

Rozdział 2 - Essential tools

Przekierowywanie > to po prostu pcha całość do pliku. >> appenduje. Redirecty mogą też być łączone i są wtedy ewaluowane od lewej do prawej.

```
SORT < JAKIS_PLIK > DRUGI_PLIK_NA_OUTPUT
```

To się powinno czytać tak:

```
(SORT < JAKIS_PLIK) > DRUGI_PLIK_NA_OUTPUT
```

By przekierować **ERROR** i **OUTPUT** do pliku to trzeba tak: **COMMAND > output 2>&1** - to tak naprawdę co robi to bierze **STDERR** i przekierowuje do **STDOUT**. A że wcześniej zwykły **OUT** (przez operator >) wrzucamy do pliku no to wszystko ładuje w pliku.

Przekierowanie od pipe różni się tym, że przekierowanie wali lub czyta z pliku. Z kolei pipe robi tak, że bierze **STDOUT** z jednej komendy i pcha na **STDIN** drugiej.

Internal commands to: **echo, printf, read, cd, pwd, cp, pushd, popd, dirs.**

To find out if command is builtin use **TYPE COMMAND**

to check out from where the command is taken use **WHICH** command.

HISTORY is a command to show history of commands. **Ctrl+R** enables searching FOR COMMAND ONLY and again pressing **Ctrl+R** searches next. **!number** executes command with given number. **!text** executes command starting with given text (NO CONFIRMATION)!!

History -c clears memory. **History -w** clears the contents of **.bash_history**

The **~/.bash_profile** would be used once, at login. The **~/.bashrc** script is read every time a shell is started. This is analogous to **/.cshrc** for C Shell.

One consequence is that stuff in **~/.bashrc** should be as lightweight (minimal) as possible to reduce the overhead when starting a non-login shell.

/ETC/ISSUE is being shown before user is logged in. **/ETC/MOTD** is shown after.

APROPOS is the same like **man -k** (bardzo ogolne szukanie po keywordach, glownie jak sie nie pamietala komendy). **Man -f COMMAND** shows short description.

Man categories that are important are:

- 1: Executable programs or shell commands
- 5: File formats and conventions
- 8: System administration commands

Updating man database is command **MANDB** but run as **ROOT**. If run by normal user it starts but fails at the first attempt to clean files.

/usr/share/doc contains larger descriptions of systems (eg. **bind/syslog**) where **pinfo** allows to browse man pages with hyperlinks.

Rozdział 3 - Mounting of directories

mount command gives overview, it read **/proc/mounts file**. It shows all.

findmnt pokazuje to samo co **mount** ale w fajnej drzewiastej formie i jest bardzo czytelne

df -Th pokazuje o wiele sensowniejszy output i skupia się na dyskach i filesystemach (flaga **T**)

INODY to identyfikatory miejsca na dysku gdzie sa skladowane dane. I kazdy plik jest tak naprawde linkiem do INODE. Kiedy usuwa sie ostatni link (czyli 'plik') to po prostu ten inode jest czyszczony. Oznacza to jednak, ze kiedy np. plik jest otwarty do edycji gdzieś, a my go usuwamy, to dalej mozna robic edycje tego pliku, ale po zamknieciu edytora system sie kapnie, ze linki polecialy i wyczysci go. **VIM** przy edycji plikow tworzy w ogole nowy plik i po wyjściu z niego podmienia INODY w pliku.

Listing files

LS opcje:

- **l** newline i wszystkie info
- **a** ukryte
- **t** sortuje po modification date
- **r** (male r) reversuje order
- **R** (duze R) wchodzi rekursywnie w podkatalogi
- **i** pokazuje identyfikator **INODE**

Kopiowanie i podobne

Kopiowanie rekursywne to flaga **-R**

flaga **-a** kopiuje rowniez permissiony

kopiowanie calego folderu i jego sunfolderow to **cp /KATALOG/**

kopiowanie wszystkich plikow (ukrytych i normalnych) to **CP -A /katalog/**. (kropka na koncu)

Usuwanie domyslnie propmtuje po zgode - flaga **-f** usuwa ten wymog

Linki

* **HARD:**

- ten sam device
- nie da sie ich zrobic do folderow
- jak usuniesz ostatni alias to i plik sie usuwa
- **RHEL 7** Musisz byc ownerem source!!!

* **SOFT (Symbolic)**

- linkowac moga wszedzie
- moga do folderow
- jak sie target usunie to symbolic link jest bezuzyteczny

LN DOCELOWY_PLIK TWORZONY_LINK

Kompresja

TAR OPCJE (-CYFRA - poziom kompresji, im wiecej tym mocniej) DOCELOWY_PLIK CO_ARCHIWIZUJEMY

- c create
- x extract (flaga -C po nazwie archiwum pozwala wskazac target folder). Podajac jako drugi paramer pelna nazwe pliku w archiwum extractuje sie tylko go
- f nazwa pliku (musi byc ostatnia flaga)
- v to verbose
- r dodanie pliku (plik skompresowany | co dodac)
- u updatuje istniejacy w archiwum plik
- z kompresja GZIPEm podczas tworzenia only
- j kompreska BZIP2 podczas tworzenia only

Z kolei przy GZIP lub BZIP2 to dajemy KOMENDA CO_DO_KOMPRESJI
Rozpakowywanie to flaga **-D** (zakladam, ze od **decompress**)

Dla kompatybilnosci z Windowsem jest tez **ZIP** i **UNZIP** (nie sa domyslnie zainstalowane!!!).
Uzycie jest dosc proste:

ZIP -R (recursive) FILE_NAME /CO_DO_ZROBIENIA
UNZIP NAZWA_PLIKU

Szukanie plikow

Służy do tego oczywiście komenda **FIND**. Użycie po kolei:

FIND GDZIE_SZUKAC -name REGEXP_Z_NAZWA_PLIKU

Dodatkowe flagi czy przełączniki:

-type f/d/l czy szukać plików, folderów czy symlinków

-user NAME właścicielem jest user (nazwa usera lub UID)

Istnieje też komenda **LOCATE**, która jest o niebo szybsza niż **FIND** ale to dlatego, że nie jedzie po dysku tylko korzysta z bazy danych, która jest raz dziennie odświeżana

(**/ETC/CRON.DAILY/MLOCATE**). Można wymusić jej odświeżenie przez **UPDATEDB** - domyślnie plik z bazą danych leży w **/var/lib/mlocate/mlocate.db**

ROZDZIAL 4 - TEXT FILES

LESS

G by isc do konca pliku (to samo co w VIMie)
/costam by szukac (kolejne szukanie to klawisz n)
?costam by szukac do tylu

* **CAT** zrzuca wszystko, by zrobic reverse **CAT** komenda to **TAC**
* Domyslnie **HEAD** i **TAIL** pokazuja 10 rekordow. Ogranicznik to flaga **-n**. No i niesmiertelna flaga **-f** ale glownie w tailu bo w headzie nie ma sensu.

* Fajna komenda jest **CUT**, ktora filtruje specyficzne pola wpierw je tnac po delimiterze. Zatem:
-d to flaga gdzie piszemy delimiter
-f to flaga z podaniem kolumny ktora chcemy dostac (liczymy od 1!!!)
Zatem by wyciagnac nazyw userow z /etc/passwd robimy **cut -d : -f 1 /etc/passwd**

* do sortowania uzywamy o dziwo komendy **SORT**. Domyslnie ona sortuje alfabetycznie.
- flaga **-n** bedzie sortowac numerycznie
-**r** reversuje sortowanie
-**t** tez pozwala dac templejt do sortowania i laczy sie to z **-kX** gdzie X to numer kolumny (tez startuje od 1!!!)

sort -k3 -t : /etc/passwd → to sortuje ten plik po 3 kolumnie, rozdzielonych :

* liczenie czegokolwiek to komenda **WC** (word count) - linie slowa znaki

* **GREP** ma mase ciekawych opcji (ogolnie **GREP** jest case sensitve!!!):
-i pomija upper/lower
-v pokazuje to co NIE ZAWIERA danego wyrazenia
-r szuka rekursywnie
-w szuka stringa tylko kiedy jest oddzielnym slowem (czyli w : 'testchlebik' szukajac 'chlebik' bez tej flagi to go znajdzie, ale z flaga juz nie)
-e uzywane jak sie szuka linii dla kilku patternow: **grep -e 'nologin' -e 'root' /etc/passwd**
-A lub -B pokazuje ilos linii PO (After) i PRZED (Before) dopasowaniu: **grep chlebik /etc/passwd**

GREP OPCJE CZEGO_SZUKAMY GDZIE_SZUKAMY

ROZDZIAŁ 5 - LOGOWANIE DO SYSTEMU

Generalnie istnieją wirtualne terminale - otwierane **Alt+F1-F6** lub za pomocą polecenia **chvt X** gdzie X to numer terminala.

Jeśli się to robi z graficznego to często **Alt+F1** czy coś tam jest zajęte. Wtedy używaj **Alt+Ctrl+F2** i tak dalej

tty1-tty6 to wirtualne terminale. Z kolei **pts1-6** to 'pseudo-terminale', które mogą być odpalane z GUI.

Komenda **ssh** ma kilka flag:

- **I** Do wyspecyfikowania nazwy usera, ale najczęściej idzie **user@host**
- **i** Podaje lokalizację pliku z kluczem prywatnym który ma być użyty (normalnie bierze **~/.ssh/id_rsa**)
- **F** Podajemy plik konfiguracyjny dla połączenia (domyślnie **~/.ssh/config**)

SSH ma plik konfiguracyjny dla każdego usera oddzielny, ale istnieje też ogólny dla wszystkich w **/ETC/SSH/SSH_CONFIG**

ssh -X to wywołanie ze strony klienta do docelowego serwera z prośbą o uruchamianie aplikacji GUI. By to jednak działało trzeba jeszcze jako ROOT dać flagę w konfigu demona - **/etc/ssh/ssh_config** i ustawić **Forward X11 yes**

Komenda **WHO** lub **W** pokazuje wszystkich obecnie zalogowanych do systemu

SSH-KEYGEN generuje parę kluczy, a z kolei **SSH-COPY-ID** kopiuje publiczny na remote server

scp /etc/hosts server2:/tmp wysyła plik hosts z obecnej maszyny do server2/tmp
scp root@server2:/etc/passwd ~ login jako root do server2 i ściąganie stamtąd pliku z hasłami do katalogu domowego
scp -r server2:/etc/ /tmp podobnie tylko nie jako root i ciągniemy cały folder rekursywnie

SCP używa **-P** (wielka litera) do wyspecyfikowania portu

ROZDZIAŁ 6 - USER AND GROUP MANAGEMENT

ROOT ma dostęp do wszystkiego. Nie ma czegoś takiego jak 'ROOT bez praw dostępu' czy coś takiego.

Komenda by pokazać info o userze to **ID USERNAME** (lub samo **ID** by pokazać siebie)

Inna komenda jest **GETENT (database) USERNAME** co pozwala pociągnąć informacje np. z LDAPa. Lista **database** jest dość długa.

SU domyślnie otwiera subshell dla **ROOTA**.

Oczywiście to **SU** - to nam otworzy ROOTA, jak chcemy dostać się jako inny user to **SU - USERNAME**.

Istnieją 2 typy shella - **LOGIN SHELL** i **INTERACTIVE SHELL**. Różnią się przede wszystkim plikami, które się uruchamia (konfigi z folderu domowego **.bash_**). Dla **INTERACTIVE** to jest **bashrc** a dla **LOGIN** **.bash_profile**. Więc jak się chce mieć przetworzone skrypty dla obu shelli to jedziemy **SU -**. Dla każdej **LOGIN SHELL** przy wylogowaniu jest też procesowany **.bash_logout**.

Ciekawa opcja z **SU** jest JEDNORAZOWE wykonanie jakiejś komendy jako inny user, a nie otwieranie shella. Przykład to: **SU -C 'komenda do wykonania' NAZWA_USERA** (dla ROOTa można to pominąć)

Komenda do modyfikowania usera to **USERMOD**. Najpopularniejszym przykładem jest dodawanie usera do grupy - **USERMOD -aG GROUPNAME USER**. Grupa **WHEEL** na RHEL/CENTOS to domyślna grupa sudoersów. Flagi w użyciu to też **L** oraz **U** (lock i unlock), **c** do dodawania komentarza do **/etc/passwd**. Na sam koniec jest **-d** do zmiany folderu domowego usera (najczęściej z flagą **-m** która przekopiuje zawartość istniejącego folderu).

Jest cała gama komend w oparciu o VI, które pozwalają bezpośrednio edytować pliki konfigu - **VISUDO** (**/etc/sudoers**), **VIGR** (**/etc/group**), **VIPW** (**/etc/passwd**). **VIPW -s** edytuje **/etc/shadow**. UŻYWANIE TEGO JEST BARDZO NIEZALECANE BO NIE MA SPRAWDZANIA SKŁADNI I MOŻNA W OGÓLE ZJEBAC CAŁY SYSTEM!!!

Zmiana hasła to **PASSWD USERNAME** - oczywiście bez nazwy usera by zmienić hasło obecnego usera. **PASSWD -l** lockuje zmianę hasła. **PASSWD -u** odblokowuje.

PASSWD potrafi zmieniać ustawienia dla hasła:

- * **-n** jak długo MINIMUM hasło obowiązuje
- * **-w** na ile dni wcześniej pokazać warning

* **-x** kiedy expiruje

Druga komenda jest **CHAGE** z podobnymi parametrami (**ale roznymi jednak!!!**), a najwazniejszym jest flaga **-l** ktora pokazuje po prostu informacje o hasle usera.

USERADD USERNAME oczywiscie dodaje usera. Podczas tworzenia usera przydaja sie takie flagi:

- * **-m** tworzy folder domowy. Jak on wyglada (jego szkielet) jest brany z **/ETC/SKEL** (to jest folder z plikami)
- * **-u** nadaje recznie UID (jak juz zajety to wywali blad)
- * **-G** grup1,grup2 dodaje usera do wskazanych grup

Podczas tworzenia usera sa uzywane domyslne pliki, z ktorych bierze sie dane. Sa to **/ETC/LOGIN.DEFS**:

- * **MOTD_FILE**
- * **ENV_PATH** - definiuje \$PATH
- * **PASS_*** trzy zmienne z info o hasle (expiration i takie tam)
- * **UID_MIN** minimalny start zakresu dla nadawania UIDa userom
- * **CREATE_HOME** boolean
- * **USERGROUPS_ENAB** boolean - czy ma tworzyc primary grupe taka sama jak nazwa usera

oraz **/ETC/DEFAULT/USERADD** (domyslna konsola, home katalog, skel, czy tworzyc folder mejlowy i kilka innych).

USERDEL USERNAME usuwa. Z flaga **-r** usuwa tez jego environment (katalog domowy, mejle)

USERMOD modyfikuje uzytkownika rzecz jasna. Aczkolwiek do zmiany hasla to jak juz bylo pisane - **PASSWD**.

GROUPADD dodaje grupe. Z w sumie jedynej sensownej flagi to **-g** pozwala ustawic GUID recznie.

GROUPMOD pozwala edytowac grupe. Mozna zmienic nazwe czy GUID, ale nie da sie zmieniac przynaleznosci userow. To sie robi per user komenda **USERMOD (-aG)**, usuniecie usera z grupy to jak mowi internet najlepiej **VIGRem** ogarnac.

GROUPMEMS -g NAZWAGRUPY -l pozwala wylistowac wszystkich czlonkow danej grupy

/ETC/PASSWD - po kolei kolumny to:

- * username
- * password (nie uzywane juz)
- * UID - serwisowi liczeni do 999 (0 to ROOT), od 1000 domyslnie leca normalni userzy. Zakresy sa definiowane w **/ETC/LOGIN.DEFS**
- * GID - identyfikator primary grupy usera

- * komentarz (**GECOS** field)
- * directory - katalog domowy usera
- * shell - z mozliwoscia /sbin/nologin (/etc/nologin.txt by pokazac komunikat takim userom przy probie zalogowania)

/ETC/SHADOW

- * login name
- * zakodowanie haslo
- * dni od epoch kiedy ostatnio bylo zmieniane haslo
- * dni zanim haslo moze byc zmienione (ustawiane domyslnie na 0)
- * za ile dni haslo musi byc zmienione
- * ile dni przed wygasnieciem user dostaje ostrzezenie
- * ile dni po wygasnieciu hasla konto jest blokowane
- * ile dni od epoch haslo zostalo zablokowane (lepiej blokowac usera niz usuwac)
- * pole zarezerwowane, ktore jednak nigdy nie zostalo uzyte

/ETC/GROUP

- * nazwa grupy
- * password grupy (uzywane bardzo rzadko - do tymczasowego dania uprawnien do plikow)
- * group id
- * members

To construct the user environment, a few files play a role:

- /etc/profile: Used for default settings for all users when starting a login shell
- /etc/bashrc: Used to define defaults for all users when starting a subshell
- ~/.profile: Specific settings for one user applied when starting a login shell
- ~/.bashrc: Specific settings for one user applied when starting a subshell

LDAP jest hierarchiczny, distributed i replicated.

Narzedzia uzywane:

- * **authconfig**: A command-line utility in which you have to specify all you want to do by using command-line options
- * **authconfig-tui** (DEPRECATED) - uzywa **NSCLD** (/etc/nslcd.conf) jako backend: A menu-driven text user interface that allows you to select options to be used from a list. Use of this utility is recommended
- * **authconfig-gtk** - uzwa **SSSD** jako backend: A utility with a GUI, which for that reason can be used from a GUI environment only

Laczenie sie **SSSD** do **FreeIPA**:

```
yum install sssd sssd-client
```

```
authconfig --enablemkhomedir --enableldap --enableldapauth  
--ldapserver="192.168.56.104" --ldapbasedn="dc=example2,dc=pl" --update
```

```
scp chlebik@192.168.56.104:/var/ftp/pub/cacert.p12 /etc/openldap/cacerts/cacert.p12
```

```
authconfig --enableldaptls --update
```

Z TEGO CO WYNIKA TO RAPTEM NSLCD rozni sie tylko tym, ze instaluje sie:

```
yum install -y openldap-clients nss-pam-ldapd
```

A potem zmienia SELinuxa

```
restorecon /etc/openldap/cacerts/cert.pem --> nazwa certyfikatu byla inna
```

Linux Academy z kolei poleca inna rzecz - **REALMD** co jest z automatu zasztyte w CENTOS7.
Walimy (**REALM LIST**)

```
REALM DISCOVER AD_SERVER  
REALM_JOIN AD_SERVER
```

I teraz mozna sie polaczyc z innym serwerem po SSH uzywajac skladni:

```
SSH -l user@serwer ADRES_SERWERA_AD
```

ROZDZIAŁ 7 - CONFIGURATION PERMISSIONS

Rozpiska po **LS -L** to pierwszy znak oznacza:

- regular file (-)
- device (b)
- symbolic link (l)
- directory (d)
-

Jak się tworzy plik to user jest ownerem, a jego primary group jest właścicielem grupowym.

find / -user username

find / -group groupname

By zmienić ownera używamy **CHOWN KTO DO_CZEGO**. To się stosuje do plików i katalogów, a by poszło rekursywnie to surprise surprise leci flaga **-R**.

CHOWN może też zmieniać grupę, ale wtedy trzeba przed nazwą grupy (bez spacji) wrzucić **KROPKĘ** lub **DWUKROPEK**: **chown :grupa plik_do_zmiany**

Istnieje też fajna składnia **CHOWN USER(:) GRUPA DO_CZEGO** i wtedy ustawiamy od razu jedno i drugie

Jednakże lepiej do tego stosować **CHGRP** bo to specyficzna komenda. Składnia i flagi te same co w **CHOWN**.

By się zorientować jakie mamy grupy dajemy komendę **GROUPS**. Wtedy pokaże nasze. Jak dla roota to **GROUPS USERNAME**. Na liście grup pierwsza wymieniona jest PRIMARY. Da się to zmienić **DLA DANEJ SESSJI** za pomocą komendy **NEWGRP nazwa_grupy**. Trzeba być członkiem grupy by ją przypisać jako primary. Chyba, że grupa ma ustawione hasło (w **/etc/groups** się ustawia), albo komenda **GPASSWD** będąc członkiem - wtedy nieczłonkowie zostaną zapytani o hasło podczas próby zmiany.

Permission	Applied to Files	Applied to Directories
Read	Open a file	List contents of directory
Write	Change contents of a file	Create and delete files and modify
permissions on files		
Execute	Run a program file	Change to the directory (DEFAULT TO DIR)

Do zmiany uprawnień używa się komendy **CHMOD UPRAWNIENIE DO_CZEGO**. W dwóch trybach:

- * liczbowy to podawanie po kolei cyfr dla usera/grupy/innych (**4 read, 2 write, 1 execute**)
- * relatywny to KTO(+|-)CO **u+r** dla read dla usera Jak się nie wyspecyfikuje dla kogo to odbiera/dodaje wszystkim (lub prefix **a od all**). Kolejne uprawnienia można oddzielać przecinkami.

chmod -R o+rX /data to X na końcu sprawia, że to się aplikuje tylko do folderów!!!

Istnieją specjalne privilege:

- * **SUID** (np. /usr/bin/passwd) -> ustawia się u+s albo dając prefix 4 - co oznacza, że efektywny UID procesu, który ma dostęp do pliku to UID ownera pliku. **Nie da się tego ustawić na skryptach!!!**

- * **SGID** - stosowane do współdzielonych folderów - ustawia się g+s lub 2

- * **sticky bit** - aplikowane do folderów - mając write na folderze można tylko usunąć pliki jeśli jest się ownerem pliku LUB jest się ownerem folderu zawierającego. Przy **CHMOD** numerycznym prefix to **1!!!** Dla relatywnego to **CHMOD +t** styka. Ten 't' widac też na końcu informacji o uprawnieniach do folderu, np: **drw-r--r--t**

ACL to zaawansowane dostępy do plików/folderów. Generalnie problemy z tym są dwa:

- * nie wszystkie narzędzia wspierają ten temat (pisze, że **TAR** nie wspiera, ale wspiera z flagą **--acls**)

- * dysk może nie być zamontowany z obsługą tego. W **/etc/fstab** powinno być dodane **'acl mount'**

Do **ACL**i stosuje się metodę **SETFACL**. Ustawia ona tematy. Zanim jednakże się ją użyje dobrze jest spojrzeć sobie na plik/folder przez **GETFACL** by zobaczyć co się dzieje. **Komenda ll pokazuje ACL**e jako plusik na końcu uprawnień.

Wygląda to tak: **setfacl -m g:GRUPA:RELATIVEUPRAWNIENIA /dir** - Flaga **-m** to od **MODIFY**, reszta jest samoopisująca.

setfacl -m g:AVENGERS:rw /home/dev/secretproject

CO JEST ISTOTNE. Powyższa komenda ustawia zmiany dla **ISTNIEJĄCYCH** obiektów. Jeśli chcemy dorzucić dziedziczenie - czyli wszystkie nowe obiekty w folderze będą też mieć taki ACL trzeba dodać do uprawnień (po flagie **-m**) prefix **d:** !!! Ten prefix to **DEFAULT!**

To jest bardzo istotne. Powyższe zatem wygląda tak:

setfacl -m d:g:GRUPA:RELATIVEUPRAWNIENIA /dir

Usuwanie ACLi jest proste:

setfacl -X g:GRUPA /dir	dla grupy/usera
setfacl -b /dir	usunie totalnie wszystko
setfacl -k /dir	usunie DOMYSLNY ACL (ustawiony za pomoca flagi -d)

Mozna tez przepisac ACLe z jednego pliku/folderu na drugi:

getfacl COS | setfacl --set-file=- NA_CO

Istnieje cos takiego jak **UMASK**. To sa domyslne ustawienia dla plikow i folderow stosowane do wszystkich userow. Niestety sa one troche inaczej robione niz w przypadku **CHOWN**, ale dla plikow domyslnie jest **666** po czym dla kazdej cyferki odejmujemy to co jest w **UMASK** i dostajemy wynik. Dla grup startujemy od **777**.

Dla wszystkich userow najlepiej dodac skrypt w **/etc/profile.d** i w nim ustawic **umask**. Jesli chcemy tylko dla konkretnego usera to najlepiej robic to w pliku **.profile** w folderze domowym konkretnego usera. Jesli walniesz komende **UMASK** w konsoli to pokaze domyslna maske dla usera - ma tam 4 cyfry, a pierwsza to jest do sticky bitow i takich tam. **UMASK 0** po prostu usunie maske. **UMASK NIE JEST PERSYSTENTNE!!!**

Sa tez '**EXTENDED FILE ATTRIBUTES**' ale opisuje to w ksiazce tak skrotowo i po lebkach, ze cos czuje, ze tutaj za duzo to nie ma.

If you want to apply attributes, you can use the **chattr** command. For example, use **chattr +s somefile** to apply the attributes to somefile. Need to remove the attribute again? Then use **chattr -s somefile** and it will be removed. To get an overview of all attributes that are currently applied, use the **lsattr** command.

ROZDZIAŁ 8 - SIECI

Kluczem do wszystkiego jest komenda **IP**. **NIE NALEŻY UŻYWAĆ JUŻ IFCONFIG BO JEST PRZESTARZAŁE!!**

IP ADDR SHOW (skróty **IP A S** lub **IP A**) - pokazuje interfejsy sieciowe i dokładne dane

IP LINK SHOW - pokazuje interfejsy, ale tylko ich stan i MAC karty bez IP

IP -s LINK - pokazuje interfejsy oraz informacje o przesłanych pakietach

IP ROUTE SHOW - pokazuje routery dla konkretnych interfejsów sieciowych. **ROUTER MUSI BYĆ W TEJ SAMEJ SIECI CO INTERFEJSY!!!!!!!**

Porty mogą być otwarte lub zamknięte. Komenda **SS -LT** pokazuje wszystko (wcześniej był **NETSTAT**). **SS -TUL** pokazuje nie tylko TCP ale i UDP.

***** Some ports are only listening on the IPv4

loopback address 127.0.0.1 or the IPv6 loopback address ::1, which means that they are locally accessible only. Other ports are listening on *, which stands for all IPv4 addresses, or on :::*, which represents all ports on all IPv6 addresses *****

Do wylistowania urządzeń w systemie można też folderu - **/sys/class/net**

Do zarządzania połączeniami służy **NETWORK MANAGER**. To jest demon co wstaje i jak wstaje czyta skrypty konfiguracyjne dla network interfejsów które są zlokalizowane w **/ETC/SYSCONFIG/NETWORK-SCRIPTS** i muszą mieć prefix **IFCFG** i suffix z nazwą interfejsu. Jeżeli się przedyskutuje ten plik to by zmiany chwycił daje się **NMCLI CON RELOAD**.

Cokolwiek się nie zmieni za pomocą komendy **IP** jest 'nonpersistent' (cokolwiek to nie znaczy w kontekście sieci). By tematy były trwałe trzeba użyć komend **NMCLI** lub **NMTUI**. To pierwsze to oczywiście command line, a **TUI** to tekstowe. Oczywiście tekstowe się łatwiej używa.

NMCLI CON SHOW - pokazuje wszystkie połączenia i ich przypisania do urządzeń (aktywne i nieaktywne). Jak się jako ostatni parametr poda nazwę to się dostaje dodatkowe info o konkretnym połączeniu.

NMCLI DEV STATUS by zobaczyć status urządzeń (wszystkich)

NMCLI DEV SHOW <devicename> lub urządzenia podanego jako parametr

TUTAJ JEST DODAWANIE/MODYFIKOWANIE USTAWIEN POŁĄCZEN, ALE TO JEST GLUPE WPISYWANIE KOMEND BEZ WYTLUMACZENIA. Zakładają chyba, że musi być

używane **NMTUI**. Generalnie najważniejsze to jest samo dodanie połączenia:

NMCLI CON ADD CON-NAME TUTAJ_JAKAS_NAZWA OPCJE (autoconnect no ifname NAZWA_INTERFEJSU)

NMCLI CON UP/DOWN/DELETE NAZWA_POLACZENIA by wystartować

Za pomocą **NMTUI** da się:

- * edytować połączenie (wpierw trzeba po zmianach je **deaktywować** i potem **aktywować ponownie!!! Najlepiej SYSTEMCTL RESTART NETWORK**)
- * zmienić nazwę hosta
- * aktywować połączenie

By zmienić nazwę hosta można:

- * **hostnamectl set-hostname nazwa_hosta (HOSTNAMECTL STATUS** by sprawdzić)
- * z **NMTUI**
- * edytować z palca **/ETC/HOSTS**

Ustawienie DNSów: (domyślnie leżą w **/etc/resolv.conf**)

- * Use **nmtui** to set the DNS name servers.
- * Set the **DNS1** and **DNS2** in the **ifcfg network connection** configuration file in **/etc/sysconfig/network-scripts**.
- * Use a **DHCP** server that is configured to hand out the address of the **DNS** name server.
- * Use **nmcli con mod <connection-id> [+]ipv4.dns <ip-of-dns>**

Notice that if your computer is configured to get the network configuration from a DHCP server, the DNS server is also set via the DHCP server. If you do not want this to happen, you have two options:

- * Edit the **ifcfg configuration file** to include the option **PEERDNS=no** .
- * Use **nmcli con mod <con-name> ipv4.ignore-auto-dns yes** .

To verify host name resolution, you can use the **GETENT HOSTS SERVERNAME** command. This command searches in both **/etc/hosts** and DNS to resolve the hostname that has been specified.

Jeśli chce się zamienić kolejność resolvowania edytuje się plik **/ETC/NSSWITCH.CONF**

ROZDZIAŁ 9 - MANAGING PROCESSES

Są dwa typy procesów:

- * **shell job** - odpalane kiedy uruchamiamy komendę z shella i powiązane z nim. Nazywane również INTERACTIVE PROCESS.
- * **demony** - startujące najczęściej przy uruchomieniu systemu, najczęściej z prawami ROOTA. Procesy uruchamiają jeden lub więcej THREADS.

Jak się uruchamia **SHELL JOB** i się wie, że trochę potrwa to można wrzucić **&** za komendą (np: **command &**). To kładzie temat do backgroundu. Jak się chce to wrzucić do foregroundu to się robi komendą **FG**. Kiedy job zajmuje duuuuuuzo czasu to można mu dać **Ctrl+Z** co pauzuje proces i można sobie coś zrobić. Jak się chce odpauzować to dajemy **BG** i wtedy ładuje znowu w backgroundzie i się mieli. **Ctrl+C** przerywa proces i usuwa go z pamięci. **Ctrl+D** wysyła sygnał EOF do procesu, że niby on czeka na coś. Ale kurde nie wiem co to daje. Komenda, która listuje wszystkie joby, które są w backgroundzie to **JOBS**. Do **FG** i **BG** można dawać też parametr, który jest identyfikatorem (numerem porządkowym) procesu. **JAK SIE STARTUJE PROCES W BACKGROUNDZIE I UBIJE SHELLA TO ONE DALEJ ZYJA. KIEDYS SIE BY TO OSIAGNAC STOSOWALO NOHUP.**

Procesy startują workery (**threads**). Admin nic nie może z nimi robić - to programista to ogarnia. Co do procesów to istnieją dwa typy:

- * **kernel processes** - w **ps aux** widac, że ich nazwy są w nawiasach kwadratowych. Nie da się ich ubić czy zmienić priorytetu inaczej niż ubijając maszynę.
- * **real time processes** - odpalane przez userów

Do listowania procesów używa się komendy **PS** rzecz jasna:

- * **bez parametrow** - pokaże procesy wystartowane przez obecnego usera
- * **aux** - short summary of all active processes
- * **-ef** - pokazuje info o procesach, ale też pełną komendę, która go wywołała
- * **fax** - pokazuje hierarchie procesów
- * **o** - pozwala wyspecyfikować nazwy kolumn którymi jesteśmy zainteresowani

Jak jesteśmy zainteresowani tylko PIDem procesów, które np. mają jakąś nazwę to używamy **PGREP NAZWA**. Output to czysta lista PIDów z nowymi liniami. Parametry:

- l pokazuje nazwę procesu obok PIDu
- u limituje output do procesów danego usera (ostatnia flaga i trzeba podać nazwę usera)
- v invertuje wynik (czyli pokazuje wszystko co nie spełnia warunku wyszukiwania)

Kazdy proces startuje z domyslnym priorytetem **0 (-19 to NAJWYZSZY MOZLIWY PRIORYTET CZYLI NAJWAZNIEJSZY!!!)**. Mozna to zmieniac - komendy **NICE (przy starcie procesu - nice -LEVEL komenda)** i **RENICE (w runtime)** podajac zakres od **-20 do 19** (punktem wyjscia jest **20** jako domyslne wartosc, parametrem co sie podaje jest **-n WARTOSC**). Niceness jest podawana jako kolumna **NI** w outputcie komendy **TOP**.

Jednakze zwykly user moze tylko obnizac waznosc procesu czyli dodawac (im nizsza wartosc niceness tym proces wazniejszy).

Sa trzy typy sygnalow, ktore dzialaja na wszystkie procesy:

- * **SIGTERM (15)** prosi proces o zakonczenie sie
- * **SIGKILL (9)** killuje process
- * **SIGHUP(1)** zawiesza proces co skutkuje tym, ze proces odczytuje ponownie swoja konfiguracje
- * **SIGSTOP (19)** - pauzuje proces by go potem uruchomic ponownie - **SIGCONT (18)** startuje tak zatrzymany proces
- * **SIGSTP (20)** - proces ma sie zatrzymac (odpowiednik **Ctrl+Z**) - czyli pchniecie procesu do backgroundu.

Wysyla sie sygnaly do procesow za pomoca komendy **KILL**. **KILL -l** pokaze sygnaly, ktore mozna wykorzystac. Domyslnie **KILL PID** wysle **SIGTERM**. Lepiej nie uzywac **KILL -9** bo zostawia sie pierdolnik. **PKILL NAZWA** SIGTERMUje proces po nazwie. **KILLALL NAZWA** robi to samo, ale ze wszystkimi procesami, ktore maja te nazwe.

ROZDZIAŁ 10 - Virtual machines

KVM jest natywne w kernelu (kernel modules musi mieć **KVM**). Jednakże by z tego skorzystać trzeba mieć kawalki **QEMU** or demona o nazwie **LIBVIRT**. Konfig tego demona jest w **/ETC/LIBVIRT/LIBVIRT.D.CONF**.

By używać **KVM** potrzeba **64bit** systemu i akceleracji sprzętowej dla virtualizacji. Daj **ARCH** lub **UNAME -i** by się dowiedzieć czy spełniasz kryteria. Akcelerację sprzętową sprawdzamy **cat /proc/cpuinfo** -> powinno być **VMX** gdzieś w outputcie dla Intel'a i **SVM** dla AMD.

Do tego trzeba mieć miejsce na dysku - **/VAR/LIB/LIBVIRT/IMAGES** - tam się składują domyślnie obrazy VMek

By zainstalować co trzeba najlepiej - **yum groupinstall "Virtualization Host"**
Być może trzeba zrobić **SYSTEMCTL ENABLE LIBVIRT + START**

Generalnie najlepiej robić wszystko z GUI - **Virtual Machine Manager** - uruchamia się z palca - **VIRT-MANAGER &**

Do instalacji VMki z command-line się robi **VIRT-INSTALL**

Do zarządzania VMkami ma się też polecenie z command line - **VIRSH**.

- * **list** Shows all VMs that are currently active
- * **list --all** Shows all VMs, including machines that are not currently active
- * **help** Gives a list of all parameters that can be used with the virsh command
- * **shutdown <vmname>** Shuts down the VM properly
- * **destroy <vmname>** Halts a VM, similar to pulling the power plug on a real computer
- * **edit <vmname>** Opens a vi interface that allows you to edit the XML configuration file belonging to a specific VM244
- * **console <vmname>** Connects to a VM directly from the console of a KVM host server
- * **start <vmname>** Starts a VM
- * **reboot <vmname>** Reboots a VM

CHAPTER 11 - Managing Software

Yum stoi od Yellowdog update manager. **EPEL** to jest repozytorium specyficzne dla Fedory i jako takie nie jest zalecane do systemow produkcyjnych. Ale mozna dodac jak sie chce miec cutting edge (**ale nie na RHELU bo sie wysypie support**).

Generalnie informacje o repozytoriach sa trzymane w plikach **.REPO**. Informacje sa przechowywane w **/ETC/YUM.REPOS.D/** Nie ma do tego GUI wiec trzeba klepac z palucha. Istnieje mozliwosc wpisania kilku repozytoriow w jednym pliku. W pliku mozna wrzucic (jak cos to **MAN YUM.CONF** i szukaj przykladow):

- * **[label]** The label used as an identifier in the repository file.
- * **name=** The name of the repository.
- * **mirrorlist=** Refers to a URL where information about mirror servers for this server can be obtained. Typically used for big online repositories only.
- * **baseurl=** The base URL where to go to find the RPM packages
- * **gpgcheck=** Set to **1** if a **GPG** integrity check needs to be performed on the packages. If set to **1**, a **gpgkey** is required.
- * **gpgkey=** Specifies the location of the **GPG** key that is used to check package integrity.

Generalnie RHEL sobie ogarnia temat tak, ze repozytoria maja rozne statusy/labele.

- * **base** - This is the base repository that contains all essential Red Hat software. Its packages are fully supported.
- * **updates** - A specific repository that contains updates only.
- * **optional** - This repository contains packages that are provided for the convenience of Red Hat customers. The packages in this repository are open source and not supported by Red Hat.
- * **supplementary** - This repository contains packages that are provided for the convenience of Red Hat customers. The packages in this repository are proprietary and not supported by RHat.
- * **extras** - repository contains packages that are provided for the convenience of Red Hat customers. Software in this repository comes from different sources and is not supported by RH
- * **@anaconda** - to nie jest tak naprawde repo, ale na listach pakietow czy cos jest uzywane jako zrodla pochodzenia pakietu przy instalacji (@anakonda to installer)

Dla security wiekszosc repozytoriow jest podpisywana za pomoca **GPG** - jest to ustawiane w pliku z repozytorium. Za pierwszym polaczeniem sie pyta czy pociagnac ten klucz na lokalnego kompa i jesli tak to potem przy update bedzie porownywal podpisy. Domyslnie te klucze laduja

w **/ETC/PKI/RPM-GPG**.

YUM ma szereg polecen:

- * **repolist** - pokazuje liste repozytoriow
- * **search** - Search for the exact name of a package
- * **[what]provides */name** - Perform a deep search in the package to look for specific files within the package
- * **info** - Provide more information about the package
- * **install** - Install the package (Z opcja **-y** to nie bedzie pytal o potwierdzenie)
- * **remove** - Remove the package
- * **list [all | installed]** - List all or installed packages (mozna tez dac jako parametr na koncu nazwe)
- * **group list** - List package groups - mozna dac **GROUP LIST HIDDEN** - to pokaze o wiele wiecej
- * **group install** - Install all packages from a group
- * **group info XXX** - daje info o calej grupie
- * **update** - Update packages specified (**KERNEL** nigdy nie jest domyslnie nadpisywany, ale instalowany obok istniejacego)
- * **clean all** - Remove all stored metadata
- * **history** - wszystkie logi z YUMa sa zapisywanie do **/VAR/LOG/YUM.LOG**. Generalnie jak sie pokaze liste to mozna dac **HISTORY UNDO XXX** gdzie **XXX** to jest numer akcji. Wtedy sie 'unduuje' to

Generalnie dalej (choc deprecated) jest **RPM**. On tylko instalowal, a nie resolwoval zaleznosci. Dlatego tez obecnie sie go nie stosuje, a paczki RPMowe mozna po prostu instalowac za pomoca **YUM INSTALL NAZWAPACZKINADYSKU.RPM** - jest to o tyle tez istotne, ze zarowno **YUM** jak i **RPM** maja swoje bazy danych pakietow. **YUM** uaktualnia tez **RPMowa**, ale w druga strone to nie dziala.

By przepytac baze **RPMowa** dajemy **RPM -QA** - pokazuje wszystko co jest zainstalowane (jak sie ma nazwe to GREPUj po niej).

By dostac informacje o specyficznym pakiecie dajemy **RPM -QI NAZWA_PAKIETU**

Wszystkie pliki w pakiecie: **RPM -QL NAZWA_PAKIETU**

Dokumentacja w pakiecie: **RPM -QD NAZWA_PAKIETU**

Pliki konfiguracyjne: **RPM -QC NAZWA_PAKIETU**

Jak masz info o pliku jakims to mozesz dowiedziec sie w jakim pakiecie jest: **RPM -QF NAZWA_PLIKU**

By zobaczyc czy sa jakies skrypty w pakiecie: **RPM -Q --scripts** (Takze do zastosowania z plikami)

Pokazuje jakie czesci packagu zostaly zmienione od instalacji: **RPM -V** (dla jednego pakietu) - wersja z flaga **-Va** pokaze dla wszystkich pakietow.

Domyslnie queriesy leca po bazie danych. Jesli jednakze masz jakis plik juz sciagniety to uzywajac flag **-P** mozna zapytac ten konkretny plik o cos (glownie z druga flaga **--scripts**). Zasadniczo jesli chcesz querowac repo zdalne no to masz problem. Bo by to zrobic trzeba zainstalowac pakiet **YUM-UTILS**, ktory daje kilka ciekawych bajerkow.

REPOQUERY ma w sumie prawie taka sama funkcjonalnosc jak **RPM -Q** (tylko, ze bez **--scripts**). Z kolei jak sie chce sciagnac z repo tylko plik to potrzebny jest util **YUMDOWNLOADER**.

CHAPTER 12 - Managing recurring tasks

Generalnie sa dwa. Najwazniejszy jest **CRON**, który jest demonem startujacym razem z systemem (RHEL uzywa go internal - chocby do rotowania logow). Demon co minute sie budzi, patrzy co ma odpalic i jak cos to odpala. Sprawdzamy status **SYSTEMCTL STATUS CROND -l**. Cron korzysta ze swojej stringowej skladni:

- * minute 0–59
- * hour 0–23
- * day of month 1–31
- * month 1–12 (or names which are better avoided)
- * day of week 0–7 (Sunday is 0 or 7, or names [which are better avoided])

- o) * 11 * * * Any minute between 11:00 and 11:59 (probably not what you want)
- o) 0 11 * * 1-5 Every day at 11 a.m. on weekdays only
- o) 0 7-18 * * 1-5 Every hour on weekdays on the hour
- o) 0 */2 2 12 5 Every 2 hours on the hour on December second and every Friday in December

Konfig jest w **/ETC/CRONTAB**, ale **tam sie nigdy nic nie zmienia!** Daje jednakze ten plik sporo informacji. To sa miejsca, gdzie sie zaczytuje konfig:

- * Cron files in **/etc/cron.d** - **MUSZA BYC EXECUTABLE BY SIE WYKONALY!!!!!!!!!!**
- * Scripts in **/etc/cron.hourly, cron.daily, cron.weekly, and cron.monthly**
- * User-specific files that are created with **crontab -e**

Kazdy user moze miec swojego crontaba. By go edytowac wali sie **CRONTAB -E**. Zmiany sa dopisywane do folderu **/VAR/SPOOL/CRON** w pliku per user. Jednak tych plikow nie wolno tez edytowac. Edytuje sie zawsze za pomoca **CRONTAB -E**. ROOT moze edytowac crontaba poszczegolnych userow za pomoca **CRONTAB -E -U NAZWA_USERA**.

Wszystkie inne crony beda odpalane z poziomu **ROOTa**. Najlepszym sposobem to wrzucenie pliku spelniajacego syntax crontaba do **/ETC/CRON.D** -> bedzie sie mielilo samo.

Ostatnimi sa convenience pliki w folderze **/ETC (cron.daily, hourly, weekly, monthly)**. Z definicji to installery pakietow tam cos wrzucaja.

Istnieje taki serwis jak **ANACRON (TYLKO DLA ROOTA)**. Jego plik konfiguracyjny jest oczywiscie w **/ETC/ANACRON** -> to jest glownie uzywane do schedulowania taskow, ktore maja byc wykonywane np. raz dziennie bez specyfikacji dokladnie kiedy. Moza to zakladac np. przy serwerach, ktore sa wylaczane na jakis przedzial czasu w ciagu dnia czy cos. Plik konfiguracyjny mowi jasno jak to dziala.

Security w cronie jest proste - istnieją pliki **/ETC/CRON.ALLOW** i **/ETC/CRON.DENY**. Jak ten pierwszy istnieje to user MUSI być w nim by w ogóle dopisywać do crona. Tak samo jeśli istnieje ten drugi to user **NIE MOŻE** w nim być by coś dopisać.

Drugim serwisem, który ogarnia egzekucję czasową jest **AT** (ATD - AT DEMON). Generalnie można pisać np: **at 14:00 lub at noon**. Jak się to wpisze to pojawia się konsola gdzie pisze się polecenia do wykonania. Konsolę zamyka się kombinacją **Ctrl+D**. Do listowania tematu daje się **ATQ** (od queue) która podaje też numerki. Tę numerkę można użyć w komendzie **ATRM NUMER** co spowoduje usunięcie zaplanowanego zadania. Też istnieje **/ETC/AT.DENY** i **ALLOW**

ROZDZIAŁ 13 - CONFIGURING LOGGING

Loguje sie na 3 sposoby:

- a) zwykle pchanie logow do plikow tekstowych
- b) **JOURNALD** - to jest serwis co przychodzi z **SYSTEMD**. Generalnie zapisuje boot procedure, kernel i serwisy do plikow binarnych. **NIE SA PERSYSTENTNE MIEDZY REBOOTAMI!** By zatem logi nie znikaly **JOURNALD** i tak pcha logi do ->
- c) **RSYSLOG** - generalnie serwis pchajacy logi do plikow znajdujacych sie w **/VAR/LOG**. Dodaje on tez troche swoich tematow, a takze mozliwosc filtrowania czy persystencji na remote.

Na co dzien admini monitoruja:

- a) **/VAR/LOG** obviously
- b) komenda **JOURNALCTL**
- c) komenda **SYSTEMCTL STATUS <unit>** i jak cos poszlo nie tak to poleci info o ostatnich komendach

Wazne pliki w **/VAR/LOG**:

- * **/var/log/messages** The most commonly used log file, it is the generic log file where most messages are written to.
 - * **/var/log/dmesg** Contains kernel log messages.
 - * **/var/log/secure** Contains authentication related messages. Look here to see which authentication errors have occurred on a server.
 - * **/var/log/boot.log** Look here for messages that are related to system startup.
 - * **/var/log/audit/audit.log** Contains audit messages. SELinux writes to this file.
 - * **/var/log/maillog** Look here for mail-related messages.
 - * **/var/log/samba** Provides log files for the Samba service. Notice that Samba by default is not managed through rsyslog, but writes directly to the **/var/log** directory.
 - * **/var/log/sss** Contains messages that have been written by the sssd service, which plays an important role in the authentication process.
 - * **/var/log/cups** Contains log messages that were generated by the print service CUPS.
 - * **/var/log/httpd/** Directory that contains log files that are written by the Apache web server.
- Notice that Apache writes messages to these files directly and not through rsyslog!**

By pisac do logow uzywa sie polecenia **LOGGER**. Najprosciej to po prostu dac message zaraz po tym. Mozna tez wyspecyfikowac (flaga **-p od priority**) unit do ktorego to poleci lub po prostu priorytet.

RSYSLOGD oczywiscie ma swoj plik konfiguracyjny w **/ETC/RSYSLOGD.CONF** - domyslnie

ten plik zaciaga tez zawartosc **/ETC/RSYSLOG.D** -> tam laduja skrypty konfiguracyjne logowanie z innych pakietow RPMowych. Zatem dobrze jest tam zajrzec. Domyslnie tez sa parametry startowe dla uslugi dostepne w pliku **/ETC/SYSCONFIG/RSYSLOG** gdzie jest zmienna **SYSLOG_OPTIONS**.

W pliku **/ETC/RSYSLOGD.CONF** mamy **RULES**, ktore okreslaja w jaki sposob jakie message sa logowane:

- * A facility specifies a category of information that is logged. Rsyslogd uses a fixed list of facilities, which cannot be extended. This is because of backward compatibility with the legacy syslog service.
- * A priority is used to define the severity of the message that needs to be logged. When specifying a priority, by default all messages with that priority and all higher priorities are logged.
- * A destination defines where the message should be written to. Typical destinations are files, but rsyslog modules can be used as a destination as well (**:NAZWAMODULU:**), to allow further processing through an rsyslogd module.

Czyli dla przykladu wyglada to tak:

*.info;mail.none;authpriv.none;cron.none /var/log/messages -> Facilities.poziom

oddzielone srednikami i na koncu lokalizacja. Poziom moze byc opisowy, albo liczbowy (jak liczba to dokladnie ten poziom oraz wyzsze beda logowane; jak sie chce **TYLKO** ten poziom to daje sie **facility.=poziom**)

LOKALIZACJA MOZE BYC TEZ INNYM MODULEM!!! Plus jesli do lokalizacji plikowej doda sie myslnik (przed) to bedzie sie buforowalo i zapisywalo tylko raz na jakis czas.

Lista facilities:

- * auth / authpriv - Messages related to authentication.
- * cron - Messages generated by the crond service
- * daemon - Generic facility that can be used for nonspecified daemons.
- * kern - Kernel messages.
- * lpr - Messages generated through the legacy lpd print system.
- * mail - Email-related messages.
- * mark - Special facility that can be used to write a marker periodically.
- * news - Messages generated by the NNTP news system.
- * security - Same as auth / authpriv. Should not be used anymore.
- * syslog - Messages generated by the syslog system.
- * user - Messages generated in user space.
- * uucp - Messages generated by the legacy UUCP system.
- * local0-7 - Messages generated by services that are configured by any of the local0 through local7 facilities

Z kolei severity levels mogą być takie:

- * **debug** - Debug messages that will give as much information as possible about service operation.
- * **info** - Informational messages about normal service operation.
- * **notice** - Used for informational messages about items that might become an issue later.
- * **warning / warn** - Something is suboptimal, but there is no real error yet.
- * **err /error** - A noncritical error has occurred.
- * **crit** - A critical error has occurred.
- * **alert** - Used when the availability of the service is about to be discontinued.
- * **emerg / panic** - Message generated when the availability of the service is discontinued

Oczywiście by logi się nie przepelniły istnieje **LOGROTATE**. Plik konfiguracyjny w **/ETC/LOGROTATE.CONF**

JOURNALD to serwis co zapisuje dane w binarnym pliku **/RUN/LOG/JOURNAL**. By się poruszać po tym trzeba używać **JOURNALCTL**. Wrzutka bez parametrów pokaże początek journala (czyli info podczas startu).

Wywołanie z **-F** pokaże ostatnie linie. Jest też **--no-pager** (bo domyślnie odpala to w lessie). Podobnie jak w **TAIL** jest przełącznik **-N**. Flaga **-P** facility przefiltruje tylko dany facility. Jest też **--since** i **--until** z datą w wersji **DDMMYYYY** do filtrowania też.

Oczywiście to nie jest persystentne, ale może być jak się stworzy plik: **/VAR/LOG/JOURNAL**, szczegóły dotyczące rotowania tego pliku są w **/ETC/SYSTEMD/JOURNALD.CONF**.

ROZDZIAŁ 14 - MANAGING PARTITIONS

Przy **MBRze** ilość **PRIMARY PARTITIONS** to **MAX 4**. Jednakże by to obejść można stworzyć więcej **EXTENDED PARTITIONS** w ramach jednej **PRIMARY PARTITION**. Na **EXTENDED PARTITION** mogłoby być **MAX 15 partycji**. **MAX rozmiar partycji w MBRze to 2 TB**.

Oczywiście powyższe w świetle dzisiejszych dysków to wybitnie za mało. Więc powstało **GUID PARTITION TABLE (GPT)**. Na nowych kompach co używają **UEFI** to jest jedyny sposób by się dostać do dysków i je poustawiać. Cechy:

- * max rozmiar to **5 ZiB (zebibytes)** - nie ma limitu na 2 TiB oczywiście
- * do **128** partycji
- * nie ma sensu dzielić tego na primary, extended i logical partitions (bo nie ma jak w MBRze limitu 64 KB na info o partycjach)
- * używa się **128-bit GUIDa** jako identyfikatora partycji
- * automatycznie na końcu dysku robi się kopię zapasową tabeli partycji

Do operowania na partycjach czy coś potrzebujemy jak zawsze nazwy dysku (leci /sda, /sdb, a jak się skończy to /sdaa):

- * **/dev/sda** - A hard disk that uses the SCSI driver. Used for SCSI and SATA disk devices. Common on physical servers but also in VMware virtual machines.
- * **/dev/hda** - The (legacy) IDE disk device type. You will seldom see this device type on modern computers.
- * **/dev/vda** - A disk in a **KVM** virtual machine that uses the **virtio** disk driver. This is the common disk device type for KVM virtual machines.
- * **/dev/xvda** - A disk in a Xen virtual machine that uses the Xen virtual disk driver. You see this when installing RHEL as a virtual machine in Xen. RHEL 7 cannot be used as a Xen hypervisor, but you might see RHEL 7 virtual machines on top of the Xen hypervisor using these disk types

Do tworzenia partycji **MBR** używamy **FDISK**. Jest to w tym stare narzędzie. Generalnie dobrze poczytać to: <https://www.binarytides.com/linux-command-check-disk-partitions/>
Samo tworzenie partycji trzeba zacząć od tego, że się wie co się ma obecnie. Wziąć się **FDISK -L** i on pokazuje podziały (oczywiście na ROOT). Generalnie na minimalnej instalacji w Centosie w VBoxie poszedł **/dev/sda**.

- * Więc by stworzyć coś nowego daje się **FDISK LINK_DO_URZADZENIA**. Pojawia się konsola **FDISK**.
- * By stworzyć coś nowego klepiemy **N** i wybieramy typ partycji (domyślnie primary).
- * Potem leci rozmiar (start bloków), a potem końcówkę - tutaj najsensowniej nie wpisywać całości tylko po prostu dać **+ROZMIAR(M|GB)**.
- * Póki co zmiany są tylko wirtualne. By to fizycznie zapisać do **MBR** trzeba dać komendę **W**.
- * Jak polecą błędy, że kernel używa starej tablicy partycji (porównaj **fdisk -l LINK_DO_DYSKU** z

cat /proc/partitions) to trzeba odswiezyc tablice kernela. **PARTPROBE LINK_DO_DYSKU**

Utworzylismy partycje **PRIMARY**. Jednakze mamy jeszcze jeden slot na partycje to utworzymy sobie **EXTENDED** tylko po to by na niej stworzyc **LOGICAL**. Na **EXTENDED** normalnie nie mozna tworzyc filesystemow - ona sluzi tylko jako opakowanie na **LOGICAL**.

Jezeli dysk ma wiecej niz **2 TiB** lub juz byl konfigurowany za pomoca **GPT** no to generalnie trzeba uzywac polecenia **GDISK**. Generalnie **FDISK** ma jakies tam wsparcie dla **GPT**, ale to jest niestabilne wiec lepiej tego nie uzywac. **NIE UZYWAJ GDISK JEZELI NA DYSKU SA JUZ PARTYCJE FDISK!!!** Generalnie u mnie **GDISK** nie byl zainstalowany w minimal distribution. Do tego w sumie uruchamia sie i uzywa jak **FDISK**.

Utworzenie partycji jeszcze niczego z nia nie robi. By cos sie dalo na niej ogarnac trzeba stworzyc na niej filesystem. To, ze wybieralismy 'Linux Filesystem' przy tworzeniu partycji to jeszcze nic nie znaczy. Do wyboru mamy:

- * **XFS** --- The default file system in RHEL 7.
- * **Ext4** --- The default file system in previous versions of RHEL. Still available and supported in RHEL 7.
- * **Ext3** --- The previous version of Ext4. On RHEL 7, there is no real need to use Ext3 anymore.
- * **Ext2** --- A very basic file system that was developed in the early 1990s. There is no need to use this file system on RHEL 7 anymore.
- * **BtrFS** --- A relatively new file system that was not yet supported in RHEL 7.0 but will be included in later updates.
- * **NTFS** --- Not supported on RHEL 7.
- * **VFAT** --- A file system that offers compatibility with Windows and Mac, it is the functional equivalent of the FAT32 file system. Useful to use on USB thumb drives that are used to exchange data with other computers but not on a server's hard disks.

Komenda co formatuje partycje na konkretny typ to **MKFS -T TYP_PARTYCJI** (**domyslnie jest ext2 wiec trzeba ustawic typ**). Sa tez dedykowane narzedzia (np. **mkfs.ext4**). Po tym jak sie stworzy filesystem mozna (w przypadku **ext2-4**) zmieniac ustawienia tego filesystemu za pomoca komendy **TUNE2FS**.

Na poczatku **TUNE2FS -L ADRES_PARTYCJI** pokaze co mozna z tym zrobic. Do tego:

- * **TUNE2FS -o ATRYBUTY,ATRYBUT** pozwala ustawic atrybuty, ktore normalnie ustawia sie w **/ETC/FSTAB**. By cos wylaczyc trzeba dac ^ przed atrybutem
 - * **TUNE2FS -O FEATURE** pozwala ustawic feature filesystemu. Nic wiecej na ten temat koles nie pisze (wylacza sie tak samo jak powyzsze)
 - * **TUNE2FS -L NAZWA** pozwala ustawic labelke dla filesystemu (alternatywa to **E2LABEL**)
- Dla **XFS** narzedzie to **XFS_ADMIN**.

CHAPTER 15 - Logical Volumes

Generalnie jest tak:

File Systems

^ ^ ^
| | |

Logical Volumes

^ ^ ^
| | |

Volume group (to jest abstrakcja)

^ ^ ^
| | |

Physical storages and devices

LVM jest fajny bo:

* umożliwia dynamiczną zmianę rozmiaru - **volume group** może urosnąć. Można też zmienić rozmiar **LOGICAL VOLUME**, ale tylko wtedy kiedy filesystem na niej to umożliwia. **EXT4** to ma, ale **XFS NIE!!!**

* Drugim bonusem są **SNAPSHOTS**. Nie za wiele oni o tym piszą - to jest tylko snapshot własnie, ale detailsy mają być w drugim egzaminie.

* Trzecia rzecz jest ponoć łatwość wymiany fizycznych nośników pod spodem - bo modyfikujemy **VOLUME GROUP**, a nie dyski per se

Zatem by utworzyć **LVM** trzeba przejść proces, który stworzy te trzy warstwy. Generalnie w objectives jest napisane, że powinno się umieć stworzyć każdą z nich, rozszerzać i usuwać.

Generowanie do tego wszystkiego służą komendy co się zaczynają od **PV, VG, LV**. Tworzymy sobie partycje za pomocą **FDISK/GDISK** w zależności co już mamy.

U mnie wyszło, że partycja ta to **/DEV/SDB2**. Jak się ją stworzy to trzeba utworzyć na niej **PHYSICAL VOLUME**.

Komenda **PVCREATE /DEV/SDB2** (weryfikacja **PVS** (najbardziej human readable) lub **PVDISPLAY** lub **LSBLK**).

Teraz przyszedł czas na **VOLUME GROUP**. Komenda jest prosta - **VGCREATE NAZWA PARTYCJA**. Jeśli do tej pory partycja nie była uznawana za **PHYSICAL VOLUME** to ta komenda ją zrobi. By wylistować wszystkie **VOLUME GROUPY** używa się oczywiście komendy **VGS** lub **VGDISPLAY** by dostać więcej info.

Stworzenie **LOGICAL GROUP** jest trochę bardziej skomplikowane. Polecenie to **LVCREATE**, ale przydają się flagi:

- * **-n NAZWA** - daje nazwę co pozwala porządek jakiś wprowadzić
- * **-L wielkosc** - TO JEST WIELKOŚĆ ABSOLUTNA I MA ZNACZENIE ROZMIAR LITER!!!
- * **-l wielkosc** - to jest wielkość relatywna. Dla przykładu **-l 50%FREE** i weźmie **50% wolnej przestrzeni z VOLUME GROUP**.

Dostęp do **LOGICAL VOLUME** odbywa się za pomocą sekwencyjności.

/DEV/VOLUMEGROUPNAME/LOGICALNAME

Istnieje coś takiego też jak **DEVICE MAPPER**. Tworzy on wpisy w **/DEV/DMX** gdzie **X** to cyferki, ale niewiele to mówi więc istnieją też wpisy w **/DEV/MAPPER**, które potem się patternują na **VOLUMEGROUP-LOGICALGROUP** np. **/dev/mapper/vgdata-lvdata**

Oczywiście to nie koniec - by działać na **LVM** trzeba na **LOGICAL VOLUME** utworzyć filesystem i go zamontować, ale to już działa tak samo jak w partycjach.

Zmiana rozmiaru **LOGICAL GROUP** jest zawsze możliwa w **GORE**. Zmiana w dół jest **NIEMOŻLIWA w XFS!** W dół możliwa jest w **EXT4** czy **BTRFS**, ale tylko po odmontowaniu.

Co do **VOLUME GROUP** na **RHCSA** trzeba tylko wiedzieć jak go rozszerzyć. Robi się to komendą **VGEXTEND NAZWA_GRUPY NAZWA_PHYSICAL_VOLUME** (jak więcej to oddzielone spacjami)

Do rozszerzania **LOGICAL VOLUMES** i przy okazji filesystemu na nich używa się **LVEXTEND** (tylko w gore) albo **LVRESIZE** (w gore i w dol). **Flaga -r oznacza też rozszerzenie znajdującego się na wolumie filesystemu.**

■ **lvresize -r -l 75%VG /dev/vgdata/lvdata** This resizes the logical volume so that it will take 75% of the total disk space in the volume group.

■ **lvresize -r -l +75%VG /dev/vgdata/lvdata** This tries to add 75% of the total size of the volume group to the logical volume. (Notice the difference with the previous command.)

■ **lvresize -r -l +75%FREE /dev/vgdata/lvdata** This adds 75% of all free

disk space to the logical volume.

■ **lvresize -r -l 75%FREE /dev/vgdata/lvdata** This resizes the logical volume to a total size that equals 75% of the amount of free disk space. (Notice the difference with the previous command.)

Do zmniejszania używa się **LVREDUCE** ze znakiem minus przy flagie **-l**. **Podczas tej operacji volume jest automatycznie odmontowywany!**

CHAPTER 16 - Basic Kernel Management

Kernel przekazuje instrukcje między programami, a CPU czy IO. Generalnie używa do tego **THREADS** (wątków), które można zlistować **PS AUX**. By wiedzieć co się dzieje z kerneliem mamy trzy narzędzia:

* **DMSG** - umożliwia podgląd czegoś co się nazywa **KERNEL RING BUFFER**. Tam sobie jąderko trzyma ostatnie informacje co się w nim działo. Innym sposobem by się do tego dostać jest **JOURNALCTL** (z flagą **-k** lub **--dmesg**)

* **/proc filessystem** - no tam to się dzieje wiele ciekawego. Książka listuje tylko **/PROC/MEMINFO**

* **uname** - jest to uniwersalne narzędzie do sprawdzania systemu i ogólnie kernela. (flagi **-a** lub **-r**). Generalnie sporo pokazuje też **HOSTNAMECTL STATUS**. Do tego jest plik, który dokładnie pokazuje wersję linuxa co się używa - **/etc/redhat-release**

Kernel jak wstaje to ogarnia sprzęt i moduły tak:

1. During boot, the kernel probes available hardware.
2. Upon detection of a hardware component, the **systemd-udevd** process takes care of loading the appropriate driver and making the hardware device available.
3. To decide how the devices are initialized, **systemd-udevd** reads rules files in **/usr/lib/udev/rules.d**. These are system provided udev rules files **that should not be modified**.
4. After processing the system provided udev rules files, **systemd-udevd** goes to the **/etc/udev/rules.d** directory to read any custom rules if these are available.
5. As a result, required kernel modules are loaded automatically and status about the kernel modules and associated hardware is written to the **sysfs** file system which is mounted on the **/sys** directory.

Generalnie to nie jest pojedynczy proces. Potem na bieżąco kernel monitoruje co się dzieje no bo się czasem wkłada nowy sprzęt czy coś. By zobaczyć na bieżąco co się dzieje można użyć komendy **UDEVADM MONITOR** co nas wrzuci do interaktywnej konsoli. Jeśli istnieją jakieś moduły, które chcemy by były wrzucane ręcznie wtedy istnieje folder **/ETC/MODULES-LOAD.D/** i tam trzeba wrzucać pliki ***.CONF**, które w sobie mają po prostu nazwę modułu do załadowania (jak coś oddzielone znakiem nowej linii).

Do pracy z modułami jest kilka poleceń.

* **LSMOD** - listuje obecnie używane moduły

* **MODINFO** - pokazuje szczegóły konkretnego modułu kernela. Czyli **MODINFO NAZWA**

* **MODPROBE** - ładuje wskazany moduł. **MODPROBE -r NAZWA** onloaduje wskazany moduł.

!!!!!!!Używanie starych poleceń **INSMOD** i **RMMOD** jest deprecated!!!!!!

Obecnie jak sie chce zobaczyc sprzet i czy sa do niego moduly daje sie komende **LSPCI -K** (instalowany z pakietu **PCIUTILS**). Pokazuje co jest podlaczone do magistrali **PCI**.

Jak sie chce ladowac moduly do kernela z dodatkowymi jakimis parametrami to trzeba stworzyc plik w **/ETC/MODPROBE.D/***.CONF i wtedy wali sie w formule: **options NAZWA_MODULU KEY=VALUE**

Jak sie chce podniesc wersje kernela to sie daje **YUM UPGRADE/INSTALL KERNEL**.

Oczywiscie instalator nie usunie poprzedniego tylko go zostawi (domyslnie 4 poprzednie wersje beda w folderze **/boot**).

CHAPTER 18 - Managing Boot Procedure

SYSTEMD ma za zadanie startować rzeczy - nazywa się je **UNITami**. Najważniejszym są serwisy, ale są też sockety, mounty i kilka innych. Komenda by je listować to **SYSTEMCTL -T HELP**.

Zaletą **SYSTEMD** jest to, że startuje się wszystkie unity tak samo. Defaulty są w folderze **/USR/LIB/SYSTEMD/SYSTEM**. Specyficzne overwrites są w **/ETC/SYSTEMD/SYSTEM**. Jeśli jakiś konfig jest generowany z automatu ładuje w **/RUN/SYSTEMD/SYSTEM**.

Każdy z plików składa się z sekcji (niektóre unity mają swoje specyficzne sekcje):

- **[Unit]**, which describes the unit and defines dependencies. This section also contains the important **After** statement, and optionally the **Before** statement.

These statements define dependencies between different units. The **Before** statement relates to another unit that is started after this unit. The **after** unit refers to a unit that needs to be started before this unit can be started.

- **[Service]**, in which there is a description on how to start and stop the service and request status installation. Normally, you can expect an **ExecStart** line, which indicates how to start the unit, or an **ExecStop** line, which indicates how to stop the unit.

- **[Install]**, in which the wants are taken care of. You'll read more about this in the next section, "Understanding Target Units."

Tych opcji generalnie jest multum - by je pokazać **SYSTEMCTL SHOW NAZWA_UNITA**.

Konkretnie **UNITy** są zgrupowane w **TARGETACH** - one w sobie zawierają szereg **UNITów** i można powiedzieć, że odpowiadają starym **RUN LEVELOM**. **TARGETy** tylko są po to by grupować **UNITy**, ale same nie zawierają informacji jakie to mają być unity. Te informacje są zapisane w poszczególnych plikach opisujących unity. Kiedy się **ENABLUJE UNITA** (czyli chce by wstawał przy starcie systemu) jest tworzony symlink w **/ETC/SYSTEMD/SYSTEM**.

SYSTEMCTL to jest podstawowy punkt wejścia do **SYSTEMD**. Generalnie składnia to

SYSTEMCTL POLECENIE NAZWA_UNITA.

- * **stop**
- * **start**
- * **status**
- * **enable** (będzie się z automatu uruchamiać przy starcie)
- * **disable** (zaprzeczenie powyższego)

Jak sie pyta o status uslugi to sie dostaje takie cos:

- * **Loaded** - The unit file has been processed and the unit is active.
- * **Active(running)** - Running with one or more active processes.
- * **Active(exited)** - Successfully completed a one-time configuration.
- * **Active(waiting)** - Running and waiting for an event.
- * **Inactive** - Not running.
- * **Enabled** - Will be started at boot time.
- * **Disabled** - Will not be started at boot time.
- * **Static** - This unit can not be enabled but may be started by another unit automatically

Polecenia dla admina, ktore pozwala troche tematu ogarnac (co jest wlaczone)

- * **systemctl --type=service** Shows only service units
- * **systemctl list-units --type=service** Shows all active service units (same result as the previous command)
- * **systemctl list-units --type=service --all** Shows inactive service units as well as active service units
- * **systemctl --failed --type=service** Shows all services that have failed
- * **systemctl status -l your.service** Shows detailed status information about services

Kazdy **UNIT** ma swoje zaleznosci. W sensie moze wymagac czegos do dzialania, albo z kolei sam jest dependency do innego. By odnalezyc co jest potrzebne danemu unitowi dajemy komende **SYSTEMCTL LIST-DEPENDENCIES NAZWA**. Zasadniczo jesli **UNIT** ma w sobie atrybut **CONFLICTS** wtedy nie da sie go uruchomic jesli to drugie jest juz aktywne. Mozna do tego dorzucic **SYSTEMCTL MASK NAZWA_UNITA** by jego przekierowanie dac na **/dev/null** i tym samym nie da sie go wlaczyc w ogole (**UNMASK** to przeciwna komenda).

Niektore z targetow moga byc **IZOLOWANE**. To znaczy uruchomione tylko one i nic poza tym:

- **poweroff.target** - runlevel 0
- **rescue.target** - runlevel 1
- **emergency.target** - runlevel 2
- **multi-user.target** - runlevel 3
- **graphical.target** - runlevel 5
- **reboot.target** - runlevel 6

Generalnie by target mogl sie w ten sposob uruchomic musi miec ustawiony atrybut **AllowIsolate=yes**. Generalnie chodzi o to by w niego wejsc. I dla przykladu mozna w ten sposob przeskoczyc na np. rescue model. Komenda to **SYSTEMCTL ISOLATE NAZWA_TARGETU**

By sie dowiedziec jaki jest defaultowy target w systemie dajemy **SYSTEMCTL GET-DEFAULT**.
By ustawic to **SYSTEMCTL SET-DEFAULT** i nazwa targetu. Warto jest sie upewnic czy sie ma
wszystkie zaleznosci by ten target wskoczyl.

GRUB2 to jest bootloader. Domyslnie ma **INITRAMFS** (filessystem) gdzie sa moduly kernela i
wszystko co jest potrzebne by w ogole system wystartowal. Z definicji sie tego nie rusza.

Generalnie jak trzeba zrobic jakies zmiany w systemie to sie edytuje plik
/ETC/DEFAULT/GRUB. Znalezc tez mozna folder **/ETC/GRUB.D** gdzie leza skomplikowane
pliki shellowe, ktorych lepiej nie ruszac.

Na bazie tych plikow tworzy sie tez **/BOOT/GRUB2/GRUB.CFG** ale tego **SIE NIE**
MODYFIKUJE BO ROBI TO INSTALATOR NOWEGO KERNELA!!

Jedyna opcja jest modyfikacja pliku **/ETC/DEFAULT/GRUB** - w sumie tam najwazniejsza jest
linijka **GRUB_CMDLINE_LINUX** gdzie sie ustawia parametry kernela (parametry **RHGB** i
QUIET najlepiej jest usunac bo wyciszaja komunikaty przy uruchomieniu). Edytujemy plik a
potem dajemy: **GRUB2-MKCONFIG > /BOOT/GRUB2/GRUB.CFG**

YUM LIST KERNEL

RPM -qa | grep kernel-[0-9]

grubby --info=ALL

Te powyzsze komendy pokazuja jakie mamy kernele w systemie, a zwlaszcza to ostatnie
wskaze ten indeks takowego. Znajac indeks mozemy ustawic do ktorego kernela domyslnie ma
sie system startowac:

grub2-set-default NUMER

grubby --set-default-index NUMER

grubby --set-default PELNA_NAZWA_I_LINK_DO_VMLINUZ

Chapter 19 - Troubleshooting boot problems

Tak naprawde to najlepiej jest robic copy&paste z ksiazki.

1. Performing POST: The machine is powered on. From the system firmware, which can be the modern Universal Extended Firmware Interface (UEFI) or the classical Basic Input Output System (BIOS), the Power-On Self-Test (POST) is executed, and the hardware that is required to start the system is initialized.
2. Selecting the bootable device: Either from the UEFI boot firmware or from the Master Boot Record, a bootable device is located.
3. Loading the boot loader: From the bootable device, a boot loader is located. On Red Hat, this is usually GRUB 2.
4. Loading the kernel: The boot loader may present a boot menu to the user, or can be configured to automatically start a default operating system. To load Linux, the kernel is loaded together with the initramfs. The initramfs contains kernel modules for all hardware that is required to boot, as well as the initial scripts required to proceed to the next stage of booting. On RHEL 7, the initramfs contains a complete operational system (which may be used for troubleshooting purposes).
5. Starting /sbin/init: Once the kernel is loaded into memory, the first of all processes is loaded, but still from the initramfs. This is the /sbin/init process, which on Red Hat is linked to systemd. The udev daemon is loaded as well to take care of further hardware initialization. All this is still happening from the initramfs image.
6. Processing initrd.target: The systemd process executes all units from the initrd.target, which prepares a minimal operating environment, where the root file system on disk is mounted on the /sysroot directory. At this point, enough is loaded to pass to the system installation that was written to the hard drive.
7. Switching to the root file system: The system switches to the root file system that is on disk and at this point can load the systemd process from disk as well.
8. Running the default target: Systemd looks for the default target to execute and runs all of its units. In this process, a login screen is presented, and the user can authenticate. Notice that the login prompt can be prompted before all systemd unit files have been loaded successfully. So, seeing a login prompt does not necessarily mean that your server is fully operational yet.

Boot Phase	Configuring It	Fixing It
POST	Hardware configuration (F2, Esc, F10, or another key)	
Replace hardware.		
Selecting the bootable device	BIOS/UEFI configuration or hardware boot menu	
Replace hardware or use rescue system.		

Loading the boot loader `grub2-install` and edits to `/etc/defaults/grub`
GRUB boot prompt and edits to `/etc/defaults/grub`, followed by `grub2-mkconfig`.

Loading the kernel Edits to the GRUB configuration and `/etc/dracut.conf`
GRUB boot prompt and edits to `/etc/defaults/grub`, followed by `grub2-mkconfig`.

Starting /sbin/init Compiled into initramfs init= kernel
boot argument rd.break kernel boot argument.

Processing initrd.target typically required.	Compiled into initramfs	Not
--	-------------------------	-----

```
Switch to the root file system          /etc/fstab
/etc/fstab.
```

Running the default target `/etc/systemd/system/default.target`
 Start the rescue.target as a kernel boot argument.

Kiedy system wstaje i pojawia sie menu GRUB2 to po naciśnięciu **E** pojawia sie mozliwosc edycji opcji, ktora byla pod kursorem (domyslnie kernel do zaladowania). Generalnie da sie tam znalezc sekcje co sie zaczyna od **LINUX16 /VMLINUZ** - to sa parametry z ktorymi normalnie startuje kernel. Nalezy usunac stamtad **QUIET** i **RHGB** bo to powoduje, ze ukrywane sa komunikaty podczas startu. Po usunieciu tego dajemy **CTRL+X** i restartujemy kernela z tymi parametrami. **TO BEDZIE AKCJA JEDNORAZOWA!!!**

By to było persystentne to w poprzednim rozdziale było mowione - trzeba edytować **/ETC/DEFAULT/GRUB** i użyć **GRUB2-MKCONFIG > /BOOT/GRUB2/GRUB.CFG**

Jak masz dalej problemy to **NA KONCU TEJ SAMEJ LINIJKI** mozna dopisac kilka dodatkowych rzeczy:

■ **rd.break** --- This stops the boot procedure while still in the `initramfs` stage. This option is useful if you do not have the root password available. The complete procedure for recovering a missing root password follows later in this chapter.

- **init=/bin/sh or init=/bin/bash** --- This specifies that a shell should be started immediately after loading the kernel and initrd. This is a useful option, but not the best option, because in some cases you'll lose console access or miss other functionality.

- **systemd.unit=emergency.target** --- This enters in a bare minimal mode where a minimal number of systemd units is loaded. It requires a root password. To see that only a very limited number of unit files have been loaded, you can type the **systemctl list-units** command.

- **systemd.unit=rescue.target** --- This starts some more **systemd** units to bring you in a

more complete operational mode. It does require a root password. To see that only a very limited number of unit files have been loaded, you can type the `systemctl list-units` command

Jak jest problem konkretny to można zawsze zbootować z **RESCUE DISC**. Najsensowniejsza opcja tam to **RESCUE U RED HAT SYSTEM**. Domyślnie rescue jest na tyle mądre, że postara się znaleźć twoją instalację z twardego dysku i podmontować ją w **/MNT/SYSIMAGE**. Jednakże by z niej korzystać i zapisane pliki zostały na dysku trzeba w pierwszej kolejności użyć **CHROOT /MNT/SYSIMAGE**. Dopiero wtedy możesz zacząć kombinować i pracować.

Generalnie najczęściej przyczyną, że coś się spieprzyło są błędy w **GRUBie**. Dlatego dobrze jest wiedzieć jak go przeinstalować. Generalnie temat jest prosty - po podmontowaniu istniejącego systemu + chrootowaniu się na niego wystarczy dać komendę **GRUB2-INSTALL SCIEZKA_DO_GLOWNEJ_PARTYCJI**.

Zdarza się, że pieprznie **INITRAMFS**. Wtedy można go też stworzyć od podstaw. Służy do tego komenda **DRACUT** (możliwe, że trzeba dorzucić **--force** by to w ogóle ruszyło). Bez parametrów po prostu tworzy nowy initdisc dla kernela który jest właśnie załadowany. Można też dorzucić kilka specyficznych opcji, które znajdują się w plikach:

- **/usr/lib/dracut/dracut.conf.d/*.conf** contains the system default configuration files.
- **/etc/dracut.conf.d** contains custom dracut configuration files.
- **/etc/dracut.conf** is used as the master configuration file

Jeżeli problem jest z montowaniem systemów plików (bo mogłeś coś zjeść w **/ETC/FSTAB**) to wtedy trzeba wiedzieć co i jak. Najprawdopodobniej podczas bootowania pojawi się komunikat, że potrzebujesz podać hasło roota (**Give root password for maintenance**) - to takie coś rzuca **FCK**. To jest komenda, która sprawdza integralność wszystkich systemów plików. Potem wpiszesz **JOURNAL -XB** by się dowiedzieć co się dzieje (ostatnie komunikaty dostaniesz z journala). Dobrze jest montować system w wersji RO: **MOUNT -o REMOUNT,RW /**

Resetowanie hasła roota. W PIZDU WAZNE MASZ TO UMIEC NA CIEZKIM KACU!!!

1. On system boot, press **e** when the GRUB 2 boot menu is shown.
2. Enter **rd.break** as boot argument to the line that loads the kernel and press **Ctrl+X** to boot with this option.
3. You'll now be dropped at the end of the boot stage where **initramfs** is loaded, just before a mount of the root file system on the directory **/**.
4. Type **mount -o remount,rw /sysroot** to get read/write access to the system image.
5. At this point, make the contents of the **/sysroot** directory your new root directory by typing **chroot /sysroot**.
6. Now you can enter **passwd** and set the new password for the user **root**.
7. Because at this very early boot stage SELinux has not been activated yet, the context type on **/etc/shadow** will be messed up. If you reboot at this point, no one will be able to log in. So you must make sure that the context type is set correctly. To do this, at this point you should load the SELinux policy by using **load_policy -i**.
8. Now you can manually set the correct context type to **/etc/shadow**. To do this, type **chcon -t shadow_t /etc/shadow**.
9. Reboot (type **EXIT** and **EXIT** again). You can now log in with the changed password for user **root**.

CHAPTER 21 - SeLINUX

Element	Use
Policy	A collection of rules that define which source has access to which target.
Source domain	The object that is trying to access a target. Typically a user or a process.
Target domain	The thing that a source domain is trying to access. Typically a file or port.
Context	A security label that is used to categorize objects in SELinux.
Rule	A specific part of the policy that determines which source domain has which access permissions to which target domain.
Labels	Same as context label, defined to determine which source domain has access to which target domain.

System można uruchomić z włączonym **SE** lub wyłączonym. By je zmienić trzeba jednak zrestartować system (bo tak głęboko siedzi). Jeśli jest włączony to z kolei można używać go w dwóch trybach:

- * **ENFORCING MODE** - SeLINUX napierdala i wymusza wszystkie reguły
- * **PERMISSIVE MODE** - Każda działalność jest logowana, ale nic nie jest blokowane.

Ustawianie w jakim trybie się uruchamia system to plik **/ETC/SYSCONFIG/SELINUX** - w pliku tym jest po prostu ustawienie **SELINUX=** i tam się podaje nazwę trybu (lub **DISABLED**).

Komenda by odczytać obecne ustawienie to **GETENFORCE**. By się przełączyć można też użyć komendy **SETENFORCE** (z parametrem **0** daje **PERMISSIVE**, a z **1** **ENFORCING**).

NIE DA SIĘ WŁĄCZAĆ Z DISABLED NA ENABLED W TEN SPOSOB!!!

Kolejna ciekawa komenda jest **SETSTATUS -V** - pokaże co jest zenablewane, a do tego jakie są etykiety i takie tam.

NA EGZAMINIE NA SAM KONIEC SYSTEM MUSI BYĆ ENABLED I W ENFORCING MODE!!!

CONTEXT to jest etykieta, która może być zastosowana do:

- plików i folderów
- portów
- procesów
- użytkowników

Generalnie **RULES** są tworzone tak by połączyć **CONTEXTy** **source** z **target domain**.

By zobaczyć **CONTEXT** dla np. plików czy innych sporo poleceń ma przełącznik **-Z**

(wielkie Z). Dla przykładu:

* **ls -Z /**

* **ps Zaux**

* **netstat -Ztulpen**

Every context label always consists of three different parts:

■ **User:** The user can be recognized by `_u` in the context label; it is set to `system_u` on most directories in Listing 21.3. SELinux users are not the same as Linux users, and they are not important on the RHCSA or RHCE exams.

■ **Role:** The role can be recognized by `_r` in the context label. In advanced SELinux management, specific SELinux users can be assigned permissions to specific SELinux roles. For the RHCSA and RHCE exams, you do not have to know how to configure these.

■ **Type:** The type context can be recognized by `_t` in the context label. Make sure that you know how to work with context types, because they are what it is all about on the exams

Do tworzenia **CONTEXT** używa się polecenia **SEMANAGE** (nie ma tego domyślnie w instalacji więc - **YUM WHATPROVIDES */semanage** i instalujesz - **u mnie się to nazywało policycoreutils-python i costam jeszcze**). Istnieje też **CHCON**, ale w skrócie - **NIE UŻYWAJ!!!**

Jak się chce dopisać **CONTEXT** do istniejącego folderu to w pierwszej kolejności by było dobrze wiedzieć jaki to ma być kontekst. Więc najlepiej dać **LS -Z ADRES** i zobaczyć co tam jest podane (interesuje nas tylko kawałek z suffixem `_t`). Więc by **DOPISAC** kontekst do swojego folderu trzeba robić tak:

semanage fcontext -a -t httpd_sys_content_t "/mydir(/.*)?"

-a to appendowanie

-t to typ

potem nazwa kontekstu

na końcu wyrażenie regularne specyfikujące foldery

To jest jednakże dopisane tylko do **POLICY**, a nie do **FILESYSTEMU**. By zapisać filesystem to dajemy

restorecon -R -v /mydir

Generalnie by odnalezc typ **CONTEXT** to:

- Look at the default environment
- Read the configuration files
- Use **man -k _selinux** to find SELinux-specific man pages for your service - **TO JEST NAJSENSOWNIEJSZE TYLKO NIE JEST ZAINSTALOWANE - PONIZEJ INSTRUKCJA!!**

1. Type **MAN -K _SELINUX**. You'll probably see just one or two man pages.
2. Type **YUM WHATPROVIDES */SEPOLICY**. This shows you the name of the RPM that contains the **sepol** binary, which is **policycoreutils-devel**.
3. Type **YUM -Y INSTALL POLICYCOREUTILS-DEVEL** to install this package.
4. Type **SEPOLICY MANPAGE -A -P /USR/SHARE/MAN/MAN8** to install the man pages.
5. Type **MAN -K _SELINUX**. You'll see no changes yet.
6. Type **MANDB** to update the database that contains names and descriptions of all man pages that are installed.
7. Once the **MANDB** command has finished (this can take a few minutes), type **MAN -K _SELINUX**. You'll now see a long list of man pages scrolling by.
8. Type **MAN -K _SELINUX | GREP HTTP** to find the man page that documents SELinux settings for the httpd service and scroll through it. Notice that it is a complete list of all that you can do with SELinux on the httpd service.

Procedura jak sie appliuj **CONTEXT** do nowego pliku.

- If a new file is created, it inherits the context settings from the parent directory.
- If a file is copied to a directory, this is considered a new file, so it inherits the context settings from the parent directory.
- If a file is moved, or copied while keeping its properties (by using **cp -a**), the original context settings of the file are applied

Generalnie jak sie cos spieprzy w kontekstach to mozna odrevertowac zmiany w filesystemie. By to zrobic uzywamy znanej juz komendy **RESTORECON**. Ba, mozna odrelabelowac caly system. Wtedy walimy **RESTORECON -RV / (-R jest wielkie)**. Mozna tez stworzyc po prostu plik **/.autorelabel** i przy kolejnym reboocie pojdzie autorelabelling (**to może trwać długo więc naprawdę nie jest polecane robić to na examie**).

Tak jak wspominalismy na poczatku mamy **POLICY**, a w nich **RULE**. Tych **RULE** jest w pizdu duzo wiec by ulatwic zarzadzanie nimi powstalo cos takiego jak **BOOLEAN SETTINGS**. By dostac liste wszystkich booleanow dajemy polecenie **GETSEBOOL -A**. Najlepiej po tym przegrepowac to bedzie wiadomo co i jak.

Jest też opcja **SEMANAGE BOOLEAN -L** (to pokaze obecny setting i defaultowy). By zmienić booleana używamy komendy **SETSEBOOL NAZWA_RULE ON|OFF**. To zmienia **RUNTIME**. By zmiany były permanentne trzeba dorzucić flagę **-P** (wielka).

Poprawne zarządzanie SELinuxem jest trudne. Generalnie o tym co się dzieje informuje nas plik **/VAR/LOG/AUDIT/AUDIT**. Zgłoszenia od SEL są logowane z typem **AVC**. Oczywiście jak się to przegląda to można było dostać. Więc instalujemy mądre narzędzie **SEALERT**.

Robi się to **YUM -Y INSTALL SETTROUBLESHOOT-SERVER**, restart serwera by mieć pewność, że wszystko się prawidłowo skonfigurowało i od tej pory wszystko co leci z SEL do auditlogu da się też odczytać z **/VAR/LOG/MESSAGES**. Przykładowa komenda to:

sealert -a /var/log/audit/audit.log

CHAPTER 22 - Configuring a firewall

Generalnie dawniej uzywalo sie **IPTABLES** i w sumie dalej mozna, ale nie jest to zalecane i defaultowe. Tak w ogole firewall jest wbudowany w kernela jako **NETFILTER**. Obecnie w RHELu domyslnym tematem jest **FIREWALLD**. **NIE MOZNA UZYWAC NA JEDNYM SYSTEMIE TEGO I TEGO BO SIE ZRA!!!**

Generalnie **FIREWALLD** ma koncepcje **ZONES**, w ktorych sa **RULE**. Domyslnie sa one aplikowane do przychodzacych pakietow, a do wychodzacych nie jest aplikowane nic. **ZONY** sa aplikowane dla przychodzacego pakietu w takiej kolejnosci:

- * skad przyszedl
- * jaki interfejs sieciowy jest w uzyciu
- * defaultowa zona

Block - Incoming network connections are rejected with an "icmp-host-prohibited" message. Only network connections that were initiated on this system are allowed.

Dmz - For use on computers in the demilitarized zone. Only selected incoming connections are accepted, and limited access to the internal network is allowed.

Drop - Any incoming packets are dropped and there is no reply.

External - For use on external networks with masquerading (Network Address Translation [NAT]) enabled, used especially on routers. Only selected incoming connections are accepted.

Home - For use with home networks. Most computers on the same network are trusted, and only selected incoming connections are accepted.

Internal - For use in internal networks. Most computers on the same network are trusted, and only selected incoming connections are accepted.

Public - For use in public areas. Other computers in the same network are not trusted, and limited connections are accepted. **This is the default zone for all newly created network interfaces.**

trusted - All network connections are accepted.

work - For use in work areas. Most computers on the same network are trusted, and only selected incoming connections are accepted.

FIREWALLD ma kilka swoich uslug (to jest cos innego niz serwisy w **SYSTEMD**)!!! By je zobaczyc dajemy **FIREWALL-CMD --GET-SERVICES**. Pliki konfiguracyjne serwisow znalezc mozna w **/USR/LIB/FIREWALLD/SERVICES** lub w **/ETC/FIREWALLD/SERVICES**.

Do zabawy **FIREWALLD** mamy juz wspomniany command line **FIREWALL-CMD** albo graficzny **FIREWALL-CONFIG**. **Generalnie nalezy zawsze pamietac ze wszystkie zmiany sa robione w**

pamieci i nalezy je potem zapisac na dysk!!!

FIREWALL-CMD w opcjach:

--get-zones	Lists all available zones
--get-default-zone default zone	Shows the zone currently set as default zone
--set-default-zone=<ZONE>	Changes the default zone
--get-services	Shows all available services
--list-services	Shows services currently in use
--add-service=<service-name> [--zone=<ZONE>] zone or the zone that is specified	Adds a service to the current default zone or the zone that is specified
--remove-service=<service-name> configuration	Removes a service from the configuration
--list-all [--zone=<ZONE>]	Lists all configurations in a zone
--add-port=<port/protocol> [--zone=<ZONE>]	Adds a port and protocol
--remove-port=<port/protocol> [--zone=<ZONE>]	Removes a port from the configuration
--add-interface=<INTERFACE> [--zone=<ZONE>] or a specific zone that is specified	Adds an interface to the default zone or a specific zone that is specified
--remove-interface=<INTERFACE> [--zone=<ZONE>]	Removes an interface from a specific zone
--add-source=<ipaddress/netmask> [--zone=<ZONE>]	Adds a specific IP address
--remove-source=<ipaddress/netmask> [--zone=<ZONE>]	Removes an IP address from the configuration
--permanent	Writes configuration to disk and not to run-time
--reload	Reloads the on-disk configuration

CHAPTER - 23 Mounting external filesystems

Z LA: By wystawic cos po NFSie (czyli stworzyc serwer bo tego w ksiazce w ogole nie bylo) to robimy tak:

1. Tworzymy folder ktory udostepniamy (/nfs)
2. Dajemy mu 777 by mogli go ludzie w kazda strone ogarniac
3. tworzymy plik **/ETC/EXPORTS** i walimy tam: `/nfs *(rw)`
4. Uruchamiamy serwisy **systemctl start {nfs-server, rpcbind, rpc-statd, nfs-idmapd}**
5. tak jak jest **mount -a** dla zwyklych montowan to dla zdalnych jest **exportfs -a**
6. upewniamy sie, ze temat jest eksportowany: **showmount -e localhost** (jak nie jest to znaczy, że spieprzyliśmy coś na firewallu)

Generalnie mozna sobie montowac rzeczy z **NFS** (NetworkFileSystem). Montowanie tematu **MOUNT ZDALNY_ZASOB LOCAL**. By uzywac tego trzeba miec zainstalowane **RPCBIND** (to jest czesc **nfs-utils**)

Domyslna wersja **NFSa** na RHELu 7 jest **NFS 4**. Ta wersja wspiera cos takiego jak **PSEUDO-MOUNT** co umozliwia dla przykladu jak sa trzy rozne zasoby na zdalnym serwerze to bedzie **MOUNT ZDALNY_ZASOB:/ LOCAL_ZASOB**

Security jest problemem bo domyslnie **NFS CLIENT** bedzie sie laczyl z serwerem NFS uzywajac UIDa. I jesli te same UIDy sa na dwuch serwerach wtedy user z klienta bedzie mial dostep do plikow tego drugiego usera. W ksiazce mowia, ze najlepiej zatem uzywac **LDAPa** czy cos (centralny rejestr userow). Ogarnianie w jaki sposob security dziala jest ustawiane podczas montowania zasobu uzywajac flagi SEC z ponizszymi parametrami:

- none** - Access to files is anonymous and mapped to the UID and GID of the user
- nfsnobody**. Writes are permitted only if **nfsnobody** has write access.
- sys** - File access is based on UID and GID values on the client and the matching of these to the IDs used on the server. **This is the default setting.**
- krb5** - Client users must prove identity using Kerberos. After that, Linux permissions apply.
- krb5i** - Like krb5, but adds the guarantee that data in a request has not been tampered with.
- krb5p** - Like krb5i, but adds encryption to each request. This has the highest level of protection, but does have a negative impact on performance

By dzialac z **KERBEROSEM** potrzeba dwoch rzeczy (to jest poza scopem RHCSA, jak cos to dadza ci ten plik i trzeba go dograc do odpowiedniego folderu i miec ten serwis z dolu + montowac z odpowiednim ustawieniem security):

- * ustawic plik **/ETC/KRB5.KEYTAB** i tam cos poustawiac
- * **NFS-SECURE** service musi byc uruchomiona na kliencie

Jak serwer oferuje inna wersje **NFSa** niz ma klient to polaczenie automatycznie sie na nia przelacza. Mozna tez podczas montowania uzyc flagi **NFSVERS=X** gdzie X to numer wersji.

Zasoby, ktore mozna podmontowac rozkminia sie na 3 sposoby:

- * przy wersji 4 robi sie **ROOT MOUNT**
 - * ponestastowc **NETSTAT -AN | GREP NFS.SERVER.IP:PORT**
 - * uzyc **SHOWMOUNT -E NFSSERVER** (ma jakies problemy z security - jak jest za firewallem)
- + trzeba zainstalowac **NFS-UTILS** by z tego korzystac wpierv

Montowanie jest proste:

mount -t nfs SERWER:/ZASOB /GDZIE/MONTOWAC/LOKALNIE

mount -t nfs 192.168.10.1:/nfs /mnt/nfs/ PAMIETAJ BY UTWORZYC FOLDER!!

Istnieje tez **SMB** lub **CIFS** (to jest poddialekt **SMB**) - generalnie to protokoly by sharowac dane miedzy roznymi systemami.

By uzywac tego trzeba na kliencie zainstalowac **CIFS-UTILS** oraz **SAMBA-CLIENT**.

Do tego trzeba dodac to co uzywamy do wyjatkow firewalla:

firewall-cmd --add-service samba-client --permanent; firewall-cmd --reload

By wylistowac zasoby dajemy **SMBCLIENT -L (wielka) IP_ZASOBU_DO_LISTOWANIA** - mozemy tez uzyc **-Unazwa_usera** (wielka) by specyfikowac konkretnego usera do listowania. Jest tez **NET SHARE -L** ale dziwnie to jakos opisane.

Montowanie po **CIFSie** wyglada prosto: **MOUNT -T CIFS -O USER=GUEST**

//192.168.122.200/data /mnt

To zamontuje w read only modzie no bo user to guest. By cos robic z zasobem trzeba logowac sie jako user utworzony na serwerze **SAMBY**. Jak sie laczy userem z serwera **SAMBy** to podczas montowania system spyta sie o haslo.

Oczywiscie montowanie tego recznie podobnie jak normalnie jest slabe. Zatem trzeba edytowac **/ETC/FSTAB** by podczas bootowania systemu - albo jest tez opcja z uzyciem jakiegos serwisu **AUTOFS** (bedzie potem).

/ETC/FSTAB

```
server1:/share /nfs/mount/point nfs _netdev,x-systemd.automount,sync 0 0
```

- In the **first** column, you need to specify the server and share name. Use a colon after the name of the server to identify the mount as an NFS share.
- The **second** column has the file system where you want to do the mount; this is not different from any regular mount.
- The **third** column contains the NFS file system type.
- The **fourth** column that is used to specify mount options includes the sync option. This ensures that modified files are committed to the remote file system immediately and are not placed in write buffers first (which would increase the risk of data getting lost)
- The **fifth** column contains a zero, which means that no backup support through the dump utility is requested.
- The **sixth** column also contains a zero, to indicate that no **fsck** has to be performed on this file system while booting to check the integrity of the file system. The integrity of the file system would need to be checked on the server, not on the client

TE OPCJE _NETDEV I X-SYSTEMD.AUTOMOUNT SA W PIZDU WAZNE!!!

Problem z montowaniem to jak zawsze security. Jak cos mozna wyspecyfikowac usera i password w **/ETC/FSTAB**, ale to generalnie skonczy sie puszczeniem hasla w plaintekscie. Drugim sensownym pomyslem jest uzycie **CREDENTIALS FILE**. By nikt niepowolany do niego nie mial dostepu wrzucic go najlepiej do folderu roota, zrobic go ownerem i group ownerem, dac uprawnienia 600 i tyle. Plik moze wygladac tak:

```
username=linda  
password=secret  
domain=mydomain
```

Montujac za pomoca MOUNT lokacje pliku specyfikuje sie flaga **-O SCIEZKA_DO_PLIKU**, w przypadku **FSTABa** robi sie to tak:

```
//server1/data /mnt/data cifs _netdev,x-systemd.automount,credentials=/root/creds 0 0
```

Mozna tez robic montowanie **NFS** i **SMB** za pomoca **AUTOFS**. To jest serwis, ktory montuje zasoby, ale nie podczas startu systemu tylko kiedy sa potrzebne. Bonus jest taki, ze ten serwis pracuje w **USER SPACE** i nie potrzebuje praw roota.

Wpierw trzeba zainstalowac **AUTOFS**, a potem trzeba utworzyc master plik konfiguracyjny w **/ETC/AUTO.MASTER.D** - nazwa jest dowolna, musi sie tylko konczyc na AUTOFS. Ponizej cwiczenie:

1. Type **yum install -y autofs** to install the autofs package.
2. Create the master map file that contains further instructions that tell the autofs service how to automount the remote file systems. Type **vim /etc/auto.master.d/demo.autofs** to create and open the file.
3. Add the master map entry for indirect mapped mounts by adding the following line:

/shares /etc/auto.demo

This uses the **/shares** directory as the starting point for all indirect mounts. The **auto.demo** file is referred to as the file that contains the instructions that further complete the automount.

4. In the same file, include the following line for directly mapped mount points:

/- /etc/auto.direct

Direct mounts always have **/-** as the starting point for the direct mounts in the master map file. Further instructions on how to perform the mount are in the **auto.direct** file. Notice that the names of these secondary files do not really matter. The only requirement is that they need to be created in the path that is indicated.

5. In the indirect mount file **auto.demo**, include the following line to mount **labipa:/data** on the directory **/shares/data** using the **rw** and **sync** NFS mount options:

data -rw,sync labipa:/data

Notice that the first field as a relative directory name contains the name of the mount point, which is followed by the mount options, which are followed by the name of the **NFS** server and share on that server. Notice that in this indirect mount, the **/shares** directory as well as its data subdirectory, will be automatically created by automount at the moment that the indicated file system is mounted.

6. Now create the direct mounts configuration in the file **/etc/auto.direct**. Give this file the following contents:

/mnt -rw,sync labipa:/home

Notice that in a direct mount, the directory that is used as the mount point should already exist before the automount can be done.

7. Type **systemctl enable autofs; systemctl start autofs** to start the autofs service.

8. At this point, you can test the automount configuration. Type **cd /shares**. This should automatically do the automount of the **labipa:/data** share in **/shares/data**. Now type **cd /mnt**. This should automatically mount the labipa:/home share on the /mnt directory

Montowanie SMB-share musi byc plus 2 rzeczy:

- * link do credentials pliku musi byc absolutny
- * nazwa montowanego zasobu musi sie zaczynac od :

BARDZO WAZNA JEST KOMENDA MOUNT -i -> JEDZIE PO FSTABIE I MONTUJE WG WPISANYCH TAM WARTOSCI

CHAPTER 24 - Configuring Time Services

Hardware clock	- The hardware clock that resides on the main card of a computer system
Real-time clock	- Same as the hardware clock
System time	- The time that is maintained by the operating system
Software clock	- Similar to system time
Universal time coordinated	- A worldwide standard time
Daylight savings time	- Calculation that is made to change time automatically when daylight savings time changes occur
Local time	- The time that corresponds to the time in the current time zone

Zarządzanie software clockiem jest proste. Albo się podłącza atomowy zegar do serwera (dobre dla datacenters), albo używa **NTP**. **NTP** to prosty serwis co po prostu ma zaszyta w pliku **/ETC/CHRONY.CONF** listę serwerów z którymi się powinien połączyć by dostać czas. Jedyne co trzeba zrobić to po prostu **NTP** włączyć komendą **TIMEDATECTL SET-NTP 1**.

date	- Manages local time
hwclock	- Manages hardware time
timedatectl	- Developed to manage all aspects of time on RHEL 7

By przerobić **EPOCH** na czytelny dla usera czas do komendy **DATE** dorzucamy flagę **--DATE** i prefix w postaci malpy: **date --date '@1420987251'**

Komenda **DATE** konfiguruje system time. Można jej też używać do pokazywania obecnego czasu w różnych formatach.

■ date	- Shows the current system time
■ date +%d-%m-%y	- Shows the current system day of month, month, and year
■ date -s 16:03	- Sets the current time to 3 minutes past 4 p.m

Hardware time to komenda **HWCLOCK**.

■ hwclock -c	- shows the difference between hardware time and system time. The
---------------------	---

output of this command is refreshed every 10 seconds.

- **hwclock --systohc** - synchronizes current system time to the hardware clock.
- **hwclock --hctosys** - synchronizes current hardware time to the system clock

TIMEDATECTL jest uberkomenda (uzywa pod spodem procesu **CHRONYD**, wczesniej uzywalo sie **NTPD**, ale teraz lepiej **CHRONYD**) i ma docelowo zastapic wszystkie inne:

- list-timezone** - Show a list of all time zones
- set-local-rtc [0|1]** - Control whether RTC (the Real Time Clock—this normally refers to the hardware clock) is in local time
- set-ntp [0|1]** - Control whether NTP is enabled
- set-timezone**

Linuxy miedzy soba wymieniaja sie czasem w **UTC**. Ale by userom bylo wygodniej uzywa sie **TIMEZONE**. Ustawia sie je na cztery sposoby:

- Use the **SYSTEM-CONFIG-DATE** utility as discussed in the next section of this chapter.
- Go to the directory **/usr/share/zoneinfo**. In this directory, you'll find different subdirectories containing files for each of the time zones that has been defined. To set the local time zone on a server, you can create a symbolic link with the name **/etc/localtime** to the time zone file that is involved. If you want to set local time to Los Angeles time, for instance, use **ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime**.
- Use the **TZSELECT** utility.
- Use **TIMEDATECTL** to set the time zone information