-
other::r--


root@:/home # setfacl -n -x u:tests:rw test.file
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:atrium:rw-
group::r--
mask::rw-
other::r--
```

Eyni işi edək, ancaq bu dəfə **-n** opsiyasını istifadə etməyək. Biz görəcəyik ki, maksimal maska hüququ necə dəyişir.

```
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:atrium:rw-
user:tests:rw-
group::r--
mask::r--
other::r--
```

```
root@:/home # setfacl -x u:tests:rw test.file
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:atrium:rw-
group::r--
mask::rw-
other::r--
```

Gördüyüümüz kimi, yetki maskası read-dən read/write-a keçid etdi. Buna görə də diqqətli olmaq lazımdır.

Bir əmr ilə ACL-in 2 və daha çox istifadəçiye mənimşədilməsi

```
root@:/home # setfacl -m m::r test.file
root@:/home # setfacl -n -m u:atrium:rw,u:root:r,u:tests:rw test.file
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:root:r--
```

```
user:atrium:r--          # effective: r--
user:tests:r--           # effective: r--
group::r--
mask::r--
other::r--
```

Maksimal maska hüquqlarının necə təsir etdiyinə baxaq

ACL yaradaq və ona maksimal yetki olan **r** (oxuma), **tests** və **atrium** istifadəçiləri üçün isə **rw** (oxuma,yazma) yetkilərini təyin edək və **atrium** istifadəçisi adından fayla yazmağa çalışaq.

Hüquqları təyin edək.

```
root@:/home # setfacl -m m::r test.file
root@:/home # setfacl -n -m u:atrium:rw,u:tests:rw test.file
root@:/home # getfacl test.file
# file: test.file
# owner: root
# group: wheel
user::rw-
user:root:r--
user:atrium:r--          # effective: r--
user:tests:r--           # effective: r--
group::r--
mask::r--
other::r--
```

**atrium** istifadəçisi adından fayla yazmağa çalışaq.

```
root@:/home # su - atrium
% echo "test string" >> /home/test.file
/home/test.file: Permission denied.
```

Gördüyüümüz kimi, **atrium** istifadəçisinə **rw** yetkisi olsa belə, hüquqlar yoxdur. İstifadəçi üçün effektiv yetki hüquqları maksimal maska yetkilərini təyin elədi və effektiv yetki hüquqları sətir şəklində istifadəçinin qarşısında göstərilir.

# Spesifik flag-lar

OS FreeBSD üzerinde spesifik flag-ları təyin etmək olar, hansı ki, faylin idarə edilməsi üçün xüsusi məhdudiyyətlər təyin edir. Məhdudiyyətlər flag-larla təyin edilir, hansı ki, faylda yerləşdirilir. Flag-ların təyin edilməsi fayl sistemin təhlükəsizliyini artırır, əsasən də təhlükəsizlik səviyyəsi **security(7)**-də. Ona görə ki, hətta səviyyə(1,2)-də sistemin superuser olan root istifadəcisinə belə flag-ların götürülməsinə qadağa qoyulur.

Təyin edilmiş flag-lara **ls(1)** əmri və **-ol** opsiyaları ilə baxmaq olar. Misal üçün, kataloqunu list edək, hansı ki, orada **schg** flag-ı təyin edilmişdir:

```
root@:/home # ls -ol test.file
-rw-r--r--+ 1 root  wheel schg 13 Feb 23 14:44 test.file
```

Misaldan gördüyüümüz kimi, 5-ci sütunda schg flag-ı təyin edilmişdir.

Gəlin istifadə edəcəyimiz flag-ları açıqlayaq və onların mənalarını təyin edək:

- **arch, archived** - Arxiv faylı.
- **Opaque** - Kataloq üçün görünməz olmayan flag.
- **Nodump** - Bu flag rezerv nüsxə çıxarılmasının qarşısını alır.
- **sappnd, sappend** - Sistem flag-ı, yalnız əlavə etmə imkanı təyin edir.  
Yalnız root tərəfindən təyin edilə bilər, yəni UID=0.
- **schg, schange, simmutable** - Sistem flag-ı, dəyişikliyə qadağa təyin edir. Ancaq root istifadəçi tərəfindən təyin edilə bilər, yəni UID=0.
- **sunlnk, sunlink** - Sistem flag-ı, silməyə qadağa təyin edir. Ancaq root istifadəçi tərəfindən

təyin edilə bilər, yəni UID=0.

- **uappnd, uappend** - İstifadəçi flag-ı, yalnız əlavə etmək imkanı təyin edir. Yalnız faylin sahibi və ya root istifadəçisi tərəfindən təyin edilə bilər.
- **uchg, uchange, uimmutable** - İstifadəçi flag-ı, dəyişikliyə qadağa təyin edir. Yalnız faylin sahibi və ya root istifadəçisi tərəfindən təyin edilə bilər.
- **uunlnk, uunlink** - İstifadəçi flag-ı, silməyə qadağa təyin edir. Yalnız faylin sahibi və ya root istifadəçisi tərəfindən təyin edilə bilər.

Flag-ların təyin edilməsi və götürülməsi üçün **chflags(1)** əmrindən istifadə edilir. Flag-ın götürülməsi üçün onun önünə **no** əlavə edilir. Misal üçün, bize lazımdır ki, faylda dəyişiklik flag-ını silək **schg**. Bunun üçün **chflags(1)** əmri ilə birlikdə **noschg** flag-ını istifadə etmək lazımdır.

```
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  schg 13 Feb 23 14:44 test.file
```

```
root@:/home # chflags noschg test.file  
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  - 13 Feb 23 14:44 test.file
```

Əgər bir fayl üçün bir neçə flag-ı təyin etmək lazım olarsa, onda ayırcı kimi vergül(,) istifadə edilir.

**sunlnk** flag-ının böyük olmayan xüsusiyyəti var, hansı ki, təyin etdiyimizdə nəzərə almalıyıq. Bildiyimiz kimi, bu flag faylin silinməsinin qarşısını alır. Ancaq heç kəs aşağıdakı əmrin qarşısını almir:

```
echo "" > name_file
```

Gördüyüümüz kimi, bu əmr ilə faylin içi tamamilə boşalacaq. Ona görə də **sunlnk** flag-ının **sappnd** flag-ı ilə birgə istifadə edilməsi mütləqdir. Gəlin kiçik bir misala baxaq.

```
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  - 13 Feb 23 14:44 test.file
```

```
root@:/home # echo "test message" > test.file  
root@:/home # cat test.file  
test message
```

```
root@:/home # chflags sunlnk test.file  
root@:/home # ls -lo test.file
```

```
-rw-r--r--+ 1 root  wheel  sunlnk 13 Feb 23 16:22 test.file

root@:/home # rm test.file
rm: test.file: Operation not permitted
root@:/home # echo "" > test.file
root@:/home # cat test.file

root@:/home #
```

Gördüğümüz kimi, biz **test.file**-in içinden bütün informasiyani sildik, yəni biz faylin özünü sile bilməsək də, faktiki olaraq faylin bütün məzmununu silmiş olduq. İndi bu flag-a **sappnd** əlavə edək və eyni işi yenidən yoxlayaq.

```
root@:/home # ls -lo test.file
-rw-r--r--+ 1 root  wheel  sunlnk 1 Feb 23 16:23 test.file

root@:/home # echo "test message" > test.file
root@:/home # cat test.file
test message

root@:/home # chflags sunlnk,sappend test.file
root@:/home # ls -lo test.file
-rw-r--r--+ 1 root  wheel  sappnd,sunlnk 13 Feb 23 16:27 test.file

root@:/home # echo "" > test.file
test.file: Operation not permitted.

root@:/home # cat test.file
test message
```

Gördüğümüz kimi, artıq həm faylin özünü, həm də içində olan informasiyani silmək mümkün deyil.

### Önəmlidir!

Yadınızdadırsa, başlığın əvvəlində təhlükəsizlik səviyyələri haqqında danışmışdıq. Onlar flag-larla birgə işləməkdə çox önəmlü rol oynayırlar. FreeBSD susmaya görə (-1)-ci səviyyə təhlükəsizliyi istifadə edir.

```
root@:/home # sysctl kern.securelevel  
kern.securelevel: -1
```

Bu təhlükəsizlik səviyyəsində UID=0(root) olan istifadəçinin həm flag-ı təyin etmək, həm də silmək yetkisi olur. Təhlükəsizlik səviyyəsini (1)-ədək qaldırıqdır, bu imkan yox olur. Yəni yenidən flag-ların silinməsi üçün sistemi (-1), ya da (0)-a endirməli olacaqsınız.

Flag-larla işlədikdə aşağıdakı əmrlər daha effektiv olacaq

- **ls(1)** - Faylların siyahısı və onlar haqda məlumatın əldə edilməsi;
- **chflags(1)** –Flag-ların təyin edilməsi və silinməsi;
- **security(7)** - Təhlükəsizlik səviyyələri haqqında oxumaq.

Fayla iki istənilən flag təyin edək, sonra onlardan birini silək

```
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  - 13 Feb 23 16:27 test.file
```

```
root@:/home # chflags sappnd,sunlnk test.file  
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  sappnd,sunlnk 13 Feb 23 16:27 test.file
```

```
root@:/home # chflags nosappnd test.file  
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  sunlnk 13 Feb 23 16:27 test.file
```

Öncədən təyin edilmiş schg flag-ı fayldan silək

```
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  schg 13 Feb 23 16:27 test.file  
  
root@:/home # chflags noschg test.file  
root@:/home # ls -lo test.file  
-rW-r--r--+ 1 root  wheel  - 13 Feb 23 16:27 test.file
```

# Indeksler

## A

access 28, 65, 87, 94, 159, 166, 244, 294, 343, 344, 345, 346, 347, 373, 378, 394, 422, 443, 461, 465, 504, 537, 559, 585  
access-list 499  
account 127, 176, 415, 446, 470, 570  
accounting 65, 67, 570  
acls 586, 587  
acltrusted-dns 395  
acpi 27, 28, 165  
action 172, 173, 174, 301, 352, 353, 354, 356, 456, 457, 459, 460, 464  
active 29, 37, 81, 82, 161, 403, 418, 421, 423, 437, 473, 475, 500, 501, 546, 547, 548, 549, 550  
addr 301, 356, 358, 413, 425, 500, 512, 513, 518  
address 61, 196, 204, 205, 206, 298, 336, 337, 362, 382, 451, 473, 475, 483, 488, 494, 498, 499, 500, 505, 525, 551  
adduser 126, 127, 175, 176, 177, 178, 180, 445, 446, 470, 471, 569, 572  
admin 78, 302, 353, 375, 386, 387, 389, 418, 420, 425, 449, 450, 456, 458, 463, 472, 473, 474, 547  
advanced 27, 437  
advskew 552, 554  
aggregation 525, 544, 545  
allow 122, 123, 242, 329, 331, 332, 333, 334, 336, 340, 341, 351, 372, 373, 374, 400, 412, 415, 466, 467, 481, 487, 568, 569  
allow-query 392, 398  
allow-transfer 390, 395, 398  
allowusers 230, 444  
alternate 325, 439, 440, 514  
altq 297, 299, 304, 310, 312, 313, 325, 351, 514, 538  
anacron 125  
anacrontab 125  
anonim 6, 209, 210, 211, 376, 416, 422  
anonymous 169, 209, 211, 378, 422, 483, 488, 504  
antispoof 315  
apache 40, 96, 101, 102, 263, 341, 347, 367, 368, 370, 371, 375, 376, 382, 383, 448, 450, 452  
apache-tomcat 3, 449, 450, 452  
apachectl 100, 370, 376  
area 322, 534, 535, 536  
arpa 389, 390, 395, 396, 398  
arping 219  
asia 22, 273, 276, 277, 278, 397  
asterisk 254, 255  
atheros 199, 201  
attach 68, 151, 153, 173, 174, 245, 280, 291

auth 254, 266, 352, 353, 357, 422, 456, 457, 459, 460  
authconfig 370, 374  
authentication 126, 176, 234, 403, 418, 446, 470, 478, 483, 488, 494, 499  
authname 375, 457, 459, 460  
authority 294, 386, 407  
authorized 230, 236, 237, 240, 517  
authpriv 254, 256  
auto 141, 246, 352, 363, 549  
auto-rotate 302  
autoattach 152

## B

backend 352, 353, 421  
background 73, 289  
backup 75, 153, 161, 194, 226, 227, 269, 523, 552, 553, 554  
badblock 403, 426  
badsect 403, 426, 427  
bandwidth 312, 313, 314  
basename 208, 484, 489  
bash 40, 41, 86, 97, 100, 113, 114, 115, 116, 118, 119, 120, 126, 127, 132, 170, 171, 176, 177, 178, 211, 265, 290, 515  
basic 311, 375  
benchmarks 204  
berkeley 367, 384  
bi-directional 299  
binaries 561, 562  
binary 137, 163, 181  
bind 185, 367, 384, 385, 386, 389, 390, 391, 392, 393, 394, 397, 399, 400, 421  
bind-address 377, 504  
bios 28, 29, 32, 437  
bind 218, 393, 399  
blackhole 496  
block 47, 55, 75, 135, 250, 300, 302, 303, 315, 324, 351, 354, 355, 356, 357, 360, 361, 362, 365, 427, 428, 429, 431, 438, 439, 440, 562, 565, 566, 567  
blocks 227, 427, 431, 433, 434, 438, 440, 442  
blowfish 152, 229, 478, 483, 488  
bool 564, 566, 567  
bridge 61, 297, 321, 322, 323, 324, 325, 327, 491, 525, 537, 538, 539, 540, 541, 542, 543, 546, 548, 549, 550  
bridging 525, 537  
broadcast 207, 417, 522, 523, 538, 540, 541, 542, 546, 547, 548, 549, 550, 551  
broken 264

browser 189, 202, 203, 264, 294, 315, 372, 373, 374, 403, 416, 421, 452  
bruteforce 354  
bsd-ipfw 352, 353  
bsd-sshd 352, 353  
bsdconfig 3, 36, 92, 133, 134, 190  
bsdinstall 26  
buildkernel 167, 185, 326, 330, 351, 473, 480, 486, 515  
buildworld 185  
bunzip 104  
bzcat 104  
bzip 103, 104, 105, 227, 258, 259

## C

cache 81, 82, 348, 377, 423  
call 73, 357  
calls 401, 424  
cancel 15  
capabilities 157  
carp 514, 525, 551, 552, 553, 554  
ccdconfig 146, 147  
cdpd 526  
cdpr 527  
cdrom 28, 36, 76, 172, 587  
center 282, 286, 287, 372, 373, 374  
cert 424, 505, 508  
certificate 505  
certs 504, 505, 507, 508  
chap 457, 459, 464  
chap-msv 464  
char 379  
check 250, 427, 429, 430, 433, 439, 440, 493, 494  
check-state 333, 340  
checksum 96, 155, 171  
chkrootkit 295, 296  
chown 49, 51, 52, 343, 377, 393, 425, 445, 471, 504, 506, 509, 582, 583  
chpass 127, 570  
chunks 35  
cidr 358  
cifs 404, 411  
cisco 120, 121, 122, 169, 199, 468, 475, 477, 491, 495, 498, 499, 500, 525, 526, 527, 538, 544, 545, 547, 548  
clock 277  
clone 441  
cloned 323, 527, 528, 529, 539, 547, 549, 550, 554  
cli 403, 432, 433, 434  
cli 403, 432  
cluster 361, 477, 510, 522, 523  
cname 388, 396  
code 3, 90, 141, 505  
commands 132, 561, 562  
comment 100, 235, 414, 415, 422  
common 373, 505, 525, 551

connect 293, 464, 465, 506, 508  
console 28, 29, 30, 36, 50, 63, 74, 120, 254, 456, 458, 459, 463  
controller 283, 405, 413, 421, 423, 425  
conv 75, 76, 77  
convert 288, 289, 420  
copy 131, 232, 254  
copyright 178, 289, 527, 568  
core 3, 167, 168, 388, 468, 527, 565  
corefile 168  
crash 131  
crdtool 281  
cron 111, 112, 122, 123, 124, 254, 571  
crontab 38, 39, 122, 123, 124, 125, 258, 344, 572  
crypto 152, 480, 486, 498, 499, 500, 501  
cryptomap 498, 499  
cshell 112  
cshrc 112, 228, 563  
csup 183, 184, 185  
ctrl 29, 34, 42, 74, 75, 116, 120, 121, 129, 130, 131, 132, 143, 157, 158, 264, 265, 280, 285, 291, 293, 522, 535, 536  
ctrl-a 280, 281  
ctrl-z 266  
curl 416  
curl 100, 261, 262, 263, 265  
custom 6, 13  
cvsup 93, 94, 102, 181, 183, 184, 185

## D

daemon 57, 63, 209, 212, 230, 253, 254, 259, 275, 276, 297, 343, 356, 405, 406, 407, 456, 472, 512, 513, 521, 522, 526, 561, 562, 566  
daily 125, 164  
database 40, 127, 378, 379, 446, 471, 484, 485, 504, 509  
datasize 179, 565  
date 43, 74, 118, 123, 124, 132, 226, 269, 278, 465, 561  
dbus 101, 156  
deaktiv 111, 126, 154  
deattach 68  
debian 33, 34  
debug 30, 68, 166, 167, 216, 232, 233, 254, 255, 256, 392, 393, 420, 480, 482, 483, 484, 486, 488, 490, 493, 506, 508  
debugging 138, 143, 167, 249, 257, 320  
decode 61  
decompress 104, 500  
decompressed 500  
decrypt 500  
default-gateway 498  
default-key 399  
default-log 393  
default-port 400  
default-server 399

defaultdepth 158  
defaultgroup 177  
defaultrouter 306, 336, 340, 391, 397, 405, 419, 455, 458, 462, 481, 486, 492, 511, 513, 534, 535, 536, 556  
defaults 28, 30, 31, 123, 160, 172, 283  
defaultshell 177  
defunct 63  
degraded 520  
delay 31, 276, 277  
delete 94, 99, 100, 101, 102, 120, 131, 162, 184, 192, 224, 268, 269, 301, 332, 338, 353, 354, 417  
deny 123, 329, 331, 332, 333, 334, 336, 340, 341, 351, 372, 373, 374, 406, 482, 487, 499, 568, 569  
denyusers 230  
describe 379  
description 100, 243, 527  
descriptor 348, 427  
descriptors 349  
designated 540, 541, 542  
designer 72  
desktop 293, 294, 321, 322, 491  
destination 442, 475, 485, 490, 496, 531  
detach 152, 154, 173, 280, 281  
devfs 155, 171, 172, 426, 434, 523, 586, 587  
device 47, 90, 91, 148, 151, 152, 154, 155, 157, 158, 165, 166, 168, 171, 172, 173, 190, 225, 227, 283, 304, 325, 351, 437, 441, 442, 475, 480, 486, 514, 519, 527, 538, 547, 552, 553, 586, 587  
devices 152, 154, 164, 169, 441  
dhclient 25, 173, 191, 550  
dhcp 21, 25, 189, 205, 211, 212, 322, 525, 539, 550, 555, 556, 557  
dhcpcd 211, 212, 557  
dhcrelay 555, 556  
dhcrely 555, 556  
dhcpserver 556, 557  
diff 42, 495, 496  
digest 500  
directory 126, 176, 225, 372, 373, 374, 392, 398, 403, 410, 418, 421, 422, 443, 444, 446, 470, 573, 574  
dirty 431, 520, 521  
disable 31, 226  
disabled 27, 296, 441, 512, 513  
disallow 378, 504  
disk 3, 7, 8, 12, 18, 32, 33, 34, 35, 56, 57, 58, 82, 84, 85, 133, 134, 135, 136, 139, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 153, 159, 164, 174, 179, 221, 223, 224, 228, 241, 245, 247, 250, 251, 252, 264, 268, 269, 281, 282, 403, 412, 426, 436, 438, 440, 441, 510, 517, 522  
diskimage 245  
display 116, 158, 159, 294  
distfiles 95, 96, 99, 194  
distinfo 94, 95, 96  
divert 329, 331, 332, 335, 340, 341  
dmesg 39, 56, 57, 58, 62, 174, 283  
dmidecode 163

dnsadmins 424  
dnstop 196, 401  
dnswalk 400  
doc-supfile 183, 185  
docs 242, 281, 361  
documentroot 372, 373, 374  
domain 191, 196, 211, 218, 253, 273, 367, 384, 385, 386, 387, 388, 391, 397, 400, 403, 405, 408, 410, 412, 413, 415, 418, 419, 420, 421, 422, 423, 424, 425, 471, 472, 551  
domen 371, 374  
down 114, 192, 201, 465, 518, 519, 554  
drop 301, 312, 329, 379, 419, 497, 498, 512, 513  
dst-addr 334  
dst-port 334, 340, 341  
dummynet 299, 327, 328, 329, 330, 339, 350, 454, 538  
dump 54, 137, 179, 221, 222, 223, 225, 226, 227, 228, 247, 292, 392, 398, 431, 440, 441, 442, 586, 587  
dumpdev 455, 458, 511, 513, 556, 557  
dynamic 163, 205, 211, 335, 337, 529

## E

e-maili 178  
early 325  
earth 237  
echo 86, 87, 113, 116, 117, 118, 119, 143, 202, 207, 208, 217, 241, 242, 248, 290, 309, 343, 349, 352, 370, 371, 372, 373, 374, 377, 391, 445, 450, 465, 472, 484, 485, 489, 496, 504, 526, 570, 574, 576, 589, 590, 592, 594, 595  
echoreq 318  
edit 141, 437  
editor 77, 115, 121, 124, 251  
effective 50, 67, 117, 592  
egrep 107, 108  
eject 282  
email 290, 291, 379, 385, 387, 505  
enable 57, 114, 125, 156, 159, 165, 169, 202, 209, 212, 250, 255, 256, 258, 270, 273, 274, 275, 300, 303, 305, 306, 323, 329, 330, 334, 336, 340, 343, 351, 352, 356, 362, 370, 377, 391, 397, 405, 416, 419, 445, 450, 455, 456, 457, 458, 459, 462, 463, 464, 472, 475, 481, 486, 487, 492, 493, 497, 498, 504, 506, 509, 511, 512, 513, 526, 530, 534, 535, 536, 556, 557, 570, 587  
enabled 60, 352, 353, 354  
encapsulation 461, 478  
encoding 157, 285, 450  
encrypt 422, 500  
encryption 150, 483, 488, 494  
enough 442  
enter 16, 23, 31, 39, 126, 127, 134, 143, 153, 176, 211, 243, 244, 291, 378, 407, 408, 423, 436, 446, 470, 472, 505, 509, 517, 527  
enterprise 424  
entries 401, 588

entry 573  
enum 421, 422  
epoch 226  
error 72, 254, 373, 429  
errorlog 373  
esac 208, 484, 489  
escape 28, 165, 292, 500  
esp-sha-hmac 499, 501  
established 214, 333, 341, 484, 490  
ether 36, 166, 173, 205, 206, 334, 336, 497, 538, 540, 541, 542, 546, 547, 548, 549, 550  
etherchannel 544  
etherip 538  
ethernet 166, 173, 189, 190, 191, 199, 212, 525, 537, 538, 539, 540, 541, 543, 545, 546, 547, 548, 549, 550  
etherq 313, 317  
ethers 206, 207, 208  
etyp 69  
euid 50, 117  
everyone 46, 48  
example 265, 528, 529, 530, 531, 532, 534, 535, 536, 551, 552, 553, 554  
examples 183, 185, 300, 356, 516, 533  
exchange 385, 388, 483, 488, 494  
exclude 347  
exclusive 299, 493  
exec 41, 51, 52, 249  
executable 48  
execute 47  
exit 15, 23, 29, 67, 208, 235, 290, 484, 489  
exiting 63, 353  
expire 387, 389, 396, 531, 561, 572  
expired 335  
expires 423  
export 124, 159, 240, 251  
expr 56  
external 57, 170, 174, 301, 312, 314  
extra 268  
extract 93, 228, 351, 454, 469, 470, 503, 507, 514, 533  
extracted 437

## F

fail 297, 350, 351, 352, 353, 354  
failed 93, 252, 500  
failover 525, 544, 545, 548, 549, 550, 551  
false 119, 168, 210, 211, 420, 529, 564, 566, 567  
fast 282, 438, 478, 544, 547  
fastest 93, 94, 184  
fastethernet 498, 499, 500  
fcedit 117  
fdce 496  
fdesc 449  
fdescfs 449  
fdisk 33, 133, 134, 141, 436, 437

fetch 62, 93, 96, 182, 194, 351, 454, 469, 503, 507, 514, 533  
fetch-recursive 96  
field 40, 347, 505  
filesize 179, 438, 565  
filesize-max 566  
filesystem 53, 161, 167, 251, 426, 434, 436, 439, 523  
filter 38, 212, 321, 352, 353, 354  
find 40, 41, 42, 74, 124, 139, 249  
finger 128, 570, 573, 574  
fingerprint 361, 496  
finish 19  
firewall 193, 206, 294, 297, 298, 299, 302, 304, 321, 322, 323, 325, 327, 328, 329, 330, 335, 336, 339, 340, 351, 354, 355, 359, 365, 454, 455, 458, 462, 466, 481, 487, 492, 493, 498, 512, 513, 537, 538, 540, 541, 543, 556  
first 291, 349, 357, 360, 361, 362, 532  
flac 284, 285, 286  
flag 32, 54, 55, 61, 63, 65, 69, 300, 356, 358, 411, 437, 438, 439, 575, 577, 581, 586, 593, 594, 596  
flash 24, 140  
floppy 37, 140  
flow 501  
flow-cache 475  
flush 303, 332, 379, 400, 481, 487, 495  
flushes 520, 521  
fname 379  
folder 49, 104, 270, 412  
foreach 434  
format 8, 56, 136, 140, 142, 148, 155, 172, 178, 225, 399, 521, 568, 572  
forward 295, 330, 334, 350, 384, 386, 444, 540, 555  
forwardable 420  
forwarding 444, 445, 529, 541, 542  
found 153, 196, 296, 441  
fping 196, 197  
fqdn 255, 413, 505  
fragment 312, 438, 439  
frame 289, 456, 459, 539  
free 62, 81, 82, 91, 161, 162, 198, 224, 430, 431, 433, 434, 438, 440, 441, 442  
freebsd 1, 2, 3, 6, 16, 18, 19, 24, 27, 28, 29, 30, 32, 33, 34, 45, 54, 69, 71, 72, 76, 80, 82, 90, 92, 93, 94, 98, 99, 100, 102, 113, 114, 120, 121, 129, 132, 133, 134, 136, 137, 140, 150, 155, 156, 161, 162, 165, 166, 167, 172, 181, 182, 183, 184, 185, 186, 187, 188, 190, 191, 193, 194, 202, 203, 205, 211, 218, 219, 221, 222, 235, 238, 241, 244, 245, 247, 257, 258, 262, 263, 264, 265, 266, 271, 281, 284, 293, 297, 298, 299, 306, 321, 322, 330, 339, 341, 355, 359, 367, 368, 375, 378, 379, 384, 385, 388, 390, 393, 403, 404, 405, 408, 410, 411, 412, 413, 414, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 437, 439, 441, 443, 444, 448, 450, 453, 456, 457, 458, 459, 460, 461, 463, 466, 468, 472, 477, 478, 479, 480, 481, 483, 485, 486, 489, 491, 492, 498, 500, 502, 504, 510, 512, 514, 518, 521, 522, 525, 526, 527, 529, 530, 537, 538, 540, 545, 546, 551, 552, 553, 554, 555, 559, 560,

575, 582, 585, 587, 593, 595  
freebsd-boot 161, 223, 224  
freebsd-cisco 477  
freebsd-ports 469  
freebsd-swap 161, 162, 224  
freebsd-ufs 161, 224  
freebsd-update 181, 182  
freebsdcisco 495, 499  
fs-nø 137  
fsck 77, 138, 152, 161, 221, 225, 248, 249, 427, 431, 433, 436, 439, 440, 498  
fsize 141, 437  
fsprogs 136, 137  
fstab 51, 58, 68, 135, 141, 143, 145, 146, 147, 148, 152, 154, 159, 197, 245, 250, 251, 413, 425, 441, 442, 449, 586, 587  
fstype 135, 141, 437, 519, 586, 587  
ftpchroot 210  
ftpd 43, 209, 210, 211  
ftphosts 210  
ftpmotd 80, 210  
ftpproxy 306  
ftputers 210  
ftpwelcome 80, 210  
full 126, 127, 169, 211, 226, 407, 410, 412, 445, 446, 470, 505  
full-dupleks 545  
full-duplex 192, 193, 549  
fullsync 518, 520, 521  
fundamental 36, 37  
fuse 58, 137, 241  
fusefs 57, 58, 137  
fusefs-ext 137  
fusefs-ntfs 57  
fusefs-sshfs 241  
fuser 71  
fusestart 241  
fwddelay 538, 540, 541, 542

## G

gateway 190, 198, 206, 212, 306, 335, 336, 340, 362, 455, 458, 461, 462, 463, 481, 487, 493, 530, 531, 534, 535, 536, 542, 555, 556, 557  
gecos 561, 562, 572  
geli 153, 154  
gelicrypt 153, 154  
geli 150, 152  
generator 240, 241  
generic 166, 167, 182, 325, 329, 350, 454, 472, 480, 485, 486, 514, 586  
geom 34, 134, 142, 143, 144, 145, 146, 147, 150, 152, 153, 226, 512, 513, 514  
getent 423  
getfac 588, 589, 590, 591, 592

getty 30  
gigabayt 7, 136  
gigabit 190, 203  
gigabitethernet 475, 545, 546  
global 172, 302, 414, 421  
gmail 196, 290, 379, 505  
gmake 448  
gmirror 143, 144, 145  
gnome 3, 159  
gpart 155, 156, 160, 161, 162, 223, 224  
gpswap 224  
grubboot 224  
granted 372, 373, 374  
grep 30, 31, 40, 41, 42, 43, 44, 51, 62, 79, 103, 107, 108, 109, 124, 152, 154, 174, 199, 204, 283, 345, 348, 349, 412, 413, 450, 466, 471, 480, 486, 521, 522, 523, 529, 584  
group 41, 50, 63, 66, 126, 172, 175, 176, 177, 251, 401, 406, 407, 419, 422, 423, 424, 445, 446, 470, 483, 488, 494, 499, 501, 545, 547, 560, 563, 570, 574, 575, 577, 579, 585, 589, 590, 591, 592  
grow 161  
growfs 160, 161  
growisofs 283  
grub 1, 3, 33, 34  
gstat 85, 144  
gstripe 142, 145  
guest 416, 424  
guid 133  
guided 18  
gunzip 76, 104  
gvinum 147, 148, 149  
gzip 76, 103, 104, 105

## H

hacker 40, 296  
hacking 353  
half-gui 181  
halt 29, 34  
hardware 165, 166, 167, 212  
hash 483, 488, 494, 545  
hast 477, 510, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524  
hastctl 519, 520, 523, 524  
hastd 512, 513, 519, 522  
header 281, 478  
hellotime 538, 540, 541, 542  
help 86, 234, 265, 270, 280, 293, 408  
hfsc 304, 312, 313, 314, 325, 351, 514  
hier 36, 39  
hierarchical 325  
hinfo 385, 388  
histfile 115, 116, 117  
histfilesize 116, 117

history 113, 115, 116, 117  
histsize 116  
hmac 483, 488, 494  
hmac-md 399, 400  
hmac-sha 495, 496  
hold 427, 428  
holdcnt 539, 540, 541, 542  
homedir 421, 561  
horizontal 132, 157  
hostname 17, 117, 186, 210, 218, 253, 280, 306, 322, 391, 397, 401, 405, 419, 455, 458, 462, 479, 481, 486, 492, 511, 513, 534, 535, 536, 552, 554, 556, 557, 571  
hosts 36, 116, 191, 229, 235, 246, 253, 255, 256, 273, 292, 302, 370, 372, 374, 401, 412, 415, 420, 517, 569  
hosttype 117  
htop 197  
htpasswd 370, 374, 375  
http 3, 62, 93, 193, 194, 203, 237, 262, 263, 264, 265, 284, 294, 296, 298, 299, 310, 319, 320, 343, 348, 353, 354, 355, 360, 363, 371, 374, 375, 376, 382, 416, 448, 450, 451, 471  
http-server 471  
httpd 69, 70, 370, 371, 374, 377  
httpd-access 370  
httpd-error 370  
https 2, 3, 142, 185, 186, 194, 203, 353, 354, 452, 471, 473  
https-server 471  
hwcsim 546, 548, 549  
hwtagging 546, 548, 549

■

icmp 86, 214, 215, 216, 217, 301, 302, 316, 318, 319, 340, 341, 356, 358, 360, 361, 468, 485, 495, 496, 497, 498, 500, 512, 513  
icmp-type 318, 360, 361  
icmptypes 341  
ident 166, 167, 329, 480, 486, 500, 514  
identifier 63  
identified 379, 504  
identifier 158, 483, 488  
identify 288  
identity 229, 230, 494, 499  
idle 81, 82, 463, 500, 574  
idmap 421  
ieee 200  
if-bound 311  
iface 456, 457, 459, 460, 463, 531, 532  
ifaces 212, 556, 557  
ifconfig 25, 40, 190, 191, 192, 193, 201, 204, 205, 206, 207, 208, 223, 306, 322, 323, 336, 340, 357, 391, 397, 405, 419, 455, 458, 462, 480, 481, 486, 492, 493, 511, 513, 519, 522, 523, 527, 528, 529, 534, 535, 536, 538, 539, 540, 541, 542, 543, 546, 547, 548, 549, 550, 552, 553, 554, 556, 557  
ifcost 539, 540, 541, 542  
ifdisabled 549  
ifmaxaddr 540, 542, 543  
ifnet 173, 518  
ifstat 86, 193  
iftop 195  
ignore 274  
ignored 66  
ignoreip 352  
ignorelogin 566  
ignoretime 180  
in-addr 389, 390, 395, 396, 398  
inconsistency 428, 429, 430  
index 102, 264, 355, 370, 372, 373, 374, 382, 383  
inetq 313  
infected 295  
info 39, 99, 100, 106, 127, 157, 254, 256, 286, 348, 349, 353, 355, 365, 446, 471, 484, 490  
init 28, 29, 30, 151, 153, 154, 519, 523  
init-connect 377  
initial 34, 494  
inode 35, 41, 222, 250, 403, 432, 434  
inodes 251, 438, 439  
input 84, 85, 86, 140, 144, 157, 214, 399  
insert 144, 145, 379, 436  
interface 27, 167, 190, 301, 329, 330, 336, 337, 340, 410, 462, 471, 475, 480, 486, 498, 499, 500, 527, 531, 546, 556  
internal 170, 211, 235, 312, 429  
internal-sftp 445  
iostat 85  
ipc 456, 457, 459, 463, 464  
ipdivert 328, 330, 339, 454, 462, 555  
ipencap 481, 487  
iperf 203, 204  
ipfiltr 299, 355  
ipfirewall 330, 339, 350, 454, 480, 486, 555  
ipfstat 355, 361, 364  
ipfw 297, 299, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 338, 339, 340, 341, 342, 350, 351, 353, 454, 462, 466, 481, 482, 487  
ipmon 355, 356, 365  
ipnat 355, 356, 362, 364  
ipopts 361  
ippool 463  
ipprimer 298  
ipsec 468, 477, 478, 479, 480, 481, 482, 485, 486, 487, 490, 492, 493, 494, 495, 496, 499, 500  
ipsec-isakmp 499  
ipsec-sa 484, 490  
ipsec-tools 479, 482, 487, 492  
ipsekkern 480, 485, 486  
ipstealth 325, 327  
irqs 82  
isakmp 481, 482, 485, 487, 488, 490, 493, 496, 499, 500  
isc-dhcp 211, 556

isc-dhcpd 212, 557  
isisd 534  
issued 423  
icmp 215, 216, 359  
id 50, 67, 541, 551  
ident 376  
inetd 155, 168  
interface 209  
internet 207, 210, 216, 261, 298, 313, 321, 335, 359  
ip 24, 25, 36, 191, 192, 196, 202, 203, 204, 205, 206,  
210, 211, 212, 214, 216, 217, 218, 219, 225, 230, 232,  
234, 236, 238, 239, 242, 255, 256, 268, 272, 274, 277,  
290, 298, 304, 305, 306, 308, 311, 318, 321, 322, 327,  
331, 332, 334, 335, 336, 337, 338, 339, 340, 341, 343,  
344, 345, 346, 355, 358, 359, 360, 362, 363, 367, 370,  
372, 373, 388, 389, 391, 394, 395, 396, 397, 401, 405,  
409, 410, 417, 418, 455, 456, 458, 478, 479, 483, 485,  
489, 491, 495, 499, 502, 504, 506, 508, 525, 528, 529,  
530, 534, 535, 536, 537, 538, 539, 540, 542, 545, 546,  
549, 550, 551, 552, 553, 554  
ipf 355, 356, 359, 362, 365  
ipfilter 297  
ipfiltr 355  
ipfw 297, 327, 328, 540  
ipsec 443, 477, 478, 481, 485, 487, 490, 491, 492, 496,  
500  
ipsec-tool 487  
irssi 293  
isakmp 485, 490, 496  
iso 4, 5, 24, 102, 281

## J

jail 65, 352, 353, 434  
java 3, 443, 448, 449  
jitter 276, 277  
jobjc 62, 63, 65  
jobs 42  
join 293, 423  
jumbo 546, 548

## K

kdestroy 425  
keep 99, 264, 300, 301, 317, 324, 355, 356, 357, 359,  
360, 362, 364  
keep-alive 457, 459, 460, 464  
keep-state 332, 333, 334, 340  
keepdirty 520, 521  
keepstate 333  
kerberos 419, 420, 421  
kern 34, 134, 143, 167, 168, 254, 348, 349, 420, 469, 497,  
498, 596

kernconf 167, 326, 330, 473, 480, 486, 515  
kernel 28, 29, 31, 34, 38, 40, 42, 54, 55, 61, 62, 63, 71,  
82, 140, 143, 150, 155, 163, 165, 166, 167, 168, 182, 214,  
250, 282, 300, 304, 325, 326, 329, 330, 334, 339, 348,  
349, 355, 454, 469, 480, 485, 486, 498, 512, 513, 514,  
515, 529, 538, 552, 553  
keygen 231  
keys 230, 236, 237, 240, 400, 517  
kill 43, 71, 73, 258, 294  
killall 43, 73, 484, 489  
kilobyte 85  
kldload 58, 142, 143, 145, 150, 153, 165, 241, 282, 300,  
328, 552, 553  
kldstat 165, 300, 302  
kldunload 166, 328  
klist 423  
kmem 561, 562  
ktrace 65

## L

label 9, 65, 134, 135, 142, 143, 145, 162, 566  
lacp 525, 544, 545, 546, 547  
lagg 544, 546, 547, 548, 549, 550  
lagghash 546, 547, 548, 549  
laggport 546, 547, 548, 549, 550  
laggproto 546, 547, 548, 549, 550  
lame 285, 286, 287  
lang 380, 566  
laptop 549  
ldap 263  
ldconfig 42, 159, 160  
leases 191, 212  
leave 126, 176, 445, 446, 470, 505  
length 148, 207, 485, 493, 496  
less 28, 41, 64, 72, 78, 100, 102, 108, 117, 136, 137, 270,  
288  
lessopen 117  
level 226, 232, 422, 439, 498  
lftp 94, 96, 98, 261, 265, 266  
libalias 330, 350, 454  
libexec 30, 37, 40, 170, 209, 210, 211, 371  
library 114, 160, 411  
libs 420  
lifo 179  
limit 29, 94, 179, 250, 251, 252, 311, 312, 328, 330, 334,  
339, 341, 350, 372, 373, 454, 480, 486, 543, 555  
limit-rate 266  
limitrequestbody 371  
limits 251  
limitxmlrequestbody 371  
line 107, 134, 191, 209, 230, 254, 257, 285, 480, 486  
lines 117  
link 35, 37, 47, 55, 56, 68, 137, 172, 173, 205, 264, 269,  
278, 296, 324, 334, 336, 376, 456, 457, 458, 459, 460,

464, 497, 525, 538, 544, 545, 548  
linproc 197  
linprocfs 197  
linux 3, 33, 34, 36, 51, 89, 111, 129, 134, 137, 159, 170, 197, 203, 237, 403, 411  
list 41, 71, 124, 132, 144, 170, 195, 290, 298, 301, 331, 394, 395, 559, 564, 566, 567, 568, 569, 574, 576, 585, 593  
listen 214, 370, 475, 482, 488, 493  
listen-on 392, 398  
live 24, 228  
loadbalance 544  
localhost 159, 192, 204, 211, 214, 231, 236, 275, 277, 353, 385, 388, 399, 400, 517  
localnet 343  
locked 127, 176, 406, 446, 470, 562  
logfile 274, 303, 305, 306, 323, 353, 512, 513  
logging 28, 328, 351, 353, 393, 420  
login 30, 65, 112, 126, 128, 176, 177, 178, 180, 239, 378, 445, 446, 463, 464, 465, 470, 504, 561, 562, 563, 564, 566, 568, 572, 574, 577  
logout 115, 463, 464, 465  
logpath 352, 353, 354  
logs 343, 344, 345, 346, 347

**M**

mac-address 206  
machine 388, 406  
machtype 117  
maclabel 566  
macos 237  
macros 323  
mail 98, 113, 116, 117, 124, 150, 178, 179, 196, 215, 216, 217, 218, 254, 279, 281, 290, 343, 363, 385, 388, 389, 394, 396, 397, 467, 468, 563, 566, 573, 574  
maillog 51, 52, 53, 128  
mailq 290  
mailrc 113  
main 52, 455, 494  
make 57, 69, 77, 95, 96, 97, 98, 99, 113, 114, 136, 161, 163, 164, 167, 185, 186, 187, 194, 195, 196, 197, 199, 202, 204, 211, 216, 279, 283, 284, 285, 288, 295, 326, 330, 339, 342, 344, 351, 352, 368, 370, 377, 380, 382, 390, 391, 400, 401, 404, 405, 441, 448, 449, 454, 462, 469, 470, 473, 480, 482, 486, 503, 507, 515, 516, 526, 527, 533, 555, 556  
makefile 94, 95, 187  
malloc 223  
manpath 566  
manual 39, 439  
mapping 226, 227  
mask 32, 190, 275, 358, 415, 468, 500, 531  
mass 140, 286, 287  
master 8, 28, 32, 128, 133, 176, 178, 194, 273, 367, 385, 387, 390, 391, 394, 395, 396, 397, 398, 399, 406, 407, 421, 544, 548, 549, 550, 551, 552, 553, 554, 560, 561, 562, 564, 566, 567, 570, 572  
max-redial 457, 459, 460  
max-src-nodes 318  
max-src-states 318  
maxaddr 539, 540, 541, 542  
maxdepth 139, 249  
maxfiles 168, 348, 349, 420  
maxfilesperproc 349, 420  
maximum 61, 349, 493  
maxkeepaliverequests 371  
 maxlen 469  
maxproc 168, 179, 565  
maxsockbuf 469  
mbuf 411  
mdconfig 76, 91, 245, 246, 248, 282, 438  
mdmfs 223, 245  
media 3, 13, 37, 174, 192, 193, 437, 546, 547, 548, 549, 550  
media-type 173  
mediaopt 192, 193  
megabit 192, 203  
membership 401  
memfs 221, 243  
memory 28, 60, 61, 62, 64, 65, 68, 84, 86, 90, 91, 167, 214, 223, 244, 246, 282  
memsync 518  
messages 42, 78, 109, 128, 173, 256, 472, 518, 523, 574  
metric 538, 540, 541, 542, 546, 548, 549, 550  
mget 407  
microsoft 273  
minpasswordlen 128, 569  
mirror 143, 144, 145, 146, 147, 149, 194, 264, 266  
mirrors-ftp 194  
missing 430, 434  
mixer 283, 285  
mkfs 136, 137  
mkisofs 281, 282  
mksnap 225, 248  
mode 27, 28, 29, 127, 170, 172, 176, 179, 422, 429, 430, 433, 439, 442, 446, 470, 483, 488, 494, 495, 496, 546, 547  
monitor 158, 195, 356, 543  
more 28, 39, 43, 114, 115, 118, 429, 571  
mtd 81, 179, 210, 498, 566, 567  
mount 12, 25, 36, 37, 39, 40, 56, 57, 58, 68, 76, 77, 92, 102, 135, 136, 137, 138, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 159, 172, 174, 197, 223, 224, 228, 242, 245, 246, 248, 249, 267, 272, 282, 404, 411, 412, 413, 425, 431, 432, 434, 436, 438, 440, 449, 516, 519, 521, 562, 578, 580, 586, 587  
mountd 270, 271  
mounted 426, 427, 433, 434, 438, 440, 523  
mountpoint 519, 586, 587  
mpls 203  
mppc 456, 457, 459, 464

msdosfs 58, 138, 249

msinfo 282

mtime 429, 430, 433

multi 282

multicast 276, 361, 538, 540, 541, 542, 546, 548, 549, 550

multilabel 586, 587

mygroup 406, 407, 414

mysql 97, 101, 197, 341, 367, 368, 377, 378, 379, 380, 502, 503, 504, 505, 507, 508, 509

mysql-server 101, 377, 504

mysqladmin 378

mysqld 377, 504

mysqldump 378, 379

## N

named 201, 384, 385, 390, 391, 392, 393, 394, 395, 397, 398, 399, 400  
names 377, 388  
nameserver 191, 391, 396, 397, 418  
nat-t 479  
natd 328, 329, 330, 335, 336, 337, 340, 462, 556  
ndis 199  
neighbor 534, 535, 536, 547  
netbios 405, 406, 414, 417, 418, 421, 423, 468  
netbsd 355, 437  
netflow 443, 468, 471, 475  
netgraph 538  
netif 193, 455, 458  
netmask 25, 190, 191, 193, 201, 212, 223, 306, 322, 336, 340, 405, 410, 419, 455, 458, 462, 480, 481, 486, 493, 511, 513, 519, 522, 523, 528, 529, 534, 535, 536, 540, 542, 546, 547, 548, 554, 556, 557  
netsmb 411  
netstart 193  
netstat 213, 214, 255, 302, 450, 475, 480, 486  
network 166, 173, 190, 206, 214, 254, 270, 271, 273, 274, 275, 293, 298, 307, 314, 315, 316, 317, 320, 401, 511, 513, 534, 535, 536, 538  
networking 167, 190  
networks 401, 420  
newfs 56, 135, 141, 142, 144, 145, 146, 147, 148, 149, 151, 154, 224, 228, 246, 438, 439, 521  
newkern 472, 473  
newsyslog 212, 221, 253, 258, 259, 302, 352, 365  
newuser 175, 176  
next 39, 129, 227  
nextkernel 166  
nfsd 270, 271  
nginx 96, 102, 353  
nginx-access 353  
nginx-error 354  
nice 65, 72, 73, 81, 82, 83, 179, 566, 567  
nload 194, 195, 516  
nmbd 406, 421

nmblookup 417

nmbdb 421

nobody 127, 169, 196, 320, 416  
node 225, 226, 246, 510, 511, 513, 514, 517, 518, 519, 520, 521, 522, 523, 524

nodump 54, 225, 593

noerror 75, 76, 77

nogroup 127

nohup 73

nologin 126, 127, 176, 179, 211, 446, 470, 471, 561, 562, 563, 566

none 29, 61, 154, 177, 212, 256, 372, 373, 378, 390, 398, 406, 407, 500, 557, 586, 587

nowait 169, 210, 211, 416

nslookup 385, 389, 392, 417

nsswitch 401, 419

ntfs 57, 58, 134, 174, 437

ntop 443, 468, 469, 470, 471, 472

ntp 273, 274, 275, 277

ntpdate 273, 274, 275, 277, 397, 418

ntpq 274, 276

ntptrace 274

null 40, 41, 51, 191, 207, 208, 291, 292, 345, 393, 403, 563

nullfs 403, 432, 434, 435

numbers 85, 298

nvclock 164

nvidia 164

## O

oggenc 285, 286

ogginfo 286

oldpwd 117

only 57, 177, 309, 337, 415, 421, 422, 494

opaque 225, 593

open 225, 227, 265, 327, 329, 330, 455, 456, 457, 458, 459, 460, 462, 463, 493, 532, 538, 556

openbsd 191, 355

openfiles 179, 565

openjdk 448, 449, 450

opensource 377

openssh 221, 229, 240, 443, 444

openssl 444, 505

operating 167

operator 561, 562, 573

oracle 3

order 372, 373, 374

originate 535

orijinal 106, 211, 215, 244, 264, 281, 544, 553, 554

osize 83

ospf 526, 532, 533, 534, 535, 536

ospfd 533, 534, 535, 536

ostype 117

other 46, 126, 242, 298, 310, 312, 313, 314, 316, 318, 319,

446, 470, 497, 571, 575, 577, 579, 585, 588  
overwrite 347  
owner 74, 407, 429, 430, 433, 561, 562, 575, 579, 589,  
590, 591, 592  
owners 424

## P

package 2, 96  
packages 2, 102, 469  
packet 32, 166, 213, 299, 324, 325, 500  
packet-level 302  
packets 527  
paddr 66  
page 66, 83, 84, 100, 118  
pagedown 120  
pagein 66  
pages 216, 264  
paging 89  
paiging 90  
pasive 179  
pass 176, 194, 226, 227, 300, 301, 302, 303, 305, 307,  
315, 316, 317, 318, 319, 320, 324, 357, 359, 360, 362,  
364, 365, 512, 514, 515, 518, 552, 554, 586, 587  
passphrase 153, 237  
passwd 39, 40, 42, 43, 50, 51, 122, 128, 176, 177, 178,  
180, 210, 223, 237, 294, 401, 414, 419, 420, 423, 471,  
560, 561, 562, 563, 564, 566, 567, 570, 572, 573, 578,  
580  
passwdqc 569  
passwdtype 177  
password 126, 127, 169, 176, 211, 225, 226, 264, 294,  
378, 401, 407, 408, 410, 413, 421, 423, 424, 425, 446,  
450, 457, 459, 460, 470, 472, 504, 509, 561, 562, 563,  
568, 572, 573  
password-based 126, 176, 446, 470  
passwords 422  
paste 131  
pathnames 429, 433, 440  
payload 478  
pciconf 60, 61, 163  
pciutils 199  
peer 276, 277, 457, 458, 459, 460, 499, 500  
peerpassword 465  
peers 311, 318, 319  
pending 66, 438  
periodic 98, 123, 125, 164  
perl 62  
perm 41, 172  
permission 40, 52, 589, 592  
permissions 46, 126, 176, 446, 470  
permit 230, 301, 499, 500  
permitted 432, 436, 438, 595  
persist 300, 311, 354  
pfctl 300, 301, 302, 305, 307, 314, 320, 323, 354

pflog 302, 303, 304, 305, 306, 323, 325, 326, 351, 512,  
513, 514  
pflogd 301, 326  
pfserver 414  
pfsync 304, 325, 351, 514  
pftop 320, 326  
pgid 62, 63, 66  
pgrep 70, 72  
phase 427, 429, 430, 433, 440, 485, 490, 494, 496  
phpinfo 382  
ping 191, 195, 196, 215, 216, 217, 219, 485, 495, 500,  
529, 534, 535, 536  
pipe 55, 334, 338  
ping 216  
pipe 338  
pkill 523  
pkts 500  
point 144, 201, 537  
policy 157, 300, 424, 485, 499  
port 2, 3, 37, 57, 61, 69, 77, 79, 92, 93, 94, 96, 97, 98, 99,  
102, 113, 163, 164, 168, 169, 186, 187, 191, 194, 195, 197,  
202, 203, 204, 211, 214, 216, 218, 230, 239, 256, 283,  
291, 298, 299, 301, 302, 304, 305, 307, 308, 309, 314,  
315, 316, 317, 318, 319, 320, 321, 324, 326, 332, 334,  
336, 337, 339, 342, 343, 344, 348, 349, 351, 353, 354,  
356, 358, 360, 361, 362, 363, 364, 368, 370, 373, 374,  
377, 380, 390, 400, 401, 404, 405, 416, 444, 449, 463,  
468, 469, 475, 482, 494, 500, 527, 538, 539, 540, 541,  
542, 544, 545, 547, 555, 556  
ports 2, 33, 40, 41, 57, 69, 71, 74, 77, 87, 88, 92, 93, 94,  
95, 96, 97, 98, 99, 102, 113, 114, 136, 137, 163, 164, 184,  
186, 187, 188, 194, 195, 196, 197, 199, 202, 204, 211,  
216, 218, 279, 283, 284, 285, 287, 288, 291, 295, 302,  
305, 320, 326, 337, 339, 342, 344, 351, 368, 377, 380,  
382, 390, 400, 401, 404, 416, 434, 435, 438, 441, 448,  
449, 454, 462, 469, 482, 487, 503, 507, 512, 513, 515,  
516, 526, 527, 533, 555, 556  
ports-all 94, 184  
ports-mgmt 97, 98, 99  
ports-supfile 94, 183, 185  
portsnap 36, 93, 101, 351, 454, 469, 503, 507, 514, 533  
portupgrade 97, 99  
posix 107, 167, 587  
postfixadmin 371  
power 27, 34, 128  
ppid 62, 63, 64, 66, 118  
pre-shared-key 495  
pre-sharedkey 399, 400  
printf 41  
priority 66, 179, 256, 325, 538, 539, 540, 541, 542, 547,  
566, 567  
priq 304, 312, 325, 351, 514  
private 229, 230, 240, 321, 360, 462, 505, 541, 543  
prob 216  
problem 57, 182, 233, 292, 364, 376, 399, 518, 523, 527,  
548, 549, 553, 562  
proc 37, 68, 159, 197, 449

procedure 401, 441, 442  
process 29, 63, 65, 66, 67, 81, 118, 280  
processl 290  
procfs 37, 68, 159, 449  
procs 68, 84  
progress 234  
project 124  
protected 424, 500  
proto 301, 302, 305, 307, 314, 315, 316, 317, 318, 319, 320, 324, 356, 358, 359, 360, 361, 362, 364, 401, 539, 540, 541, 542  
protocol 186, 191, 205, 209, 211, 254, 273, 301, 358, 461, 525, 526, 527, 538, 539, 540, 545, 551  
protocols 131, 132, 192, 301, 401, 420  
protocomp 464  
proxy 193, 194, 276, 293, 307, 314, 315, 320, 339, 340, 341, 343, 344, 363, 364, 415, 494  
proxy-arp 463  
pscp 239  
pseudo 90  
pseudo-ttys 166  
pseudo-user 561, 562  
psftp 239  
pstfp 239  
pstree 69, 70  
public 93, 229, 230, 235, 237, 240, 241, 254, 313, 314, 316, 317, 318, 321, 335, 362, 370, 415, 416, 417, 455, 457, 458, 459, 460, 462, 464, 465, 479, 510  
puttygen 239

## Q

quagga 533, 534, 535, 536  
quake 41  
qualified 412  
quantifiers 108  
queries 325, 393, 514  
query 185, 377, 393, 394  
querying 417  
querylog 393  
queue 74, 312, 313, 314, 315, 316, 317, 318, 319, 320, 338, 419, 469, 512, 513  
queueing 300  
queuing 325  
quit 293  
quota 221, 247, 250, 251, 252  
quotaoff 252  
quotaon 251  
quotas 250, 251

## R

racoont 481, 482, 483, 484, 487, 488, 489, 490, 493, 495,

499  
racoontl 485, 490, 496  
radius 456, 459  
raid 142, 143, 145, 146, 147, 148, 149, 269  
random 118, 126, 153, 154, 166, 169, 176, 211, 244, 325, 446, 470, 496  
range 60, 61, 212, 362, 421, 557  
rbash 126, 176, 211  
read 47, 51, 52, 57, 58, 68, 69, 136, 272, 415, 422, 591  
readonly 248, 438  
real 25, 62, 66, 86, 91, 205, 248, 303, 314, 320, 431, 441, 455, 463, 477  
reboot 28, 29, 30, 31, 34, 58, 91, 144, 145, 165, 326, 330, 351, 473, 515  
reconnect 280  
record 8, 28, 32, 133, 385, 386  
recursive 104  
recv 335, 336, 500  
redirect 314, 337, 497, 498  
redundancy 525, 551  
refresh 387, 389, 396, 421  
regex 108  
regular 226, 227  
rehash 163, 164, 194, 326, 400, 469  
relay 525, 555, 556, 557  
release 3, 93, 94, 182, 183, 184  
releng 184, 187  
reload 301, 330, 365, 378, 400, 504  
remote 230, 253, 276, 401, 443, 461, 465, 483, 488, 494, 500, 517, 518  
remove 2, 125, 378, 429, 430, 433, 504, 573  
renew 420  
repair 439  
replay 495, 496, 501  
replication 424, 520, 521  
reply 485  
report 170, 347, 518  
repository 181, 183, 186  
request 217, 343, 371, 496, 505  
require 294, 372, 373, 374, 375, 481, 485, 487, 490, 495  
reset 86, 410, 436  
resize 160, 161, 288  
resized 161  
resolve 214, 347  
restart 78, 169, 193, 207, 208, 212, 231, 255, 271, 330, 349, 382, 391, 392, 397, 398, 399, 400, 445, 466  
restore 153, 221, 222, 225, 226, 228, 247, 440, 441, 442  
restrict 274, 275, 422  
retry 94, 387, 389, 396  
rmuser 128, 569, 572, 573  
rndc 399, 400  
rndc-confgen 399  
rndc-key 399, 400  
role 450, 519, 520, 521, 523, 524, 540, 541, 542  
rolename 450  
roles 450  
rotate 258, 288, 289

router 170, 205, 335, 336, 340, 455, 458, 468, 475, 477, 491, 498, 499, 526, 527, 529, 530, 534, 535, 536, 549  
routers 169, 212, 557  
routing 189, 193, 198, 213, 214, 454, 461, 525, 526, 529, 532, 533, 534, 535, 536, 557  
rpcbind 270, 271  
rrdtool 468  
rstp 539, 540, 541, 542  
rsync 261, 267, 268, 269  
rprio 66  
rule 311, 355  
rules 157, 172, 305, 306, 323, 328, 335, 356, 362, 364, 419, 512, 513  
running 81, 225, 348, 540, 541, 542, 546, 548, 549, 550  
runscript 52, 53  
ruser 64, 66

## S

safe 27, 441  
save 240, 474  
sbsize 179, 565  
schg 54, 55, 225, 593, 594, 596  
screen 34, 42, 158, 279, 280, 281, 285  
script 36, 51, 52, 122, 234, 235, 294, 328, 336, 340, 344, 345, 347, 351, 455, 463, 481, 487, 511  
scripts 62, 463, 464, 465  
scrub 307, 312, 324  
scsi 140, 282  
search 96, 130, 191  
secondary 385, 518, 520, 521, 522, 523, 524  
secret 399, 400, 424, 455, 458, 460, 463, 465, 467  
section 158, 505  
sector 32, 141, 151, 437  
sectors 437, 439  
secure 29, 30, 232, 378, 504  
securelevel 498, 596  
security 37, 77, 98, 168, 295, 351, 414, 419, 421, 478, 482, 484, 485, 487, 490, 496, 497, 503, 507, 593, 596  
sendmail 73, 290, 353, 419, 512, 513  
serial 30, 32, 191, 387, 389, 395, 396  
server 17, 24, 29, 30, 56, 94, 116, 125, 138, 168, 169, 184, 195, 196, 203, 204, 205, 209, 211, 218, 222, 225, 229, 230, 233, 236, 248, 253, 254, 255, 256, 257, 270, 273, 274, 276, 277, 290, 294, 298, 299, 304, 306, 317, 320, 339, 344, 353, 361, 363, 367, 368, 370, 371, 376, 377, 382, 384, 385, 387, 388, 391, 397, 403, 404, 405, 406, 407, 408, 410, 411, 412, 414, 415, 416, 417, 418, 420, 421, 422, 436, 443, 444, 450, 451, 457, 461, 462, 463, 473, 478, 491, 497, 503, 505, 525, 530, 532, 533, 535, 537, 545, 551, 552, 554, 555, 556, 585  
services 131, 132, 192, 209, 298, 401, 420, 512, 513  
setenv 59, 160, 179, 449, 566, 567  
sefctl 588, 589, 590, 591, 592  
setgid 45, 46, 50, 51, 53, 66, 506, 508, 578, 579, 580  
setgid 45  
setkey 481, 484, 487, 490, 495  
settings 501  
setuid 45, 46, 50, 51, 52, 53, 66, 506, 508, 578, 579, 580  
setuid 45  
setup 202, 333, 340, 341  
sflow 468  
sftp 233, 234, 263, 265, 266, 443, 444, 445  
shaper 299  
share 33, 36, 112, 114, 115, 138, 157, 177, 178, 183, 185, 277, 278, 300, 356, 410, 415, 417, 422, 425, 503, 516, 533, 563, 566, 567, 572  
shared 71, 160, 198, 482, 483, 488, 489, 493, 494  
shell 23, 30, 111, 112, 113, 115, 118, 119, 120, 126, 127, 128, 170, 176, 178, 211, 421, 446, 470, 561, 562, 563, 566, 567, 572, 574  
show 28, 161, 224, 301, 323, 329, 334, 335, 338, 347, 379, 500, 509, 547  
showmount 271  
shutdown 29, 31, 34, 128, 182, 183, 225  
signt 72, 120  
sigkill 72, 73, 572  
signal 63, 72, 73  
simplex 538, 540, 541, 542, 546, 548, 549, 550  
site 194, 282, 458  
size 32, 41, 60, 61, 75, 82, 90, 141, 151, 245, 263, 265, 377, 415, 422, 426, 429, 430, 433, 434, 437, 438, 439, 441, 442, 494, 501, 523, 565  
skew 552  
skipto 332, 340  
slave 273, 367, 387, 390, 391, 394, 395, 396, 397, 398, 399  
slow 377, 547  
smart 164, 467  
smartctl 164  
smartd 165  
smbclient 408, 409, 421, 422  
smbd 405  
smbfs 165, 411, 412, 413, 425  
smbpasswd 409, 414  
snapshot 3, 222, 225, 226, 247, 248  
socket 55, 71, 332, 337, 497  
sockets 65, 253, 337  
soft 35, 55, 225, 250, 251, 375, 428, 429, 430, 495, 496  
source 3, 114, 115, 170, 181, 183, 441, 475, 485, 490, 496, 500, 561, 562  
spamd 72, 73  
spanning 525, 538, 539, 540, 541  
spawning 523  
spdflush 481, 487, 495  
speed 195, 282  
speedtest 195  
squid 297, 333, 339, 341, 342, 343, 344, 345, 346, 348, 349  
srcip 518  
ssh-a 227, 350, 360  
ssh-ipfw 352, 353

ssh-keygen 229, 235, 236, 517  
sshd 23, 25, 78, 82, 169, 223, 229, 230, 231, 306, 323, 340, 391, 397, 419, 444, 445, 455, 458, 462, 481, 486, 492, 511, 513, 516, 534, 535, 536, 556, 557  
ssid 201, 549  
stable 186, 187, 188, 245  
state 62, 63, 64, 66, 82, 83, 300, 301, 305, 311, 317, 324, 335, 355, 356, 357, 359, 360, 362, 364, 495, 496, 500, 505, 523, 540, 541, 542, 547, 553  
stateful 299, 327, 334  
stateless 355, 456, 457, 459, 464  
static 198, 207, 208, 336, 456, 457, 459, 460, 481, 487, 493, 530, 531, 532, 557  
staticarp 207, 208  
stderr 42, 392  
sticky 45, 50, 51, 53, 55, 542, 578, 579, 580  
strings 44  
stunnel 477, 502, 503, 504, 505, 506, 507, 508, 509  
subdomain 386, 388  
subnet 212, 316, 468, 494, 557  
sudoers 77  
suid 50  
super 168, 209, 439, 440, 581  
superuser 48, 119, 123, 561, 562, 593  
svnup 181, 185, 187, 188  
swap 9, 12, 66, 79, 81, 82, 83, 86, 89, 90, 91, 134, 154, 162, 198, 223, 224, 437, 441, 442, 586, 587  
swapfile 90  
swapinfo 90, 91, 154  
swapon 91  
swapping 89, 90  
switch 280, 526, 527, 537, 539, 544, 545, 546, 547  
sync 34, 75, 76, 77, 274, 275  
synfin 329, 419, 498  
sysinstall 3, 36, 92, 93, 102, 133, 134, 181, 190  
syslog 212, 221, 253, 254, 255, 256, 257, 258, 365, 391, 397, 398, 462, 466, 529  
syslog-**ng** 221, 253, 257, 258  
stat 43, 85, 86, 193  
system 28, 29, 31, 52, 66, 73, 81, 82, 122, 147, 148, 149, 155, 159, 161, 167, 171, 172, 173, 218, 221, 243, 244, 246, 247, 248, 283, 374, 407, 410, 420, 431, 434, 436, 438, 439, 440, 469, 561, 562  
sysutils 33, 57, 69, 71, 87, 88, 136, 137, 163, 164, 184, 197, 199, 202, 279, 283, 320, 326, 438, 441

## T

table 133, 300, 305, 311, 351, 352, 353, 354, 379  
tables 378, 379, 504  
tail 42, 174, 287, 346, 353, 392, 394, 397, 398, 399, 484, 490, 493, 523  
tape 45, 59, 225, 227, 410  
tcpdump 303, 305, 308, 309, 326, 485, 496, 538, 543  
tcpflags 309, 333

tcpflow 309  
tcpmssfix 463  
tcptraceroute 215, 216  
tcpudp 298  
tcsh 126, 127, 176, 211, 446, 470  
tcshrc 112  
telnet 166, 263, 534, 535, 536  
temp 49, 243, 244, 267, 268, 450  
tempfs 221, 243  
terminal 29, 30, 36, 37, 62, 63, 64, 66, 67, 73, 111, 112, 120, 534, 535, 536, 546, 566, 567, 568, 569  
testbed 436, 437, 438, 439, 440  
tftp 168, 169, 170, 254  
tftpd 169, 170  
ticket 420  
timeout 33, 315, 371, 475, 500, 539, 540, 541, 542  
timestamp 331  
timezone 566, 567  
tmpfs 244, 245  
token 234, 517  
tomcat 443, 448, 449, 450, 451, 452  
traceroute 215, 216  
trahshow 195, 196  
transparent 297, 315, 339, 340, 341, 342, 343, 344  
tree 288, 525, 538, 539, 540, 541  
tsiz 64, 67  
ttys 29, 30, 36, 37, 80, 108, 568, 569  
ttyv 30, 63, 574  
tunefs 226, 586, 587  
txcsum 546, 548, 549  
type 32, 40, 41, 60, 139, 157, 159, 172, 173, 176, 249, 329, 330, 336, 340, 351, 356, 358, 359, 390, 395, 398, 436, 455, 458, 462, 481, 487, 493, 500, 556  
types 52, 376

## U

ucarp 512, 513, 514, 516, 518, 519, 521, 522, 523  
ugen 57  
uids 497  
umount 58, 76, 145, 148, 154, 160, 161, 223, 246, 282, 427, 432, 586, 587  
uname 43, 128, 167, 232, 290, 325, 329, 454, 472  
unicast 276  
union 223  
unions 223  
uniq 309  
unison 261, 267, 269, 270  
unit 505  
unix 29, 45, 51, 59, 107, 108, 129, 132, 167, 171, 172, 230, 235, 238, 244, 253, 292, 293, 394, 403, 406, 411, 414, 415, 422, 437, 438, 526, 560  
unmount 77, 246  
unreachable 196  
unreadable 422

up-script 463  
update 32, 93, 181, 182, 185, 186, 187, 351, 428, 429, 430, 440, 454, 469, 498, 503, 507, 514, 533  
usage 208, 251, 484, 489  
user 13, 28, 29, 41, 50, 51, 54, 55, 56, 62, 63, 64, 66, 67, 81, 117, 119, 122, 127, 167, 169, 172, 194, 210, 211, 225, 250, 251, 252, 254, 264, 279, 320, 346, 347, 348, 379, 406, 414, 417, 418, 444, 445, 446, 450, 455, 456, 458, 463, 465, 470, 471, 472, 504, 572, 573, 574, 577, 585, 589, 590, 591, 592  
useradd 127, 572, 573, 574  
usermod 127  
usermount 172, 242  
username 82, 126, 127, 128, 169, 211, 406, 410, 412, 414, 445, 446, 450, 470  
userquota 250, 251  
users 43, 176, 347, 378, 415, 421, 422, 424, 425, 504, 570  
usertab 347  
utilit 140, 164, 197, 203, 215, 229, 399, 400, 573

## V

valid 401, 415, 422  
vendorname 158  
verbose 28, 31, 101, 103, 104, 136, 165, 235, 268, 269, 288, 328, 330, 339, 350, 410, 454, 480, 486, 555  
verify 294, 500  
version 32, 102, 116, 157, 229, 230, 231, 238, 239, 290, 291, 406, 421, 450, 475, 527  
vesa 31, 165  
vhid 511, 512, 513, 514, 552, 553, 554  
view 131, 417, 475  
vipw 180, 562, 570  
virtual 30, 36, 67, 68, 69, 76, 80, 84, 86, 89, 140, 143, 145, 146, 149, 153, 167, 172, 210, 243, 246, 248, 293, 304, 321, 322, 323, 355, 357, 372, 373, 374, 465, 475, 491, 495, 496, 499, 510, 512, 513, 517, 519, 520, 521, 523, 525, 527, 532, 544, 546, 548, 550, 551, 552, 553, 565  
vlan 166, 525, 527, 528, 529, 542, 543, 545, 546, 548, 549, 550  
vmnet 321, 322, 323, 510, 532  
vnode 76, 91, 245, 246, 248, 282, 438  
voip 169, 204  
volume 32, 147, 148, 285  
vpnipp 466, 467  
vsiz 64  
vysh 534, 535, 536

## W

wait 170, 442  
waiting 472, 527  
warn 256, 442  
wbinfo 424  
wcpu 82, 83  
wget 261, 262, 263, 264, 265, 515  
wheel 41, 49, 51, 52, 53, 243, 291, 406, 427, 431, 445, 573, 576, 578, 580, 583, 584, 589, 590, 591, 592, 593, 594, 595, 596  
whereis 40, 94, 441, 454, 469, 516  
which 40, 91, 121, 127  
whoami 43, 235, 570  
whois 219  
wicontrol 201  
wifi 201, 545, 549  
will 94, 294, 442, 505  
winbind 406, 419, 421  
winbindd 419  
winscp 446  
winserver 412, 413, 425  
wired 81, 82  
wlan 40, 200, 201, 549, 550

## X

xargs 41, 42  
xauth 294  
xcalc 159  
xhost 159  
xorg 156, 157, 158  
xstartup 294, 295  
xstat 67  
xtarttup 294

## Z

zcat 104, 571  
zebra 534, 535, 536  
zero 86, 90, 141, 145, 246, 251, 252, 283, 292, 331, 522  
zero-sa 90  
zone 218, 384, 385, 386, 388, 390, 395, 398, 399  
zoneinfo 36, 277, 278, 566, 567

# **İstifadə olunmuş ədəbiyyat siyahısı**

1. Absolute\_FreeBSD\_\_The\_Complete\_Guide\_to\_FreeBSD\_\_2nd\_Edition
2. Absolute BSD The Ultimate Guide to FreeBSD
3. Операционная система UNIX
4. Christopher Negus - BSD UNIX Toolbox 1000 plus Commands for FreeBSD OpenBSD and NetBSD – 2008
5. FreeBSD\_backup
6. FreeBSD-Handbook
7. Лукас М.FreeBSD.Подробное руководство.2 изд.Символ+. [RUS,857c.,2009]
8. <http://google.ru>
9. <http://freebsd.org>

