**Кратка история за Linux**

Една от първите стабилни операционни системи е UNIX, която излиза през 1969г. През годините е имала различни промени. За момента се ползва като операционна система за големи корпорации и правителства. Съществуват и нейни разновидности като HP-UX.

Персоналните компютри навлизат към края на 70-те и особено в началото на 80-те години. Най-привлекателната част на копютрите по онова време е GUI (графичен потребителски интерфейс), което не е било особено надеждно. За персоналните компютри с GUI се отличават: Xerox (1981), а след това и Apple(1983), но нямат успех тъй като са много скъпи (20-30 хил. $). Най-голям успех е постигнат през 1984 с илизането на Mac OS X за Macintosh.

Поначало буквата X идва от Unix и с нея се означава графиката (т.е. GUI). В последствие се използва за означаване на графиката и при всички UNIX базирани системи. Важно е да се знае, че при повечето UNIX системи графиката може да се заобикаля, например като се премине в терминален режим.

През средата на 80-те години не само компютрите са били сравнително скъпи, но и софтуера за тях. По същото време възниква идеята за безплатен софтер и за безплатна OS. Тази идея е въведена от Ричард Столман, който създава и проекта GNU през 1983г. Абревиатурата GNU произлиза от „GNU is not UNIX“ и се използва за open source операционна система. Обикновенно повечето абревиатури за open source нещата са „рекурсивни“. Та за Ричард Столман – той не успява да създаде такава операционна система. Той написва първия open source компилатор за C++, който се нарича gcc и се ползва и до днес (естествено с новите си версии).

През 1989г. Столман създава и лиценза GPL (General Public License) за програмите който ще се използват за GNU.

По същото време в университета в Хелзинки се е ползвала операционната система MINIX – която е била орязана версия на UNIX, но с отворен код с цел изучаване от студентите. Тогва студентът Линус Торвалдс започва личен проект за операционна система. Към проекта се присъединяват и други студенти. Така през 1991 година възниква първата версия на ядрото Linux (Linus + Unix). Заедно с някои от инструментите и програмите създадени към проекта GNU се сформира операционната система GNU Linux, която е изцяло под лиценза GPL. Голяма роля за създаването на Linux по това време изиграва и Stolman. Към проекта се присъединяват лавинообразно и много други програмисти.

Създават се и различни общности (communities) поддържащи различни възгледи за обвивката на ядрото. Важно е да се знае, че под Linux трябва да се разбира само ядрото на операционната система. Всички останали програми и инструменти за работа (терминал, графична система и др.) с него се наричат обвивка. Възникават и различни дистрибуции:

Wiki -> Linux distribution

В началото (1991-1992) ядрото като обвивка е имало само терминала. По същото време на пазара се появява Windows 3.11 и съответно Linux с нищо не може да се конкурира. Затова се налага създаването на графична среда. За целта участват всички общности поддържащи Linux ядрото.

Първата графична среда е KDE [keidi: i:] която излиза 1996 година. Написана е чрез платформено независимата библиотека Qt. Чрез Qt са написани програми като Skype, Google Maps и др. Като абривиатура KDE означава *K Desktop Environment.* Но Qt има проблем с лиценза и съответно всички програми, включително и KDE са засегнати от това.

Поради тази причина общностите са принудени и написват open source библиотека GTK+. Чрез нея е написана графичната система GNOME [gnoum], която излиза през 1998, официално 1999г.

Малко след това и Qt е обявена под лиценза GPL и съответно KDE става напълно беплатна.

Затова след 2000-та година повечето дистрибуции вървят с два варианта за избор на графична система KDE или GNOME. За момента има разработени и много други (>10) графични системи.

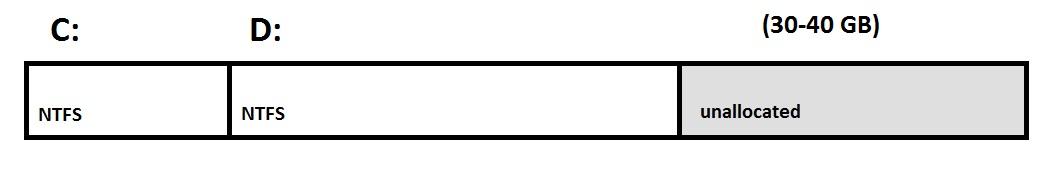
Важно е да се знае, че за всяка графична система има и различен набор от програми. Например за KDE програмите често започват или съдържат буквата K – Kwrite, KAmarok, Ark и др.

Някои дистрибуции плътно се придържат само към една от тези системи. Например Debian базираните (Ubuntu, Linux Mint и др.) са с GNOME с цел никога в историята си да не са имали нещо платено по тях.

Други дистрибуции като Red Hat и Suse са предимно с KDE. Тези две дистрибуции стават и първите комерсиални (retail) ( red hat е първа): Red Hat Enterprice и Suse Enterprice. Но тъй като ползват open source ядро са „задължени“ да имат и open source варианти. Те са съответно: Fedora Core и openSuse.

**Инсталация на Linux:**

Когато се инасталира Linux на отделен хард диск съответнно е най-безпроблемно. Когато на диска има друга ОС – например Windows трябва да се съобразят някои особености. Първо трябва да бяде инсталирана Windows системата, така Linux OS ще я разпознае, ще може да индексира за да може след това да създаде меню чрез което да бъде стартирана и тя. Програмата която зарежда Линукс се нарича GRUB (**GNU GRand Unified Bootloader**), и тя ще конструира меню. Освен това е препоръчително (за да няма проблем с форматиране) да бъде отделен дял от хард диска без форматиране (unformatted или unallocated). Разделянето може да стане и през Windows чрез инструмента Disk Managment.



По този начин инсталатора ще разпознае този диск и ще продължи инсталацията върху празния диск. За файлова система в Linux се ползва Ext3FS (стара е) и Ext4FS.

Обикновено Линукс отрива отделните носители с информация (хардискове, флаш памети и др) като устройства с имена sda, adb, sdc и т.н. След името на устройството стои номера на дяла. За MBR номерата от 1 до 4 са запазени за главни дялове, а от 5 нагоре за логически. При GPT номерацията е от 1 нагоре.

Инсталацията може да се извърши от DVD или bootable Flash Drive.

След инсталация Линукс ще е създал допълнителни дялове:

/ - главна или коренова (root dir) директория. Основна директория за всички останали.

/swap - това е виртуалната памет. Отделена е на отделен дял без форматиране.

/home – домашна директория за всички локални потребители

/root – домашна директория за потребителя root (суперпотребителя)

Основните потребители в Linux са:

root - това е запазено име за потребителя с пълни права за всяка линукс система. Нарича се още

суперпотребител (superuser). В терминала root е с промпт #.

Всички останали потребители са локални и са с ограничени права. В терминала са с промт $. За да имат тези потребители определени права могат да се добавят към групата на администраторите sudo. По време на инсталацията, на една от нейните стъпки се създава и един локален потребител потребител на който се задва име и парола. Този потребител е и към групата sudo.

При Ubuntu потребителя root съществува, но в началото няма парола. Такава може да бъде добавена от създадения локален потребител чрез следната команда в терминала:

sudo ˽ passwd

!Но няма да я добавяме!

Kомандата sudo се използва за изпълнение на администраторски комнади от локален потребител, стига той да е в групата sudo (да има sudo привилегии). Тази команда е силно препоръчителна и чрез нея се избягва преминаване като потребител root. Обикновено в Linux логване като root се прави изключително рядко и се избягва. Поради това в Ubuntu няма парола за root, а се използват привилегиите на sudo. Командата passwd служи за промяна на парола на даден потребител. Когато не е посочен такъв потребител като аргумент се подразбира root. За да може обикновенния потребител да влезне като root може да се използва командата su (substitute user):

sudo ˽ su˽-

Тирето зарежда променливите на средата на root, включително и преминаване в неговата домашна директория (т.е. /root). За да преминем към локалния потребител използваме само :

su˽-l˽localUser

Опцията -l е за да променим и домашната директория на новия user.

Друг вариант за илизане от сесия е чрез Ctrl+D (logout).

След инсталация може да остане Линукс като система по подразбиране за зареждане. Това се променя от системата за начално зареждане GRUB.

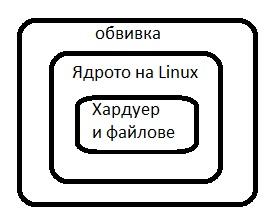
-> Google search: How do I change the GRUB boot order?

Обикновенно се променя конфигурационния файл на grub, който се намира в /etc.

================================================================

**1. Обвивката (shell) в Linux**

В самото сърце на операционната система Linux е разположена серия от машинни инструкции, която се нарича ядро (kernel) – това е техническа програма, която не е много подходяща за директно използване от потребителя, тъй като е създадена да комуникира предимно на системно ниво. За тази цел служи така наречената обвивка на ядрото (shell). Чрез нея потребителите могат да комуникират в разбираема форма с ядрото на операционната, основно чрез инструкции. Обвивката случи да превежда инструкциите от командния ред до машинен код. Т.е. още се нарича команден интерпретатор. На фигурата по долу е показана взаимовръзката между ядрото на Linux, хардуера и програмите.



След като ядрото се зареди, всички елементи от централния кръг се управляват от него; достъпът до всички хардуерни устройства и файлове се осъществява чрез ядрото. Областта, дефинирана от средния кръг (където се разполага ядрото), често се нарича пространство на ядрото. Външният кръг е пространството на потребителя; обвивката, всички програми, стартирани от нея, както и демоните, оперират в пространството на потребителя. Тези програми осъществяват достъп до хардуера и другите файлове (ресурсите) само чрез ядрото – предоставянето или отказването на достъпа се определя на базата на позовленията.

Обвивката не е нищо повече от програма, която се изпълнява в пространството на потребителя и ви предоставя интерактивен интерфейс към системата (команден промт) за въвеждане на данни или за задаване на програми, които да се стартират. Подразбиращата се обвивка за повечето Linux дистрибуции е bash (Bourne Again SHell).

BASH е актуализирана версия на първоначалната обвивка Bourne в операционната система Unix. Обвивката разпознава голям брой команди, които се съдържат в директориите /bin и /sbin

**Работа с терминала:**

В Linux освен стандартната подразбираща се обвивка BASH има и 6 виртуални терминала.

Те могат да се отворят с комбинацията

Ctrl+Alt + ( от F1 до F6 ) съответно за терминалите от tty1 до tty6.

Ctrl+Alt+F7 се отваря графичната система.

През виртуалните терминали графичната система може да биде спирана и след това стартирана, т.е. в Линукс тя може да бъде заобикаляна.

Терминала в Ubuntu може да се отвори с комбинацията Ctrl+Alt+T.

Стандартният формат на конзолните команди в Unix е:

***команда*** *[-опции] [аргумент]*

Разделителя е интервала (˽), като командите се пишат с малка буква. Linux е чуствителен към регистъра и прави разлика между малки и големи букви. Опциите се задават с тире ( - ) преди тях.

Примери:

whoami - показва потребителското име на текущия потребител

pwd – извежда името на текущата директория (print working directory)

ls - извежда списък (съдържанието) на текущата директория // list

ls –a - ls е стартирана с една опция а (all) ще изведе всички, дори и скритите файлове и директории

ls –al - ls е стартиран с две опции , l (long) ще изведе подробна информация за файла или директорията

ls –al /bin - ще изведе всички файлове и директории в подробен формат от дир. /bin

Тъй като командите са твърде много не е нужно да се помнят всичките им опции. Ще покажем как да разглеждаме опциите за всички команди. Когато напишем само началото на команда, например pw и когато натиснем два пъти табулация ще ни изведе всички команди които започват с pw. Ако не сме въвели нищо и натиснем два пъти Tab, разбира се ще изведе всички команди от обвивката като списък. Тъй като са много ще ни попита дали да ги изведе. За всяка команда важи извеждането на помощна информация чрез дописването на - -help след нея.

ls --help - извежда кратка информация за опциите на дадена команда

Можем да отворим и по подробна информация за всяка команда чрез така наречените man (manual - ръководство) страници или чрез info страниците. Отваря се в целия терминал, като прелистването става чрез Page Up и Page Down . Излиза се чрез Q от клавиатурата:

man ls - man страницата за командата ls

man man -man страница за самата команда man

info ls - info страницата за команда ls

Изпълнението на команда може да се прекрати чрез комбинацията Ctr+C или Ctrl+Z.

Други команди :

clear - изчиства екрана

date - извежда датата

cal - извежда календар

uptime – колко време е работила системата след последния рестарт

За смяна на потребителя се използва командата substitute user:

su ˽ otherUser

su ˽ -l ˽otherUser -- сменя потребителя + домашната му директория

su ˽ -- когато е без аргумент искаме да сменим с root

================================================================

**2. Инсталиране на програми:**

За Debian дистрибуциите се използват .dpkg пакети и често се използват командите apt-get:

Например за инсталация на програмата Gparted:

sudo˽ apt-get˽ install˽ gparted

а за деинсталиране:

sudo˽ apt-get˽ remove˽ gparted

често се използва и вариант с purge за деинсталиране , чрез който се премахват и конфигурационните файлове:

sudo˽ apt-get˽ purge˽ <package\_name>

за обновяване с нови пакети:

sudo˽apt-get˽update

за upgrade на системата

sudo˽apt-get˽upgrade

За други дистрибуции се използват други средства. Например за Red Hat и Suse се използват .rpm пакети заедно с прилежащите им среди за инсталация:

yum - за red hat

yast - за suse

==========================================================

**3. Файлова система на Linux**

Google search: Linux Filesystem Hierarchy Standard (FHS)

--------------------------------------------------------------------------------------------

Когато стартираме терминала той се отваря като текущата директория ще бъде домашната директория на потребителя с който сме влезнали.

/root - домашната директория за суперадминистратора

/home/local - домашната директория на потребителя local

В терминала можем да сменим текущия потребител с командата su (substitute user). Ако я стартираме без опции тя ще запази работната директория от предходния потребител. Затова е препоръчително да е ползва с опция –l , чрез която променяме и работната директория.

su˽ –l - влизаме като администратор, като отиваме и в домашната му директория /root, ще ни иска да въведем администраторската парола. Паролите не се изписват в терминала, когато я въвеждаме, дори без \*.

su˽ local - променяме в потребителя local като работната директория остава същата като на предходния потребител. Администратора може свободно да преминава като друг потребител.

За Ubuntu влизането като суперпотребител става чрез:

sudo˽su˽-

Излизане става с Ctrl+D

------------------------------------------------------------------------------------------

Обхождане по директориите:

cd˽ [път] – променя работната директория change directory

cd - ако е без параметри ще отиде в домашната директория

За директориите имаме служебни символи:

**.**  – указва текущата директория

**..** – указва предходната директория (родителска)

**/**  - главна директория

**~** -домашна директория

- -предишната работна директория

Например чрез cd .. ще отидем в предходната директория, а с cd ~ ще отидем в домашната директория. Разделителя интервал е задължителен. Когато се обхождат директориите имаме два начина за описание на пътя до даден файл или директория:

- абсолютен; той започва винаги от главната директория /

- относителен; започва от текущата директория в която се намираме

Пример: ако текущата директория е /home и искаме да отидем до поддиректорията local в нея, ще стане чрез два начина :

- абсолютен : cd˽ /home/local

-относителен cd˽ local

Вижда се, че с относителния път в този случай е по кратко, но ако искаме чрез относителен път да отидем до /etc ако в момента сме в /home/local то това ще стане така:

cd ../../etc

а чрез абсолютния:

cd /etc

----------------------------------------------------------------------------------------

стартираме листа :

ls –l за да видим пълната информация за всеки един фаил

Обикновено терминала ги оцветява, като директориите са в синьо, изпълнимите файлове в зелено, връзките (link) в светло зелено и т.н. Подробната информация за файла има следния вид:

-|9 символа|бр.файлове|собственик|група|размер|дата на последна мод.|име

Първият символ означава вида на файла:

- обикновен файл

d директория

l символна връзка – symbolic link (shortcut)

Ако първия символ е друг файлът е служебен (S сокет, b блоково устройство и др.)

9-те символа са за правата на достъп.

Пример за файла /etc/passwd:

-rw-r--r—1 root root 695 Dec 7 12:48 passwd

-----------------------------------------------------------------------------------------------

Създаване на директории става както е посочено по долу. Директориите също се разглеждат като файлове, само че те съдържат информация за други файлове. Препоръчително е имената на директориите да не съдържат интервал, тъй като когато я изписваме като аргумент на команда интервала ще се чете като разделител. Ако все пак държим да има интервал, той трябва да се скрива(escape) с наклонена черта : my\ dir . Също така не е желателно да се създават файлове и директории в главната / директория.

mkdir mydir - създава директория с името mydir

rmdir mydir - изтрива директорията mydir (само ако е празна)

създаваме пак директорията mydir и влизаме в нея :

cd mydir

3.1 Файлове – създаване:

Всички файлове които започват с точка ( **. )** са скрити. Имената на файловете е препоръчително да не съдържат интервали.

touch f1 - създава празен файл с името f1

Друг начин за създаване на файл е като се използва знака за насочване на поток ( >) и в комбинация с някоя команда, която извежда резултат:

date >f2 - ще отпечата датата като създаде файл с името f2

cal >f3 - същото като date, само че за календара

echo ˽ “helloworld”˽ > ˽ f4

За прочитането на файл най-често се използва командата cat (concatenation), която конкатенира низа прочетен от файла с конзолата

cat f3 - прочита файла f3 и го отпечатва в конзолата

Чрез нея също може да се създаде файл, като се използва отново знака (>) и този път ни се дава възможност да пишем във файла. Спирането става след последния Enter изпълним комбинацията Ctrl+C.

cat >file - създаваме файл с името file като пишем в него.

cat >. hiddenfile - създаваме скрит файл. Той ще бъде видим чрез командата ls -l

---------------------------------------------------------------------------------------

3.2. Копиране на файл

cp <файл> <директория> - копира файл в посочената директорията

cp file1 file2 - копира един файл в друга директория с друго име

mv <файл> <директория> - премества файл в посочена директория (cut)

mv file1 newfile1 - ще преименува file1 в newfile1

Метазнаци:

За определени цели се използват служебни символи които могат да заместват други:

\* - замества произволен брой знаци

? – отговаря на единствен знак

Пр.

b\* - всички думи които започват с b

b??? – всички четирибуквени думи които започват с b

зад1. Създайте в домашната директория, поддиректория с името newdir, копирайте в нея всички файлове от /bin, чийто имена започват с p и покажете резултата:

cd ~

mkdir newdir

cd newdir

cp /bin/p\* .

ls –l

в случая ( . ) означава текущата директория в която се намираме.

------------------------------------------------------------------------------------------------

3.3 Връзки към файловете:

Създават се чрез командата ln и биват два вида: твърди и меки (символни). Твърдите връзки съдържат системния адрес на който се намира даден файл върху твърдия диск. Могат да сочат към файл от същата файлова система. Работата с твърдата връзка е еквивалентна с работа със самия файл.

ln file1 hlink1 - създава твърда връзка с името hlink1 към файла file1

cat hlink1 - изчита връзката; еквивалентно на отваряне на файла

ls –li - опцията –i показва твърдите връзки. Чрез нея се извеждат номерата на i-възела на всеки файл

Символните връзки съхранява пътя до файла във вид на URL. Могат да сочат към локални или отдалечени файлове. Символните връзки започват с l , когато стартираме ls –l. Символните връзки са оцветени в светло зелено.

ln –s file2 slink2 - създава символна връзка с името slink2 към файла file2

readlink slink2 - показва пътя до файла

cat slink2 - отваряме файла чрез символна връзка slink2

---------------------------------------------------------------------------------------------------

3.4. Изтриване на файлове и директории

rmdir <директория> - изтрива празна директория

rm <фаил> - изтрива фаил

rm –r <директория> - изтрива директория рекурсивно, т.е. с всички поддриректории и файлове

rm –ri <директория> - изтрива с питане за всеки файл от директорията

rm –rf <директория> - изтрива пълна директория без питане !!

Ако искаме да изтрием фаил с името –f може да ни създаде затруднение, т.к. rm –f ще се интерпретира като нормална команда и –f ще бъде флаг на нея. Решението е да се използва по-пълното име на файла (**./-f**) или да се използва аргумента -- на командата rm, който и казва, че всичко след него са имена на файлове, а не опции (**rm -- -f**)

touch˽\ ˽ ’-f’

rm ˽ --˽\ ˽ ’-f’

-----------------------------------------------------------------------------------------------------

3.5. Работа със текст в обвивката:

cat /etc/passwd

//Файлът passwd е кунфигурационен файл и съдържа всички потребители в системата (повечето служебни). Използва се следния запис, където всеки ред е потребител и параметрите са разделени с двуеточие:

username : x : UID : GID : full name and description : home dir : shell

x- това поле не се използва, по рано са се съхранявали хешираните пароли (сега са във файла shadows)

UID - User ID

GID - Group ID

shell - подразбиращата се обвивка, най-често за реалните потребители е /bin/bash

//

head -5 /etc/passwd

tail -5 /etc/passwd

nl /etc/passwd - номерира редовете (number line)

grep root /etc/passwd – търси по шаблон (например думата root)

grep [M] /etc/passwd -връща всички редове които съдържат главно F

за по сложни шаблони се използват регулярни изрази (RegEx)

wc –w /etc/passwd -- преброява колко думи (разделителя е интервал) има в дадения файл

wc –c /etc/passwd Символите

wc –l /etc/passwd Редовете

find <име на дир> -type f –name <име на файл> -print

търси дали даден файл(f) или директория(d) се срещат като се почне от начална <име на дир>

find /var –type f –name messages -print

Но ще ни трябват права за да претърсваме в някои директории. Затова може да пробваме:

find /etc –type f –name passwd -print

**3.6.** Комбиниране на команди

pwd;ls;whoami -- групираме

Функции OR-IF и AND- IF

or –if:

ls || pwd -- ще се изпълни само първата

lsssss||pwd --

and –if:

ls && pwd -- и двете

lssssss&&pwd -- нито една

------------------------------------------------------------

Използване на конвейр |

При конвейра (pipe, pipeling) резултата от една команда се подава като вход на следваща команда.

ls | wc –w - Резултата от командата ls е списък с имена на файлове и директории, който се подава на wc -w, която от своя страна ще преброи тези имена и това ще е резултата

ls | sort

ls | sort –r

cat /var/log/syslog | less - ако е много дълъг файла ще го изведе на няколко екрана

Командата less разбива резултата на няколко екрана. Изхода от нея е с Q от клавиатурата.

// Файлът syslog съдържа всички системни логове за предупреждения и грешки.

ако нямаме позволения влизаме като администратор или изпълняваме с:

sudo cat /var/log/syslog | less

============================================================

**Позволения за достъп до файловете (File Permissions):**

За достъп се разпознават следните потребители:

* собственик - този който е създал файла
* група на собственика (потребителите който принадлежат на групата на собственика)
* всички останали

Потребителя **root** има пълни права.

Правата са за четене, запис и изпълнение:

|  |  |
| --- | --- |
| **Означение** | **Право** |
| r | read |
| w | write |
| x | execute |
| - | липсва |

Правата към даден файл или директория могат да се проверят чрез командата ls˽-l

От листинга правата се отнасят към първата колона. В таблицата са дадени примерни стойности за един файл и една директория.

|  |  |  |  |
| --- | --- | --- | --- |
| Тип на файла | собственик | група | всички останали |
| - | rwx | rw- | r-- |
| d | rw- | r-x | --- |

//Типа за обикновените файлове се означава с ‘-’ . Други типове за файловете са l-символна връзка, b-блоково устройство , s-символно устройство и др.

Промяната на правата може да се извърши от собственика чрез командата change mod:

chmod <value> filename

Където value е режима и може да се зададе с цифри или със символи.

1. Задаване чрез цифрови стойности

Ако искаме да зададем на filename следните права:

* за собственика: rw-
* за групата r-x
* за всички останали: r--

използваме съответствието, че където има право е 1, а където няма е 0 и преобразуваме до 10-тично число:

|  |  |  |
| --- | --- | --- |
| r w - | r - x | r - - |
| 1 1 0 | 1 0 1 | 1 0 0 |
| 110=> **6** | 101=> **5** | 100=> **4** |

Изпълняваме :

chmod ˽ 654 ˽ filename

Проверяваме чрез:

ls ˽ -l

Ясно е че :

chmod ˽ 777 ˽ filename - ще зададе пълни права за всички потребители

chmod ˽ 000 ˽ filename - ще махне всички права

II . Задаване на правата чрез символи:

* символи за потребители: u (user), g(group), o (other), a (all)
* режим : + (задаване), -(премахване), = (приравняване)
* достъп: r,w,x

Примери:

u+wx - задаване права на собственика за запис и за изпълнение

go-x - премахване на групата и на всички останали правото за изпълнение

g+x,o+w - задаване на групата право да изпълнява и на всички останали да могат да четат (интервал след запетаята не трябва да има)

a= премахване на всички права за всички

a=x задаване на всички да имат право **само**  да четат

a=rwx - задаване на пълни права за всички

Горните примери се изпълняват по следния начин:

chmod u+wx filename

----------

Освен правата съществуват и някои допълнителни флагове като: sticky bit, setuid, setgid.

Правата могат да се задават и чрез графичната система или някои програми.

======================================================================

**4. Работа с текстови редактори и файлови мениджъри:**

**4.1.** Работа с редактора vim

sudo apt-get install vim

vim - отваряне на текстовия редактор

Редактора има два режима : команден и за въвеждане.

За да зададем режим за въвеждане натискаме клавиша “i” - за insert или “a”-за append от клавиатурата.

-> пишем няколко реда и за запазим написаното като файл минаваме в команден режим с ESC. Тогава може да създадем файла:

:w myvimfile

-> може да напишем още някой ред и да съхраним:

:w

Ако искаме да излезем:

:q - изход, но преди това трябва да сме записали

:q! - изход без да запишем последните промени

:wq - запис и изход

Ако искаме да отворим файла за редакция с vim:

vim myvimfile

--

vim -R myvimfile - отваря файла в режим Read Only

view myvimfile

-------

**4.2.** Други текстови редактори:

nano,pico, view, joe,ex

**4.3.** Файлови мениджъри:

mc - отваряне на файловия мениджър Midnight commander

sudo apt-get install mc

други:

sudo apt-get install gnome-commander

sudo apt-get install vifm

sudo apt-get install lfm

=============================================================

**5. Архивиране на файлове**

Създаваме няколко файла:

touch file1 file2

-----------

**5.1 .TAR** (type archive) - служи само за пакетиране на няколко файла в един, но не за компресиране

за да архивираме (compress) с tar формат

tar -cf archive.tar file1 file2

ls

нека да разархивираме (extract)

rm file1 file2

tar -xf archive.tar

ls

--------------------

**5.2.** За архивираме с GNU Zip - един файл

gzip file1

ls

разархивиране:

gunzip file1.gz

------------------------------------------

**5.3** За архивиране с BZ2 - само за един файл

bzip2 file2

bunzip2 file.bz2

Обикновено архивите първо се пакетират с tar и след това се компресират с gz или с bz2

gzip archive.tar - ще създаде архива archive.tar.gz

-----------------------------------

**5.4**  Архивиране с RAR

Необходимо е да се инсталират програмите rar и unrar

sudo apt-get install rar

sudo apt-get install unrar

За да добавим файловете file1 и file2 към архива arc.rar изпълняваме:

rar a arc.rar file1 file2

Разархивирането става с командата unrar:

unrar e arc.rar

=============================================================

**6. Администриране на процеси**

За да предгледаме всички (-e) стартирани процеси може да използваме следните команди:

ps

ps -eF

ps -ely

ps -ef

В последния листинг има :

UID - user ID,

PID - process ID

PPID - parent process ID

TTY - терминал (tty7 е графичната среда)

CMD - името на процеса (тези в правоъгълни скоби са нишки на ядрото)

Може да се използва и BSD синтаксис:

ps ax

ps aux

За да се изпечатат процестие в дървовиден вид (родителски - дъщерни):

ps -ejH

ps axjf

pstree

Ако от командите резултата в терминала е много като текст, може да го разделим на няколко екрана чрез командата less и използване на конвейр (|)

ps -ef | less

**6.1. Статус на процес** (от страницата: man ps):

PROCESS STATE CODES  
 Here are the different values that the s, stat and state output specifiers (header "STAT" or "S") will display to describe the state of a process:  
 D - uninterruptible sleep (usually IO)  
 R - running or runnable (on run queue)  
 S - interruptible sleep (waiting for an event to complete)  
 T - stopped, either by a job control signal or because it is being traced.  
 W - paging (not valid since the 2.6.xx kernel)  
 X - dead (should never be seen)  
 Z - defunct ("zombie") process, terminated but not reaped by its parent.  
  
 For BSD formats and when the stat keyword is used, additional characters may be displayed:  
 < - high-priority (not nice to other users)  
 N - low-priority (nice to other users)  
 L - has pages locked into memory (for real-time and custom IO)  
 s - is a session leader  
 l - is multi-threaded (using CLONE\_THREAD, like NPTL pthreads do)  
 + - is in the foreground process group.

----------

Нека да стартираме процес от друг терминал, например tty1 - > Ctrl+Alt+F1. Влизаме първо с локалния потребител и после като root (sudo su -) и стартираме програмата Midnight Commander с командата mc.

После се връщаме в терминал от графичната среда Ctrl+Alt+F7

ps –ef | grep mc

ps –ef |grep –v root

Спираме програмата Midnight Commander от първия виртуален терминал и излизаме от сесия (Ctrl+D).

----------------

**6.2. Манипулиране с процеси.**

За управление на процесите най-често се използва изпращане на сигнали до тях. Сигналите се изпращат до даден процес чрез командата kill, като се знае номера на процеса (PID).

Ако искаме да прегледаме списък с възможните сигнали използваме:

kill -l

или

man 7 signal

За прекратяване на процес се използва SIGNKILL -9

Например нека да стартираме командата cat във фонов режим

cat&

Чрез командата ps може да проверим PID на cat и съответно да я прекратим с:

kill -9 catPID

Когато трябва да се унищожат всички инстанции на даден процес се използва killall и неговото име.

Сигнали към даден процес могат да се изпращат и от клавиатурата, където (Ctrl + C) -> 2)SIGINT , (Ctrl + D) - > 3)SIGQUIT.

Друг често срещан вариант за наблюдение на процесите е чрез командата

top

Представя се списък с изпълняваните процеси, който се обновява през 3 sec. Колоната PR указва приоритета на процеса, като "-20" е най-високия, а "20" е най-ниския.

За Ubuntu процесите могат да се разглеждат и управляват в графичната среда чрез System Monitor.

===============================

**6.3. Процеси във фонов режим - daemons**

Това са процеси които се изпълняват във фонов режим и обикновено изпълняват определена услуга.

Значението на думата daemon означава дух покровител и е неправилно превеждано като demon.

Най-популярния от този тип процеси е cron, чрез който се изпълняват програми по определен график. Конфигурационните файлове на cron се наричат crontab (от cron table) и обикновено се намират в /etc/crontab или за отделните потребители в /var/spool/cron.

Нека редактираме crontab файла за даден потребител (local):

crontab –u local –e

Може да изберем с кой текстов редактор да редактираме файла. Примерно vim.

Във файла всеки един ред задава график за изпълнение на една програма (или комнада). Графика се задава от първите пет символа, където:

┌───────────── min (0 - 59)  
 │ ┌────────────── hour (0 - 23)  
 │ │ ┌─────────────── day of month (1 - 31)  
 │ │ │ ┌──────────────── month (1 - 12)  
 │ │ │ │ ┌───────────────── day of week (0 - 6) (Sunday to Saturday;  
 │ │ │ │ │ 7 is also Sunday)  
 │ │ │ │ │  
 │ │ │ │ │  
 \* \* \* \* \* command to execute

Всички редове започващи с # са коментари за конфигурационните файлове.

Графика може да се зададе и чрез макроси:

|  |  |  |
| --- | --- | --- |
| Entry Description Equivalent to | Entry Description Equivalent to | Entry Description Equivalent to |
| @yearly (or @annually) | Run once a year at midnight of 1 January | 0 0 1 1 \* |
| @monthly | Run once a month at midnight of the first day of the month | 0 0 1 \* \* |
| @weekly | Run once a week at midnight on Sunday morning | 0 0 \* \* 0 |
| @daily | Run once a day at midnight | 0 0 \* \* \* |
| @hourly | Run once an hour at the beginning of the hour | 0 \* \* \* \* |
| @reboot | Run at startup | N/A |

\* означава всички. Но не е добре да се поставя звезда в първото поле освен ако не искаме да се изпълнява всяка минута.

45 10 \* \* 1-5 означава ”10:45 от понеделник до петък”

при полетата за деня от седмицата и деня трябва да се внимава, например:

0,30 \* 13 \* 5 означава ”всеки половин час в петък и всеки половин час на 13-то число от месеца ”

Полето за команда е всяка валидна команда от обвивката. Може да сложим списък от команди, напр. (команда1; команда2;команда3)

Пример:

да напишем в crontab файла:

\* \* \* \* \* env DISPLAY=:0 gnome-calculator

и след това излизаме от vim с ESC като пишем :wq

Това ще стартира на всяка минута калкулатора което не е особено смислено, но показва че е възможно да се стартират и графични програми .

спирането на cron за дадения потребител (local) става с:

crontab -u local -r

================================

**6.4 Нива на стартиране (runlevels)**

Използват се за указване в какъв режим да се стартира системата : еднопотребителски , многопотребителски, без графична среда и т.н. Обикновено не се ползва вече. Използваха командите init и telinit за превключване на различните нива, а проверката за ниво е чрез who -r.

За Ubuntu нивото на старитране по подразбиране е зададено във конфигурационния файл :

/etc/init/rc-sysinit.conf

Най-често се налага спирането на графичната среда с цел отстраняване на някои проблеми. Това може да стане и без да се минава в друго ниво на стартиране.

Пример за спиране и пускане на X Server за Ubuntu. При Ubuntu се използва LightDM (display manager):

1. Ctrl+Alt+F1
2. Login -> local user
3. sudo service lightdm stop

(check with Ctrl+Alt+F7 )

4. правим необходимите настройки

5. sudo service lightdm start

Ниво 0 е за спиране на системата, но по най-грубия начин. По приемлив вариант е чрез:

shutdown -h +1 - спиране на системата след 1 минута

==============================

**7. Администриране на потребители**

Оникновено администрирането на потребители се прави само от root или в частност от потребителите в групата sudo.

Отново разглеждаме файлът /etc/passwd, който представлява списък с потребители, които системата разпознава. Тя го чете, когато някой потребител влиза в сесия – от него взима UID идентификатора на потребителя и домашната му директория. Всеки ред представлява запис за потребител, като всяко поле е разделено с двоеточие

Добавянето на потребители може да става чрез командата useradd или чрез adduser. Разликата е че useradd е компилирана програма, докато adduser е perl скрипт за по-лесно и разбираемо използване. В себе си adduser използва useradd. В повечето дистрибуции се препоръчва използването на adduser.

//По-долните команди е необходимо да се изпълняват от root или чрез sudo.

useradd –m newuser - добавяме нов потребител с домашна директория

passwd newuser - добавяме парола за потребителя (нека да е същата като на локалния потребител)

cat /etc/passwd - проверяваме

ls /home/newuser

ls -l newuser

// vipw - редактира файлът passwd

groups newuser -проверяваме коя е текущата група на потребителя

groupadd new - добавяме нова група с името new

// gpasswd new добавяме парола за новата група

отваряме Midnight Comander

sudo mc

отиваме в директорията /etc и отваряме файлът group с F4 за редакция

задаваме новия потребител да бъде към групата

….

new: x: GID: newuser F2 – SAVE, F10 - exit

….

проверяваме

groups newuser

sudo mc

отиваме във файла с хешираните пароли:

/etc/shadow

Когато се създава парола на потребител тя се хешира с криптогравска хеш функция до определен стринг. Всеки път когато влиза потребител написаната от него парола е хешира отново и се проверява с хеша във файла. Ако съвпада се отваря сесия за потребителя. Има няколко начина за хеширане на пароли:

MD5 започва с $1$ - Изключително слаба хеш функция.

NIS

Blowfish започва с $2a$

SHA512 започва с $6$

За потребителя root при Ubuntu има символа !, т.е. няма създадена парола! За локалния потребител и за newuser хешовете са различни, въпреки че използвахме еднакви пароли. Това е така, тъй като към паролата се добавя допълнителен стринг (най-често username) преди хеширане.

Ако искаме да забраним новия потребител:

* слагаме звезда (\*) пред хеша на дадения потребител, съхраняваме и излизаме

--> пробваме за да се убедим, че няма да успеем да се логнем.

За изтриване на потребител може да използваме командата userdel или deluser (аналогична на adduser и отново е perl скрипт).

userdel newuser

rm –r /home/newuser

groupdel new

-------------------------------------------------------------------

Вариант с adduser и deluser:

sudo adduser newuser

sudo usermod -aG sudo newuser - добавяме newuser към групата sudo

sudo deluser --remove-home newuser

======================================================================

**8. файлови системи**

В Linux най-често са използвани файловите системи ext3FS и ext4FS.

За да разгледаме монтираните устройства използваме:

df -hT

или

sudo lsblk -f

Дисковете които са sata се откриват по следния начин:

/dev/sda1 - sata устройство (хард диск) ‘а’ , 1-ви дял

Ако имаме няколко хард диска те ще се откриват с имената: sda, sdb, sdc, sdd и т.н.

Дяловете на един хард диск са номерирани от 1. По номера се определя и дали дадения дял е главен (primary) или логически (logical). Главните дялове са с номера от 1 до 4. Логическите започват от 5 и всички се съдържат в един главен дял наречен Extended.

sudo parted /dev/sda ‘print’ - Отпечатване информация за всички дялове на диска sda. (Дяловете са с номера +1)

За да бъде възможно четенето на информация от даден носител той трябва да бъде монтиран. Монтирането е термин характерен повече за Unix системите и означава, че даден носител може да бъде използван ако се укаже неговата файлова система. Монтирането става в определена директория от където да се чете. Стандартната точка за монтиране е в директорията /mnt, а /media се използва за преносимите устройства (usb hdd и usb flash drive).

Например ако на хард диска има друга операционна система (Windows) и знаем, че дяла /dev/sda1 е с файлова система ntfs и там се намира C:, можем да го монтираме в /mnt

sudo mkdir /mnt/C -създаваме точка за монтиране

sudo mount -t ntfs-3g /dev/sda1 /mnt/C

NTFS-3g е open source драйвер за разбота с NTFS дялове.

След приключване на работа устройтвото трябва да се демонтира:

sudo umount /dev/sda1

При преносимите устройства (флаш-памети и др. ) се използва автоматично монтиране, като по подразбирене точката на монтиране е директорията /media. Също така, когато направим инсталация на Linux при вече съществуваща инсталация на Windows, ще бъдат автоматично монтирани ntfs дяловете от Windows.

---------

Ако имаме инсталация на Linux върху виртуална машина, то за нея се отделя виртуален диск. Принципно не е възможно да се монтират дялове от хоста и затова горния пример с монтиране и демонтиране на C няма да е възможен. Нека направим пример за монтиране и демонтиране на директория когато Linux е инсталиран на виртуална машина (Virtual Box), а ОС на хоста е Windows. Създаваме директория в Windows с името HostDir и в нея слагаме един тестов файл test.txt.

От VBox -> Devices -> Shared Folders Settings -> Add Share -< посочваме пътя до HostDir>- ok

За да използваме файловата система (vboxsf) за VBox трябва да инсталираме допълнително:

VBox -> Devices -> Insert Guest Additions CD Image -> Install (Run) -> Restart

След това отиваме в терминала на Linux :

cd ~

mkdir FromHostDir - създаваме точка за монтиране (директория от която са четем)

Монтираме с файлова система vboxsf :

sudo mount -t vboxsf HostDir FromHostDir

Можем да променим нещо във текстовия файл test.txt за да проверим дали се отразява на хоста.

cd FromHostDir

cat >test.txt

sudo umount FromHostDir - демонтираме

Ако искаме може да изтрием точката на монтиране:

rm - r FromHostDir

--------------------

За проверка и възстановяване на файлови системи може да се използва командата fsck.

sudo fsck /dev/sda6

За да изпълним fsck за дяла /dev/sda6 той трябва да бъде демонтиран. Командата fsck връща стойности по които се определя текущото състояние.

ls /sbin/fsck\* - за да прегледаме кои файлови системи може да проверяваме

**8.1. Системи за начално зареждане (boot loaders)**

Стандартната система за начално зареждане в Linux e GRUB (GNU GRand Unified Bootloader). Чрез него има възможност да се зареждат и други операционни системи, като Windows например. Когато ОС са повече от 1, GRUB ще конструира Boot Menu.

Главните конфигурационни файлове за GRUB се намират в директорията /boot/grub

Един от тези файлове е grub.cfg, но обикновено не се редактира директно.

За промяна на настройките се използва файла /etc/default/grub

Най-често това са редовете:

GRUB\_DEFAULT=4

GRUB\_TIMEOUT=6

чрез които се указват номера на ОС по подразбиране и времето което да се изчака преди автоматичното зареждане.

Номерата на ОС са номерирани от 1, вместо от 0.

След промяна по файла се изпълнява:

sudo update-grub

------------------------------------------------------------------

**9. Мрежови настройки:**

host www.google.com

Една от основните команди е

ifconfig - interface configuration

Чрез нея могат да се разглеждат конфигурираните мрежови устройства и интерфейси, а също и да се настройват, чрез съответните аргументи и опции.

eth0 : Ethernet мрежова карта (в случая е първата - 0)

lo : localhost

enp0s3 : Ethernet network peripheral # serial #

ppp0 : ppp устройство

ippp0 : ISDN ppp устройство

wlan0 : wifi

ra0 : безжична мрежа

Понякога се използва задаването на виртуални мрежови интерфейси – за да се хостват повече уеб сайтове

eth0:0 , eth0:1 и т.н.

Командата ifconfig има разновидност и за wireless карти

iwconfig

Портове:

Портът е абстракция на физическия адрес през който се осъществява комуникацията. Портовете са достъпни чрез своя номер от 0 до 65535. Портовете които са с номер по-малък от 1024 обикновено са запазени за системни нужди и не се използват за други приложения.

Сокетът е абстракция на мрежов софтуер, осигуряващ входно изходна комуникация за дадено приложение. Сокътет се създава при валиден IP- адрес и номер на порт. Към един порт могат да се създадат много сокети, позволяващи на много клиенти да използват една услуга. Комуникацията през сокети става чрез потоци.

резервираните портове , могат да се видят във файла:

cat /etc/services | less

21 FTP

23 telnet

22 ssh

80 http

25 mail SMTP

Портовете от 0 до 1023 са запазени

ping <хост> ICMP заявка за ехо (Internet Control Message Protocol)

Ctrl+C за край

traceroute –n <хост> - показва пътя на пакетите

-n използва числови стойности

netstat –avn - показва таблица на връзките с другите компютри

sudo ifconfig eth0 down -изключваме мрежовата карта

sudo ifconfig eth0 up -включваме я

Проверяваме рутиращата таблица

route –n Ако на мястото на gateway имаме \* т.е. няма зададена

Чрез командата route може да се добави default gateway

route add default gw 192.168.3.1

Други приложения:

ssh - Secure Shell Protocol for Remote Login

VNC

------------------------

Пример за SSH връзка до отдалечен компютър:

Ако имаме два компютъра за примера ще използваме следните означения: Comp1 и Comp2. Съответно локалните потребители ще означим с local1 и local2. А ip адресите: 192.168.1.1 (Comp1) и 192.168.1.2 (за Comp2).

За двата Comp1 и Comp2 инсталираме ssh чрез:

sudo apt-get install openssh-server

Настройките за SSH се намират във файла /etc/ssh/sshd\_config. От там можем да променим номера на порта и др. характеристики.

Възможно е да направим връзка директно от Comp1 чрез:

ssh local2@192.168.1.2

но по-добре да използваме криптиран канал чрез публичен и частен ключ. За целта най-често се използва криптографския алгоритъм RSA.

Затова от Comp1 създаваме двойка ключове като чрез:

ssh-keygen -t rsa

//ще изиска да въведем името на файла където да се съхранят, но може да използваме тези които ни предлага. Ще трябва да въведем и passphrase като парола за идентификация на ключа.

Ключовете ще се съхранят съдържат в /home/local1/.ssh, като:

id\_rsa - съдържа частния ключ, който си остава при Comp1

id\_rsa.pub -съдържа публичния ключ, който и ще пратим на Comp2

Изпращането на публичния ключ от Comp1 на Comp2 става чрез:

ssh-copy-id local2@192.168.1.2

//ще ни поиска паролата за local2

Вече може да се логнем на Comp2 със local2 от Comp1:

ssh local2@192.168.1.2

и да променим правата за достъп до файла с публичните ключове (authorized\_keys) на local2

cd /home/local2/.ssh

chmod 600 authorized\_keys

Изхода става с

exit

Когато трябва да достъпим Linux система от Windows със SSH се използва програмата PuTTY.