BÖLÜM 4

SHELLin işləmə prinsipi, terminalda qısa keçidlər, CRON, istifadəçi üzərində əməliyyatlar, Vİ redaktoru, sistem RAİD-ləri,sərt disklərin şifrələnməsi

- SHELL, onun işləmə prinsipi, terminal qisa keçidləri, CRON
- İstifadəçilərin yaradılması, silinməsi və deaktiv edilməsi
- Vİ mətn redaktoru və vim
- Fayl sistemlə praktik işlər
- Disklərin bölünməsi və sistem RAİD-ləri
- Sərt disklərin 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 ardıcıllıqları 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 UNİX/Linux əməliyyat sistemlərində olan vi tekst redaktoru açıqlanır. Diskin bölüşdürülməsi strukturu, fayl sistemin yoxlanılması və proqram təminatı vasitəsilə raidlərin qurulması barədə danışılır. Həmçinin disklərin fərqli üsullarla şifrələnməsi açıqlanır.

SHELL, onun işləmə prinsipi, terminal qısa keçidləri, CRON

Əməliyyat sistemi üzərində istifadə edilən istənilən əmrlər və əməliyyatlar hansısa **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 ardıcıllıqla işə düşür. Aşağıda səliqə ilə həmin SHELL fayllarının təyinatları açıqlanır.

dot.login - Bu faylda olan əmrlər CSH mühitinə giriş edən kimi işə düşəcək.

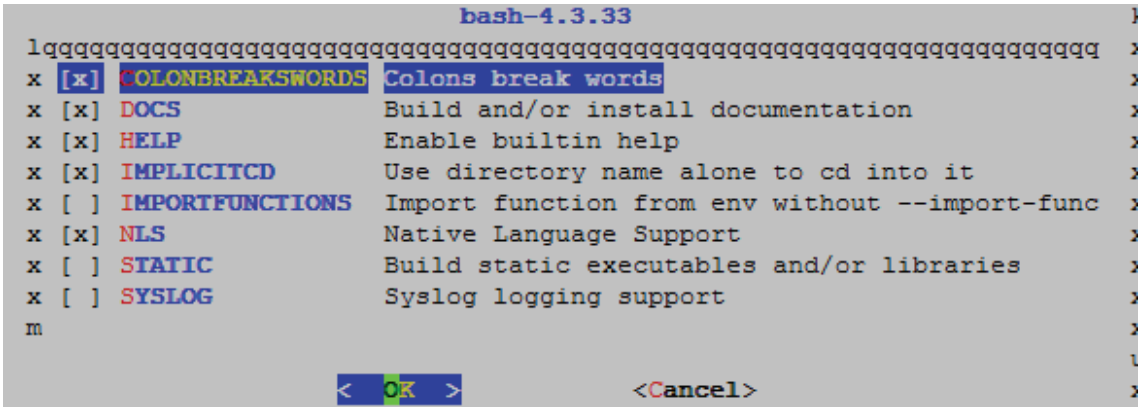
dot.tcshrc yada dot.cshrc - Bu fayllarda olan əmrlər CSH işə düşən kimi işləyəcək.

dot.history	- CSH mühitində işə salınan bütün əmrlər bu faylda saxlanılacaq.
dot.mailrc	- '**mail**' əmri üçün quraşdırmalar bu faylda olur.
dot.mail_aliases	- 'mail' əmri üçün quraşdırmalar bu faylda olur.
dot.rhosts	- 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.
echo $0	- Hansı SHELL mühitini istifadə etdiyimizi çap edir.
echo $$	- İstifadə etdiyimiz SHELL mühitinin PID-ni çap edir.

BASH Shell mühitinin yüklənməsi və quraşdırılması

FreeBSD əməliyyat sisteminin üzərində susmaya görə BASH olmur. Ona görə də, öncə biz onu yükləməliyik.

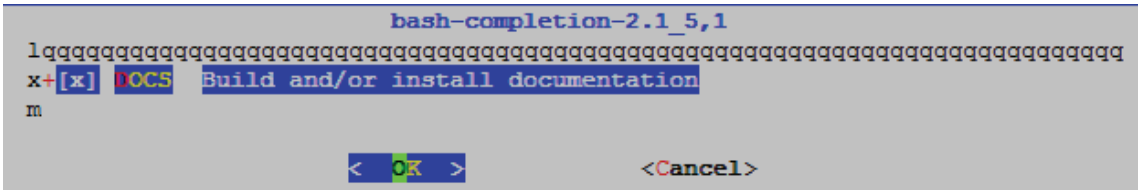
cd /usr/ports/shells/bash	- BASH Shell-in port ünvanına daxil oluruq.
make config	- Lazımi modulları seçirik.



make install clean - Yükləyirik.

cd /usr/ports/shells/bash-completion/ - Komfort üçün əmrlərimizin avtotabulyasiyasını da əlavə edək.

make config - Lazımi modulları seçirik.



make install - Yükləyirik.

> **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.
======================================================================
```

cd /usr/share/skel - Ünvana daxil oluruq ki, BASH üçün skelet faylları yaradaq.

ee əmri susmaya görə FreeBSD əməliyyat sistemində olan mətn redaktorudur. Yəni bu əmri gördükdə təəccüblənməyin. Məsələ burasındadır ki, CSH mühitindən istifadə etdikdə siz ingilis əlifbası ardıcıllığında gələn əmrlərin axtarışını tarixçədə **Up** və **Down** düymələri ilə axtarıb tapa bilərsiniz. Ancaq bu BASH-da susmaya görə olmur.

ee dot.inputrc - Faylı açırıq və içinə aşağıdakı sətirləri əlavə edirik.
"\e[A": history-search-backward - Bash-History Down düyməsi aktiv olsun.
"\e[B": history-search-forward - Bash-History Up düyməsi aktiv olsun.

ee dot.bashrc - Bu fayl onun üçün yarayır ki, istifadəçinin istifadə elədiyi SHELL susmaya görə BASH deyil, və bash çağırılır. Fayla aşağıdakı sətirləri əlavə edirik.

```
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 profilinə 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 user-in 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 saxlanılması üçün istifadə olunacaq.

Beləliklə, sistemə əlavə edilən yeni istifadəçi də **SHELL** olaraq **bash (/usr/local/bin/bash)** təyin edilsə, bu quraşdırmaları mənimsəyəcək.

Bash shell sessiya avtomatik bağlandıqda "**.bash_logout**" faylını işə salır.

Bash işə düşəndə hər istifadəçinin ev qovluğunda olan "**.bash_history**" faylını RAM-a yazır. Bu fayl susmaya görə "**$HISTFILE**" dəyişəninə mənimsədilib. **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**" faylının history həcmini və history faylının həcmini çap edin.

```
/home/namaz/.bash_history 500 500
```

history 5 - History-dən ən son **5** əmri çap edin.
!! - Sonra əmri yerinə yetirin.
!544 - **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 - Tarixçədə sətrə görə axtarış edir.
mail root < /etc/hosts - "root" istifadəçi adına görə /etc/hosts faylının kontentini 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.

echo $EUID	- Hal-hazırkı user üçün Effective User ID-ni çap edəcək.
echo $FCEDIT	- '**fc**' əmri tərəfindən istifadə olunan işləri çap edəcək.
echo $GROUPS	- Hal-hazırkı istifadəçinin üzv olduğu əsas Group-u çap edəcək.
echo $HISTCMD	- Hal-hazırkı əmrin history rəqəmini çap edir.
echo $HISTFILE	- History faylının yerləşdiyi yeri çap edir.
echo $HISTFILESIZE	- History faylının sətir sayına görə ümumi tutum rəqəmi.
echo $HOME	- Hal-hazırkı istifadəçinin ev qovluğu.
echo $HOSTNAME	- Hal-hazırkı maşının adı.
echo $HOSTTYPE	- Hal-hazırkı maşının HOST tipi.
echo $LESSOPEN	- LESS açan bir dəyişəndir.
echo $LINES	- Terminalın hal-hazırkı sətir sayını çap edir.
echo $LOGNAME	- Hal-hazırda daxil olmuş istifadəçinin adını çap edir.
echo $MACHTYPE	- İstifadə olunan maşının OS tipini çap edir (i686 ya X64).
echo $MAIL	- Hal-hazırkı istifadəçinin mailbox qovluğunu çap edir.
echo $MAILCHECK	- Mail yoxlanışı üçün istifadə olunan vaxt intervalını çap edir (60 saniyə susmaya görə).
echo $OLDPWD	- İşlədiyimiz qovluqdan öncəki qovluğu çap edir.
echo $OSTYPE	- Hal-hazırkı OS-un hansı ad altında identifikasiya olmasını çap edir.

echo $PAGER	- man əmrinin səhifələri page eləməsi üçün istifadə olunan əmr (more əmri ilə).
echo $PATH	- Sistemdə istifadə olunan sistem əmrlərinin ünvanlarını çap edir.
echo $PPID	- Hal-hazırda bash tərəfindən işə salınan əmrin Process ID-si.
echo $PRINTER	- Susmaya görə yüklənmiş olan Printeri çap edir. Hansı ki, '**lpr**' və ya '**lpq**' əmrlərindən istifadə olunur.
echo $PROMPT_COMMAND	- İşə salınan əmri adı ilə çap edir və işə salır. Məs: '**PROMPT_COMMAND=ls ; echo $PROMPT_COMMAND**'
echo $PS1	- **SHELL Prompt**-un istifadə etdiyi işarəni çap edir. Məs: **PS1=`date`** CLI-da solda tarix görsənəcək.
echo $PWD	- Hal-hazırda yerləşdiyimiz qovluğu çap edir.
echo $RANDOM	- Bu dəyişəni çağırdıqda, **0**-la **32767** arasında olan təsadüfi bir rəqəm generasiya olacaq.
echo $SECONDS	- **SHELL** mühit işə salınmağa başlayandan istifadə olunan müddət (saniyələrlə).
echo $SHELL	- Hal-hazırkı shell-in tam ünvanını çap edir.
echo $SHELLOPTS	- Aktiv olan **SHELL** opsiyalarını çap edir.
echo $SHLVL	- İstifadə etdiyimiz SHELL iç-içə olaraq neçənci SHELL mühitidir.
echo $TERM	- İstifadə etdiyimiz terminalın tipini çap edir.
echo $TMOUT	- Bu dəyişənə mənimsədilən rəqəm mühitinə təyin

edilən vaxtın bitməsidir. Susmaya görə heç nə yoxdur. Ancaq təyin edilən vaxt saniyələrlə olur. Əgər vaxt bitdisə, 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 namaz - **namaz** adlı istifadəçinin **$SHELL**-ni BASH-la dəyişirik.

Qeyd: BASH mühitində hər hansısa yerinə yetirilən iş bitdikdən sonra statusu yoxlamaq istəsək, '**echo $?**' əmrindən istifadə etməliyik. Əgər cavab **0(sıfır)**-sa **true(doğru)**, əgər **1(bir)**-sə **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 işarəsidir.

Qeyd: '**$**' - adi istifadəçi işarəsidir.

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 yazılanlar 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 qayıdış. (cisco, csh, zsh)

Ctrl + c - Proqrama SIGINT siqnalı ötürür. Adətən, mövcud işi dayandırır. (csh, zsh)

Ctrl + d - Kursorun altında duran simvolu silir. (delete düyməsinin analoqu) (cisco, csh, zsh)

Ctrl + e - Sətrin sonuna keçid. (cisco, csh, zsh)

Ctrl + f - 1 simvol irəli gediş. (cisco, csh, zsh)

Ctrl + k - Kursordan sağa doğru sətrin sonunadək silir. (cisco, csh, zsh)

Ctrl + l - Ekranı silir, clear əmrinin analoqu. (csh, zsh)

Ctrl + r - Tarixçədə axtarış, axtarış təkrarı. (Yəni axtarışı sətirləyir.) Ya da inkremental axtarış. (zsh)

Ctrl + j - Axtarışı dayandırır və tapılan əmrin üstündə dəyişiklik etmək imkanı yaradır. Əgər axtarış yerinə yetirilməyibsə, return əmrinə 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 sətrin əvvəlinədək bütün simvolları silir. (cisco, csh-də, zsh üçün tam sətri silir)

Ctrl + w - Kursordan sola doğru olan sözün əvvəlinədək bütün simvolları silir. (cisco, csh, zsh)

Ctrl + xx	- Sətrin əvvəlinə 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	- Hal-hazırkı işi müvəqqəti dayandırır. (csh, zsh)
Ctrl + x; Ctrl + e	- 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 keçid.(zsh)
Alt + >	- Əmr tarixçəsində ilk əmrə keçid.
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 + /	- Faylın 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 çəkir (cisco, csh, zsh)
Alt + c	- Kursor altındakı simvolu böyük edir, qalanlarını isə sözün sonunadək kiçik. (cisco, csh, zsh)
Alt + d	- Kursor yerləşən yerdən sağa doru olan sözü sonadək silir. (cisco, csh, zsh)
Alt + f	- Kursoru bir söz qabağa aparır (cisco, csh, zsh)
Alt + l	- Kursorun yerləşdiyi simvoldan sonadək bütün simvolları kiçik edir.(cisco, csh, zsh)
Alt + t	- Kursorun altında olan sözlə öncəkinin yerini dəyişir. (zsh)
Alt + u	- Kursorun yerləşdiyi simvoldan sözün sonunadək bütün simvolları böyüdür. (cisco, csh, zsh)

Alt + back-space	- Kursor yerləşdiyi ünvandan sözün əvvəlinədək bütün simvolları silir.(cisco, csh, zsh)

TAB imkanlarına baxaq:

Tab + Tab	- Əmrin artırılması. Əgər boş sətirdə etsək, bütün mümkün ola biləcək əmrlərin siyahısını çap edəcək.
(string)2T	- Bütün mümkün ola biləcək artırmaları çap edəcək.
(dir/)2T	- dir adlı qovluğun bütün alt qovluqlarını çap edəcək.
***2T**	- Gizlilərdən başqa bütün alt qovluqları çap edəcək.
~2T	- **/etc/passwd** faylında olan bütün istifadəçiləri çap edəcək. İstifadəçi adını tamamlamaqla onun ev qovluğuna keçmək olar. Misal üçün, **~namaz/** - namaz adlı istifadəçinin ev qovluğu.
$2T	- Sistem dəyişənləri üçün bütün siyahını tamamlayır.
@2T	- **/etc/hosts** faylında olan host adlarını tamamlayır.
=2T	- Mövcud qovluğu siyahı halına salır. **ls** kimidir.

CRON - System Scheduler

CRON inzibatçının əvəz edilməz alətidir və həmişə inzibatçının işinə yarayacaq. Misal üçün, sistemdə hansısa bir skript yazıla bilər ki, o, ildə, ayda, həftədə, gündə, saatda, dəqiqədə bir dəfə və ya bir neçə dəfə işə düşməlidir. Məhz bu məqamlarda bizim köməyimizə CRON çatır.

Bacarıqları:

1. Qlobal **/etc/crontab** faylını və istifadəçi bazalı **/var/cron/tabs** qovluğunda olan faylllarını işə salır.
2. '**crontab**' – istifadəçilərin **cron** cədvəlini idarə etməyə kömək edir.
3. CRON yazılma strukturunun açıqlanması aşağıdakı kimi olacaq:
   a. '**m**' - dəqiqə - **0-59** - ardıcıllığı ilə bölünür. '***/2**' – hər iki dəqiqdən bir işə salır.
   b. '**h**' - saat - **0-23** ardıcıllığı 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 içində 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 siyasət əsasında təyin edir?
Qeyd: Əgər siyasətinizdə bəzilərindən başqa hər kəsə izin varsa, onda istifadə olunur: '**cron.deny**'
Qeyd: Əgər siyasətinizdə bəzilərindən başqa hər kəsə qadağadırsa, 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 əmrin tam yolunu yazmaq lazımdır. Misal üçün, yazacağınız kodlar BASH-da işə düşməlidirsə, CRON işləməyəcək. Çünki BASH-ın binar faylı /usr/local/bin ünvanında yerləşir və PATH dəyişənində 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 faylın kontrolu birbaşa superuser tərəfindən və hər bir istifadəçinin öz faylına özü tərəfindən olur.

İşimiz:
1. '**/etc/crontab**'-ın susmaya görə olan konfiqlərini açıqlayaq.

Qeyd: '**periodic**' sistemdə vaxtaşırı işə salınacaq işi yerinə yetirir, bu qovluqda: '**/etc/periodic**'

Öz növbəsində '**/etc/periodic**' isə bütün görəcək işini bu faylda olan quraşdırmalardan oxuyur: '**/etc/defaults/periodic.conf**'
Əgər bizə lazım olan hansısa işin periodic işləməsini istəsək, '**/etc/defaults/periodic.conf**' faylını '**/etc/periodic.conf**' faylına 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/namaz/`date +%F`.uptime**'

Qeyd: İşin script-dən oxunmasını istəyirsinizsə, onda əmrləri fayla yazıb, sonra da faylın yolunu crona yazırıq.

```
ee /home/namaz/uptime.sh                    - Fayla aşağıdakı sətirləri əlavə edirik.
#!/bin/sh
uptime | awk '{print $1,$2,$3}' >> /home/namaz/`date +%F`.uptime
#END

chmod 700 /home/namaz/uptime.sh             - Faylı yerinə yetirilən edirik.
```

b. '**/etc/crontab**'-a əlavə edirik, '***/2 * * * * root /home/namaz/uptime.sh**'

3. Scripti istifadəçi adından cronla işə salaq: '**namaz**'
   a. **namaz** istifadəçi adı ilə daxil olub işə salırıq: '**export EDITOR=ee**'
   b. **namaz** istifadəçi konsolunda işə salırıq: '**crontab -e**'
   b1. '**crontab -e -u namaz**' - Eynilə '**root**' istifadəçi adından '**namaz**' istifadəçisinin cronunu təyin edə bilərik.
   c. Sətri əlavə edirik: '***/2 * * * * /home/namaz/uptime.sh**'
   Və root istifadəçi adı ilə '**cd /var/cron/tabs/;cat namaz**' ünvanında yaranan '**namaz**' adında fayla baxırıq.

crontab -e - istifadəçilər üçün şəxsi crontab faylıdır.

Aşağıdakı CRON-la deyirik ki, '**namaz**' adlı istifadəçiyə **1**-ci gündən **5**-ci günədək, saat **8:15**-də '**stats.txt**' faylı mail-lə yollanacaq.

```
15 8 * * Mon,Tue,Wed,Thu,Fri mail namaz < /var/project/stats.txt
```

Bu cron **yanvar, aprel, iyul, oktyabr** aylarının 1-ci günü sistemdə bütün "**doc**" fayllarını axtarır və '**documents.txt**' faylına yazır.

```
* * 1 1,4,7,10 * find /doc | grep .doc$ > /var/sales/documents.txt
```

crontab -eu namaz - '**namaz**' adlı istifadəçi üçün yeni cron əlavə edirik. (Yalnız **root** edə bilər.) '**-u**' istifadəçi, '**-e**' editor rejimə keçid edin.

crontab -l - İstənilən istifadəçi öz cron faylının məzmununa '**-l**' (list) opsiyası ilə baxa bilər.

crontab -l -u namaz - namaz adlı istifadəçinin crontab faylına root istifadəçi adından baxmaq istəyirik.

crontab -r - İstənilən istifadəçi öz crontab faylını 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 planlamanın işə düşmə vaxtında server sönübsə, onda server yenidən işə düşəndə, o, start olmayacaq. Ancaq anacron buna baxır və onu işə salır. Anacron susmaya görə sistemdə olmur, onu yükləmək lazımdır. 'pkg install anacron'

pkg install anacron - anacron-u yükləyirik.

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 - Faylın sonuna aşağıdakı sətri əlavə edirik ki, yenidən yüklənmədən 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 araşdıracayıq.

adduser - İstifadəçiləri əlavə etmək üçün əmrdir. Test üçün bir istifadəçi əlavə edib çıxışına 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: şifrə_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.

rmuser username - username adlı istifadəçinin silinməsi əmrini veririk.

rmuser -y username - İstifadəçini sildikdə bütün suallara yes cavabı verin.

rumuser -v username - İstifadəçini sildikdə daha da detallı informasiya çap edir.

> **Qeyd:** İstifadəçi silindikdə '**rmuser**' bu məlumatları konsola ötürmür. Və hamısı silinir. istifadəçi crontabı, növbədə işləyən bütün cronlar, bütün proseslər silinir, tmp-dən istifadəçiyə aid olan bütün fayllar silinir, istifadəçinin qrupu silinir.

pwd_mkdb /etc/master.passwd - Əmrlə sistemdə olan şifrə bazasını yeniləyirik.

> **Qeyd:** **/etc/login.conf** faylında istifadəçiyə aid olan şifrə və digər susmaya görə olan siyasətləri dəyişmək olur. (Məs: **minpasswordlen=8:**). Dəyişikliyin həmin an işləməsi üçün bu əmri yığırıq. "**cap_mkdb /etc/login.conf**"

id username - Əmr istifadəçinin hansı ID-yə mənsub olduğunu çap edir.

finger -l username - İstifadəçi haqqında tam detallı informasiya çap eləyir.

touch -t 8001031301 messages - **messages** jurnal faylının son dəyişmə tarixini **1980**-ci il **03.10** və saat **13:01**-ə çevirir.

touch -r maillog messages - maillog jurnal faylının tarixini messages faylına yazır.

uname -a - OS versiyasını və platformanı göstərir.

uptime - Serverin dayanmadan işlədiyi müddəti göstərir.

shutdown -p now - Söndürmək üçün. (**-h** tarixçəni saxlayıb söndür) -p power

/etc/shells - Sistemdə istifadə oluna biləcək bütün SHELL mühitləri faylda göstərilir.

Vİ mətn redaktoru və vim

Ümumiyyətlə, FreeBSD əməliyyat sistemində susmaya görə öncədən olan və nisbətən istifadəçiyə rahat mətn redaktoru ee olur. Ancaq istənilən UNIX/Linux platformalı serverdə Vİ mətn redaktoru mövcud olur və sintaksis hər yerdə eynidir. Ona görə də hər bir UNIX/Linux inzibatçının Vİ ilə işləmə qabiliyyətinin olması şərtdir.

Vİ 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.

a - Yazı yazmaq üçün əmrdir.
x - Hər bir simvolu tək-tək silir.
dd - Bütün sətri silir.
"SHIFT+?" yada / - Söz axtarışı üçün **n - next.** Axtardığım növbəti söz
ESC, SHIFT+:, q! - Yadda saxlamadan çıxış (Ardıcıl əmrlər kombinasiyası).
ESC, SHIFT+:, wq! - Yadda saxlayaraq çıxış.
ESC, SHIFT+:, w /tmp/newfile.txt
- Çıxanda faylı başqa adla yadda saxla. (Başqa sessiya açıb yoxlayın.)
Ctrl+f - Bir səhifə aşağı düş.
Ctrl+b - Bir səhifə yuxarı qalx.

Ctrl+d - Yarım səhifə aşağı düş.
Ctrl+u - Yarım səhifə yuxarı qalx.
:1 - Faylın 1-ci sətrinə, yəni əvvəlinə qayıt.
SHIFT+G - Faylın son sətrinə düş.
SHIFT+H - Kursoru olduğumuz səhifənin əvvəlinə qaytar.
SHIFT+M - Kursoru olduğumuz səhifənin ortasına apar.
SHIFT+L - Kursoru olduğumuz səhifənin sonuna apar.
- - Kursoru öncəki sətrin əvvəlinə apar.
SHIFT+$ - Sətrin sonu.
SHIFT+^ - Kursoru sətrin əvvəlinə apar. Ya da '**0**' sıfır. (Eyni işi görür.)
SHIFT+(- Kursoru öncəki abzasın əvvəlinə apar.
SHIFT+) - Kursoru sonrakı abzasın əvvəlinə apar.
SHIFT+{ - Kursoru öncəki paraqrafın əvvəlinə apar.
SHIFT+} - Kursoru sonrakı paraqrafın əvvəlinə apar.
w - Kursoru növbəti sözə apar.
SHIFT+W - Kursoru növbəti sözə apar.
b - Kursoru öncəki sözə apar.
SHIFT+B - Kursoru öncəki sözə apar.
e - Kursoru növbəti sözün sonuna apar.
SHIFT+E - Kursoru növbəti sözün sonuna apar.
h - Bir simvol sola.
k - Bir sətir yuxarı.
j - Bir sətir aşağı.
l - Kursoru **l** hərf sağa çək.
/search_string - Axtarış edilən sözü '**search_string**'-in yerinə yazırıq.
/? - Öncə axtarış edib tapdığımız sözü çap et.
SHIFT+I - Teksti yerləşdiyimiz sətrin əvvəlindən yazmağa başlayacaq.
SHIFT+A - Teksti yerləşdiyimiz sətrin sonundan yazmağa başlayacaq.
SHIFT+O - Yazmaq üçün yerləşdiyimiz sətirdən əvvəldə bir boş sətir aç.
SHIFT+S - Hal-hazırkı sətri sil və yeni boş sətir aç.

vi tekstdə dəyişiklik

i - Hal-hazırkı simvolun önündən yazmağa başla.
SHIFT+I - Teksti hal-hazırkı sətrin əvvəlindən yazmağa başla.
a - Teksti hal-hazırkı simvoldan sonra yazmağa başla.
SHIFT+A - Teksti hal-hazırkı sətrin sonundan yazmağa başla.
o - Bu sətrin altından yeni sətir aç və yazmağa başla.

SHIFT+O - Bu sətrin üstündən yeni sətir aç və yazmağa başla.
s - Hal-hazırkı simvolu sil və yenisini yaz.
SHIFT+O - Hal-hazırkı sətri sil və yenisini yaz.
c? - Replace üçün idi, amma sınaqdan keçmədi.
SHIFT+C - Kursordan sonra sətrin sonunadək sil.
r - Hal-hazırkı simvolu daxil etdiyim simvolla əvəz et.
SHIFT+R - Kursordan sonra hər simvolu daxil edilən simvollara əvəz et.

vi Delete Paste

x - Kursorun altında olan simvolu ardıcıl olaraq sağa doğru olanları hər istifadədə sil.
SHIFT+X - Kursorun altında olan simvolu ardıcıl olaraq sola doğru olanları hər istifadədə sil.
SHIFT+D - Kursordan sonra olanları sətrin sonunadək kəs.
SHIFT+Y - Hal-hazırkı sətri nüsxələ.
p - **CUT** və ya **copy** olmuş sətri kursordan sonra yerləşdir.
SHIFT+P - **CUT** və ya **copy** olan sətri kursordan öndə yerləşdir.
u - Bir əmr öncə etdiyimi geri qaytar. **UNDO** (**VIM**-de "**CTRL+r**")
. - Son istifadə etdiyimiz əmri özünə mənimsə.
J - Yerləşdiyimiz sətrin sonuna növbəti sətri əlavə et.

vi rəqəmlərlə

7cw - Növbəti **7** sözü sil və yenisini yaz. (**8**-ci sözü silməyəcək.)
d5d - Hal-hazırkı sətir də daxil olmaqla növbəti **5** sətri sil.
3p - Öncə kəsdiyimiz mətni kursordan sonra 3 dəfə yerləşdir.
9db - Kursordan öndə olan 9 sözü kəs.
10j - Kursoru 10 sətir aşağı endir.
y2) - Mətni kursordan sonra 2 başlığın sonunadək nüsxələ.
5CTRL+F - 5 səhifə irəli get.
6J - Növbəti 6 sətri yan-yana düz.
vi +25 /tmp/services - 25-ci sətirdən başlayaraq '**/tmp/services**' faylını aç.
vi + /tmp/services - '**/tmp/services**' faylının sonuncu sətrindən başlayaraq faylı aç.
vi +/tty /tmp/services - '**/tmp/services**' faylını ilk "**tty**" sətrini taparaq aç.
vi -r /tmp/services - '**/tmp/services**' faylını crash rejimdə bərpa edərək aç.
view /tmp/services - '**/tmp/services**' faylını yalnız oxuma rejimində aç.
vi protocols rc.conf services - **3** faylı da birdən aç.
:n - Növbəti faylı aç.

:prev	- Öncəki faylı aç.
:wn	- İşlədiyim faylda save edib, ardınca sonrakı faylı aç.
:n!	- Yadda saxlamadan növbəti faylı aç.
:!date	- Sistem əmri date-i vi-dan işə sal.
:!!	- Öncə işə saldığımız sistem əmrini işə sal.
:20	- Faylın 20-ci sətrinə get.
:5,10w abc.txt	- **5**-ci sətirdən **10**-dək olan aralığı '**abc.txt**' faylına yaz.
:e abc.txt	- Bu fayldan çıx və '**abc.txt**' faylını redaktə etməyə başla.
:.r abc.txt	- '**abc.txt**' faylını mövcud fayla yerləşdir.
:s/UNIX/FreeBSD	- Faylda ilk tapılan '**UNIX**' sözünü '**FreeBSD**' sözü ilə əvəz et.
:s/UNIX/FreeBSD/g	- Faylın kursor yerləşən sətrində bütün '**UNIX**' sözlərini '**FreeBSD**' sözü ilə əvəz et.
:%s/UNIX/FreeBSD/g	- Bütün faylda olan '**UNIX**' sözlərini "**FreeBSD**" sözü ilə əvəz et.
:g/FreeBSD /p	- Faylda '**FreeBSD**' sözü olan bütün sətirləri çap et.
:g/UNIX/s//FreeBSD/gp	- Bütün əvvəlində boşluq olan '**UNIX**' sözlərini tapıb, boşluğun yerinə '**FreeBSD**' sözünü yaz.

vi EX commands

:set	- Bütün mümkün ola bilən opsiyaları çap et.
:set	- Susmaya görə olan quraşdırmalardan başqa nələrsə dəyişibsə, onları çap et.
:set number	- Faylda olan sətirləri rəqəmlə.(Söndürmək üçün '**:set nonu**')
:set list	- Sətrin sonlarına '**$**' simvolu və əvvəlinə isə '**^I**' simvolu əlavə et.
:set wm	- VI-in səbəbi WrapMargin.

Ekranı 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.

pkg install vim	- Bu, həddən artıq uzun vaxt alacaq.
vim rc.conf	- Faylı açırıq.
:split services	- Ekranı horizontal olaraq iki yerə böləcək. **BASH** varsa, faylın adını yazdıqda tab işləyir.
:vsplit protocols	- Ekranı vertikal olaraq iki hissəyə bölür. (Bunu oxuya biləcək həddədək eləmək olar.)
'Ctrl+WW'	- Bölünmüş ekranlar arasında keçid.
:q	- Yadda saxlamadan çıxış eynidir.
:wq	- Yadda saxlayaraq çıxış 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 bölümlərinin maksimum həcmini 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, "**bsdlabel**" utiliti isə hər bir hissə həddindən kənar işləmək üçün istifadə olunur. Hər iki alətin 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 quraşdırmalar yadda qalmayacaq.

Qeyd: "**U**" əmri etdiyimiz dəyişiklikləri yalnız o halda geri qaytarır ki, "**w**" write-i öncə edilməmişdir.

U	- write olunmayıbsa, son dəyişiklikləri geri qaytar.
D	- Seçilən hissəni sil.
T	- Partition Slice tipini seçin. 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 diski ilk slice-a bütövlükdə mənimsət (məsləhət görülür ki, yalnız ilk yükləmədə bunu istifadə edək).
Z	- Seçdiyimiz slice-ın 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** diski haqqında ümumi şəkildə informasiya çap et.
fdisk da1	- **da1** diski 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 diski yalnız slice ayırdıqdan sonra görür.

Qeyd: BSDLabel-ı istifadə etmədən öncə 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** - gigabaytla 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 yığmasaq, bütün e d i l ə n dəyişikliklər silinəcək.

W - Etdiyimiz dəyişiklikləri yadda saxla.
Q - Yadda saxlamadan çıxış.

bsdlabel da1s1 - BSD slicenin partition informasiyasını çap edir. (Qeyd: Bu yalnız slisi-i '**bsdlabel -w**' əmri ilə yazdıqdan sonra işləyir, yəni orada ən azı **1** label olmalıdır ki, çap edilsin).

bsdlabel -n -w da1s4 - "**-w**" Yazılmış label-in nəticələrini çap edin, "**-n**" amma write etməyin.

bsdlabel -w da1s4 - "**da1s4**" standart label-i disk1-ə yazırıq.
bsdlabel -e da1s4 - "**da1s4**" label-nı vi editorda açıb FSTYPE_ni ("4.2BSD" etmək üçün)

newfs -N /dev/da1s4 - Formatlananda nəticəni çap et, amma write etmə.

newfs /dev/da1s4 - **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 mənimsədirik.

Bütün quraşdırmaların sistemin yadında saxlanması üçün, **/etc/fstab** faylının içinə aşağıdakı sintaksislə sətri əlavə edirik:

```
/dev/da0                /newdisc        ufs     rw              1       1
```

mount -a - sistemi yenidən yüklənmə etmədən /etc/fstab faylının içində yazılan bütün quraşdırmaları işə salır.

Qeyd: Susmaya görə FreeBSD sistemi '**/**' slice-i **read/write** rejimində istifadə edir. Dəyişmək istəsək, **/etc/rc.conf** faylına **root_rw_mount="NO"** sətrini əlavə etmək lazımdır.

FreeBSD ext2

FreeBSD əməliyyat sisteminə **ext2** fayl sistemi mount edib istifadə etmək mümkündür. Bunun üçün xüsusi proqram təminatı yüklənməlidir.

```
cd /usr/ports/sysutils/e2fsprogs
```
- **ext2** FS-in dəstəklənməsi üçün bu paketi yükləyirik.

```
make install all
```

```
mkfs.ext2 -v /dev/da1s1
```
- **da1** diskinin **1**-ci slice-nda "**ext2**" formatında format edin. "**-v**" verbose.

```
mkfs.ext2 -v -c /dev/da1s1
```
- "**/dev/da1s1**" slice-nda "**-c**" bad blokları yoxlanış edin və "**-v**" verbose et.

```
mount -t ext2fs /dev/da1s1 /disk
```
- "**/disk**" qovluğuna "**/dev/da1s1**" slice-ni "**-t**" (tipi) '**ext2fs**'-lə mount et.

```
df -HT
```
- Diskləri "**-H**" (İnsan tərəfindən oxunula bilən formatda) Megabayt və Gigabayt tipli göstər, "**-T**" file sistem tiplərini də göstər.

```
tune2fs -l /dev/da1s1  | less
```
- "**/dev/da1s1**" diski haqda bütün atributları çap et, "**-l**" diskdə qeyd olunan susmaya görə olan dəyişənləri çap et.

```
dumpe2fs -h /dev/da1s1 | less
```
- "**/dev/da1s1**" diski haqda bütün atributları çap et, "**-h**" ancaq superblock haqqında informasiyanı çap et.

```
tune2fs -c 31 /dev/da1s1
```
- "**/dev/da1s1**" FS-in sərt yoxlanışı maksimum "**-c**" 31 dəfə olsun.

```
tune2fs -c -1 /dev/da1s1
```
- "**-c**" "**-1**" FS-in sərt yoxlanışını, ümumiyyətlə, söndür.

tune2fs -i 10 /dev/da1s1 - "**da1s1**" FS-nə yoxlanışı "**-i**" 10 gündən sonra et.

tune2fs -i 1d /dev/da1s1 - "**da1s1**" FS-nə yoxlanışı "**-i**" 1 gündən sonra et.

tune2fs -i 3w /dev/da1s1 - "**da1s1**" FS-nə yoxlanışı "**-i**" 3 həftədən sonra et.

tune2fs -i 6m /dev/da1s1 - "**da1s1**" FS-nə yoxlanışı "**-i**" 6 aydan sonra et.

tune2fs -i 0 /dev/da1s1 - 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** yükləməlisiniz və **/boot/loader.conf** faylına **fusefs_load="YES"** sətrini əlavə etməlisiniz ki, sistem yenidən yüklənməsində avtomatik modul yüklənsin.

mkfs.ext4 /dev/da1 - **da1** diskimizə **ext4** fayl sistem yazırıq.
ext4fuse /dev/da1 /mnt/ - Sonra **da /mnt** qovluğuna mount edirik.

Qeyd: **e2fsprogs** 3-cü tərəf proqram yükləndikdə, həmçinin fayllara atributların təyin edilməsi və onlara baxış keçirilməsi üçün **chattr** və **lsattr** adlı binary fayllar da yaranır.

File atributlarının dəyişdirilməsi

FreeBSD əməliyyat sistemində ext2 və ext3 file sistemi yaratmaq üçün xüsusi 3-cü tərəf "**e2fsprogs**" proqram yükləndikdə, eynilə fayllar üzərində atributların təyinatı funksionallığı da yaranır. Bu alt başlıq həmin atributların imkanlarını açıqlayır.

lsattr -aR /tmp/ | less - Rekursiv olaraq linux atributlarını çap edir.

Atributlar:

a - Ancaq əlavə eləmək;
c - Sıxılmış;
d - Dump olmasın;
i - Dəyişməz, bu opsiya ilə təyin olunan faylı silmək, ad dəyişmək və ya link etmək olmaz;
j - Verilənlərin jurnallanması;

s - Təhlükəsiz silinmə;
t - Aşağı qarışıqlıq olmasın;
u - Silinməzdir;
A - Yenilənmə vaxtı yoxdur;
D - Qovluqla sinxron şəkildə yenilənmə;
S - Sinxron yenilənmə;
T - Direktiv iyerarxiyanın başı;

Atribut təyin etmək ucun "+", 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 müddətdə 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	- "**/dev/da0s1e**" alətini ufs yoxlanış edir.
fsck_ufs -B /dev/da0s1f	- '**-B**' arxa fonda file sistemi yoxlanış et.
fsck_ufs -B -f da0s1e	- "**fsck_ufs**" özü avtomatik '**da0s1e**' ünvanını tapıb "**-B**" arxa fonda "**-f**" force rejimdə yoxlanış edəcək, hətta təmiz olsa belə.
fsck_ufs -d /dev/da0s1a	- "**-d**" debugging rejimdən başqa olan bütün əmrləri çap et.
fsck_ufs -y /dev/das01a	- Əgər hansısa faylı düzəldə bilirsə, bütün suallara "**yes**" cavabı verin.
fsck_msdosfs /dev/dals2	- "**msdosfs**" file 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 və "**-l**" daxili file sistemimizi çap edin. Yəni əgər biz nfs və ya SMBFS-lə share mount eləmişiksə, onlar çap edilməyəcək.

df -ht ufs	- "**-h**" human readable və "**-t**" tipi '**ufs**' olan file sistemi çap et.
df -h /home/	- "**-h**" human readable-da yalnız "**home**" slice-nı çap et.
du -h /home/	- "**-h**" human readable-da "**/home**" silce-nın istifadə etdiyi disk tutumunu açıqlayaraq, çap et.
du -sh /home/file	- "**-h**" human readable-da və "**-s**" seçilmiş fayl üçün "**/home/file**" faylının həcmini çap et.
du -sch /home /var	- '**-h**' human readable-da "**/home**" və "**/var**" slice-nin ayrı həcmi və "**-c**" ikisi birlikdə olan ümumi həcm.
find /var -maxdepth 1 -type d -exec du -sh {} \;	- 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 RAİD-ləri

Disklərin bölünməsi

FreeBSD ilk yüklənməsində sistemin işlətdiyi 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 öncə test üçün yeni diski sistemə əlavə edəcəyik və onu format edib sistemimizə mount edəcəyik.

Bir diski 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.

bsdlabel -B -w da4s1 auto - "**-B**" boot code "**/boot/boot**" ünvanından oxunub **da1s1** diskinə yazılacaq. Və disk düzülüşünü "**auto**"-dan alın, yəni "disktab"-dan, hansı ki, simvolları "**/etc/disktab**" faylından alır.

bsdlabel -e da4s1 - da4s1 adlı diskdə FSTYPE-i 4.2BSD təyin edirik. Nəticə aşağıdakı kimi olacaq.

```
# /dev/da1s1:
8 partitions:
#          size     offset    fstype   [fsize 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 disk yaradırıq.

newfs /dev/da4s1a - Ardıcıllığı bütün slice-lar üçün edirik.

mount /dev/da4s1a /1 - Slicelar-ı mount edirik.

Sonda **/etc/fstab** faylına aşağıdakı sətri əlavə edirik ki, sistemin yenidən yüklənməsində işləsin.

/dev/da2s1a /1 ufs rw 0 0

Sistem RAİD-ləri

Bu alt başlığımızda əməliyyat sisteminin imkanları ilə proqram səviyyəsində RAID qurub test edəcəyik. Başlamazdan öncə 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)

kldload geom_stripe	- **geom_stripe** modulunu sistemdən çağırırıq. Susmaya bu modul yüklənmiş olmur. Ya kernel-dən çağırmalı, ya da KERNEL-in daxilində kompilyasiya etməlisiniz.
mkdir /raid0	- **/raid0** qovluğu yaradırıq mount point
gstripe label -v st0 /dev/da1 /dev/da2	- **gstripe** əmri **st0** adlı disk çağırır və bu diskə 2 ədəd da1 və da2 diskinin həcmini tikir.
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**-ı 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 - Virtual diski startupa əlavə edirik.
Ctrl+u
Ctrl+e - Enter sıxaraq yeni sətrə keçid alıb aşağıdakı sətri əlavə edirik və yadda saxlayaraq çıxırıq.
/dev/stripe/st0 /raid0 ufs rw 2 2

> **Qeyd:** Eynilə logic "st.." disklərini belə mirror(güzgü)etmək olar, ancaq mount olunmadan öncə.

RAID1 quraşdırılması

sysctl kern.geom.debugflags=17 - debugging rejimi aktiv edirik.

Həmçinin kernel dəyişənlərinin sistemin yenidən yüklənməsindən sonra avtomatik olaraq işə düşməsini istəyiriksə, **/etc/sysctl.conf** faylının sonuna aşağıdakı sətri əlavə edirik.
kern.geom.debugflags=17

kldload geom_mirror - Disklər üçün güzgü işini görən modulu çağırırıq.

gmirror load - Öncəki əmrlə eyni işi görür. Mirror modulunu yükləyirik (kldload-la eynidir).

/boot/loader.conf - Modulun sistem yenidən yüklənməsindən 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.

gmirror insert gm0 /dev/da4 - **gmirror gm0** diskinə güzgülənmə üçün 2-ci diski əlavə edir.

gmirror configure -a gm0 - '**gm0**' alətinin bütün disklərimiz üçün güzgü rolu oynayacağını elan edirik.

gmirror status - Güzgülənmiş diskləri çap edəcək.

newfs /dev/mirror/gm0 - **gm0** güzgü diskini formatlarıyıq.

> **Qeyd:** Mirror-a əlavə etdiyimiz disklərin həcmi eyni olmalıdır.

mkdir /guzgu - **/güzgü** adlı qovluq yaradırıq mount point.

mount /dev/mirror/gm0 /guzgu - **gm0** güzgü diskini '**/guzgu**' ünvanına bağlayırıq.

> **Qeyd:** Əgər sistem yenidən yüklənmədən sonra qalxmasa və bu simvol çıxsa, -> "**mountroot>**" düymə ilə reboot olub "**6**"-cı seçimi edirik. Və aşağıdakı əmrləri daxil edirik.

load geom_mirror - Modulu yükləyir.
boot - Sistemi yükləyir.

Əgər diskimizin biri sıradan çıxsa, əvvəl adını dəqiqləşdiririk ki, hansıdır. Sonra çıxardırıq.

gmirror forget gm0 - Mirror-u passiv edib diski çıxarırıq.
gmirror insert gm0 /dev/da1 - Mirror-a yeni disk əlavə edirik.
gmirror list - Tərkibində olan diskləri və həcmlərini göstərir.

gstat - Disklərin saniyələrlə giriş-çıxışını həcmlə göstərmək üçün utilitdir.

cd /guzgu - Test üçün ünvana girib, **gstat** əmrini başqa seansda daxil edərək buraya informasiya yazıb disklərin **input/output**-na baxaq.

jot 10000000 > file - Müəyyən bir informasiya yazdıqda eyni vaxtda '**gstat**' əmrinin çıxışına digər seansda baxın.

cat '/dev/zero' > /guzgu/file - Həmçinin bu əmr ilə də fayla yazıb test edə bilərsiniz.

Sonda isə '**/guzgu**' diskini umount edib, disklər haqqında informasiyanı '**/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 - **geom_stripe** modulunu çağırırıq
kldload geom_mirror - Disklər üçün güzgü işini görən modulu çağırırıq.

gstripe label -v ss0 /dev/da1 /dev/da2 - gstripe **ss0** adlı disk çağırır və bu diskə **2** ədəd **da1** və **da2** diskinin həcmini tikir.

gstripe label -v ss1 /dev/da3 /dev/da4 - gstripe **ss1** adlı disk çağırır və bu diskə **2** ədəd da3 və da4 diskinin həcmini tikir.

gmirror label -vnb round-robin gm0 /dev/stripe/ss0
- Güzgü modulu **round-robin** alqoritmi ilə **gm0** güzgü diskinə **ss0** logical diski əlavə edir.

gmirror insert gm0 /dev/stripe/ss1 - Güzgü modulu **round-robin** alqoritmi ilə **gm0** güzgü diskinə **ss1** logical diski əlavə edir.

-b - Balans.
-n - Komponentlərin avtomatik sinxronizasiyasını dayandırır.
-v - Görülən işi consola yollayın.

gmirror configure -a gm0 - **gm0** diski bütün disklərimiz üçün güzgü rolunu oynayacaq.

newfs /dev/mirror/gm0 - **gm0** güzgü diskini formatlarıyıq.

mkdir /raid10 - Mount edəcəyimiz qovluğu yaradırıq.

mount /dev/mirror/gm0 /raid10 - gm0-ı raid10 ünvanına mount edirik.

/etc/fstab - Disklərimizin startup faylına aşağıdakı sətri əlavə edirik ki, güzgü diskimiz sistemin yenidən yüklənməsində avtomatik işə düşsün.

```
/dev/mirror/gm0          /raid10          ufs      rw                2        2
```

CCDCONFIG vasitəsilə RAID konfiqurasiyası

ccdconfig - birləşdirilmiş disklər driver üçün konfiqurasiya utilitidir.

CCD vasitəsilə RAID0 yaradaq.

ccdconfig ccd0 32 0 /dev/da1 /dev/da2 /dev/da3 /dev/da4
- RAID util, **/dev/ccd0** virtual alət altına **4** ədəd diski tikir 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 yenidən yüklənməsindən 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 yenidən yüklənməsindən sonra avtomatik yüklənsin.

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 config-ləri '**rc**' burdan oxuyur. Əgər aşağıdakı sətir faylda olmazsa, sistemin yenidənyüklənməsində disklər mount olmayacaq və tək istifadəçili rejimə keçid alacaqsınız.

GVINUM - Logical Volume Manager control program

Adından göründüyü kimi, logik disklərin idarə edilməsi üçün proqram təminatıdır. Dəstəklədiyi RAID siyahısı aşağıda göstərilir:

RAID0 - Disk bölüşdürülməsi
RAID1 - Güzgülənmə
RAID5 - Cütlüklə növbələşmə
RAID10 - Bölüşdürmə və Güzgülənmə

/dev/gvinum - Obyektlər bu ünvanda saxlanılır. Disk adları və iyerarxiyaları.

Verilənlər bazasının konfiqurasiya faylları hər bir 'vinum' diskdə saxlanılır.

newfs /dev/gvinum/VOL_NAME - Bu strukturu istifadə edərək, diskimizə File System yazırıq.

Qeyd: GVINUM modulu susmaya görə kernelde yüklənmiş olmur. Yəni əgər biz '**gvinum**' diskini '**/etc/fstab**' faylına əlavə etsək, o, sistemin yenidən yüklənməsindən sonra işləməyəcək. İşləməsi üçün aşağıdakıları edirik:

/boot/locader.conf - Sistemin Modul StartUP faylına **GEOM_VİNUM** modulunu əlavə edirik.

geom_vinum_load="YES" - gVinum modulu yüklənsin.

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/da1
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 diski UFS file system-ə format edirik.

mkdir /volconcat - **volconcat** adlı qovluq yaradırıq ki, GVINUM formatlınmış 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 diski '**umount**' eləmək lazımdır.

Qeyd: gvinum konfiqurasiyasını 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 diski mount etmək üçün qovluq yaradırıq.

mount /dev/gvinum/raid5 /raid5/ - **raid5** adlı virtual diskimizi yaratdığımız **/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 içinə əlavə etmək lazımdır.

Kernelin içində istifadə eləmək istəsəniz, aşağıdakı sətri kernel faylına əlavə edib kompilyasiya etməlisiniz:

```
options GEOM_BDE
```

Modulla çağırmaq istəsəniz, aşağıdakı əmri işə salmalısınız:

```
kldload geom_bde
```

Həmçinin StartUP-da işləməsi üçün /boot/loader.conf faylına 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 həmçinin **/home** və **/var/mail** qovluqlarını da şifrələyə bilərsiniz, ancaq bunun ardıcıllıq 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 müdaxilə ediləmədən GBDE deşifrə edə bilməyəcək. Hər bir şifrələnmiş disk hissəsi ayrı blok fayldan istifadə edir.

gbde init /dev/da1 -i -L /etc/gbde/da1.lock - gbde disklə işləməyə başlamazdan öncə onu inisializasiya etmək lazımdır. İnisializasiya 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 çıxdıqda 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/da1.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, çünki bu faylın nüsxəsi heç kəsin əlinə keçməli deyil. Fayl itərsə, informasiya bərpası da çox çətin olacaq. Hətta rəsmi yazarlar özləri belə bunu etmirlər.

gbde attach /dev/da1 -l /etc/gbde/da1.lock - Şifrələnmiş diski sistemə qoşaq. Bir az öncə yaratdığımız şifrəni daxil edirik. **ls /dev/da1*** əmrini daxil etsəniz, **/dev/device_name.bde** quruluşlu yeni disk yaranacaq.

mkdir /gbdecrypt - Şifrələnmiş diskimizi mount edəcəyimiz qovluğu yaradırıq.

newfs -U -O2 /dev/da1.bde - Diskimizə fayl sistem yazırıq. **Qeyd:** Fayl sistem mütləq **.bde** genişlənməli diskə yazılmalıdır.

mount /dev/dal.bde /gbdecrypt/ - Diskimizi mount edirik.

df -H | grep bde - Şifrələnmiş diskimizə baxaq
/dev/dal.bde 5.1G 8.2k 4.7G 0% /gbdecrypt

fsck -p -t ffs /dev/dal.bde - Diskimiz **/etc/fstab** 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 /etc/rc.conf faylına əlavə etmək lazımdır. Ancaq yenə də sistem qalxdıqda şifrəni əllə konsoldan daxil etməlisiniz.

```
gbde_autoattach_all="YES"
gbde_devices="dal"
gbde_lockdir="/etc/gbde"
```

gbde detach /dev/dal - Qorunan diskin ayrılması üçün əmri daxil etməniz kifayətdir.

GBDE sektorların tərkibini 128 bitlik AES ilə şifrələyir. Hər bir sektor fərqli AES açarla şifrələnir.

GELİ vasitəsilə disklərin şifrələnməsi

GELİ-də olan üstün xüsusiyyətlər aşağıdakılardır:

- **CRYPTO(9)** infrastrukturu istifadə edir və avadanlıq səviyyəsində şifrələnmə görən kimi onu istifadə edir.
- Fərqli şifrələnmə alqoritmləri dəstəkləyir. Hal-hazırda **AES**, **Blowfish** və **3DES**.
- Kök diskin şifrələnməsini dəstəkləyir, ancaq sistem qalxanda şifrəni daxil etmək lazımdır.
- İki bir-birindən asılı olmayan şifrələnmə açarını dəstəkləyir(misal üçün, əsas açar və şirkətin açarı).
- Sektor-Sektor şifrələnməsinə 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 şifrələnməsi imkanı.

İşə salmaq üçün kerneli aşağıdakı sətirlərlə kompilyasiya etmək lazımdır:

```
options GEOM_ELI
device crypto
```

Ya da **/boot/loader.conf** faylına aşağıdakı sətri əlavə etmək lazımdır:

```
geom_eli_load="YES"
```

kldload geom_eli - Həmçinin konsoldan çağıraq ki, işimizi davam etdirə bilək.

Aşağıda göstərilən proses açar faylın generasiyasını açıqlayır, hansı ki, şifrələyici üçün əsas açarın hissəsi olacaq. Diskimizi **/gelicrypt** qovluğuna mount edəcəyik. Bu açar faylın köməkliyi ilə təsadüfi verilənlərin yığımı alınır, hansı ki, onunla əsas açar şifrələnir. Bundan əlavə, o, kod sözlə şifrələnəcək. Provayder sektoru həcmi **4kB** olacaq.

Əsas açar, kod sözlə qorunacaq. Açar fayl üçün verilənlər isə **/dev/random** virtual alətindən 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: şifrə
Reenter new passphrase: şifrə_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 alətinin adı **/dev/da2.eli** olacaq.

```
Enter passphrase: şifrə
```

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                          - şifrələnmiş diskimizə baxırıq.
/dev/da2.eli    5.2G    8.2k    4.8G      0%    /gelicrypt

umount /gelicrypt                         - Diskimizi sistemdən ayırırıq.
geli detach da2.eli                       - 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 şifrələmək istəsəniz, sadəcə **/etc/fstab** faylında, swap bölünmüş diskin sonuna şifrələmə tipi olaraq **.eli** yada **.bde** yazmanız kifayətdir.

Aşağıdakı sətirlərdəki kimi:

```
/dev/da0p3.eli  none            swap    sw      0       0
/dev/da0p3.bde  none            swap    sw      0       0
```

Sistem yenidən yüklənməsindən sonra **swapinfo -h** əmrini daxil etdikdə aşağıdakı nəticəni əldə etmiş olacaqsınız:

```
Device          1K-blocks     Used    Avail Capacity
/dev/da0p3.bde    1016752       0B     993M     0%
```