

UD5 SO Multiusuari i multitasca

Que es linux?

Linux es la denominació d'un SO tipus UNIX i el nom del nucli (kernel)

Es un dels exemples més prominents de

- Software lliure
- Codi obert

Definicions:

- UNIX: Es un sistema operatiu de principis del anys 70 desenvolupat per Ken Thompson i Dennis Ritchie als laboratoris Bell es S.O. multiusuari multitasca. Es un sistema fàcilment portable a diferents tipus d'ordinadors. La idea bàsica era crear un entorn pel desenvolupament de programes.
- FSF (Free Software Foundation) any 1985
Objectiu: eliminar les restriccions sobre còpia, redistribució, enteniment i modificació dels programes
Missió: donar cobertura legal i econòmica al projecte GNU.
- GNU (Acrònim de GNU is Not UNIX) Es un projecte iniciat per Richard Stallman amb l'objectiu de crear un S.O. completament lliure de tipus UNIX. Van desenvolupar una gran quantitat de software basic per el S.O. però no el nucli. (kernel)
- Linux: És el nucli del SO. El va fer Linus Torvalds + comunitat de desenvolupadors.
- GNU/Linux (Distribució): Es la combinació entre nucli i aplicacions SO complet.
- GNU GPL (GNU General Public Lidence): Pretén garantir la llibertat de compartir i modificar el software i que aquest sigui lliure per a tots els usuaris.

Entorn de feina:

- L'entorn de feina per defecte és textual.
- Els entorns gràfics són:
 - o KDE
 - o GNOME

Usuaris i grups

Multiusuari: El S.O. ha de tenir un sistema de permisos clarament definit. Grups ens permeten distribuir els usuaris en col·lectius de les mateixes característiques.

Raons de l'èxit de Linux1.

1. Multiusuari i multitasca
2. Escrit en llenguatge d'alt nivell (C:) que és fàcil d'entendre, modificar i transportar.
3. Interfície senzilla i potent.
4. Sistema d'arxius de fàcil manteniment. (ext3)
5. Desenvolupat per i per a programadors.
6. Lliure distribució.
7. Codi obert.

Linux a l'administració pública

Alemanya, Xina, Brasil

Distribucions Espanya

- Linkat a Catalunya.
- LinEx a Extremadura
- Molinux a Castella la Mancha

Distribucions:

- Genèriques: Propòsit general
 - o Debian: Més estable i de més prestigi.
 - o Red Hat: Més prestigi. De pagament perquè ofereix serveis alternativa
 - o Fedora: Distribució basada en Red Hat i feta per la mateixa comunitat de Red Hat
 - o Suse: Basada en Red Hat. És de pagament perquè ofereix serveis alternatius.
 - o openSuse: Distribució lliure per a usuaris domèstics.
 - o Ubuntu: Basada en Debian. També distribueix versions per a servidors.
 - o Mandriva: Basada en Red Hat. Distribució d'escriptori.
- Específiques:
 - o Geexbox: Distribució autoarrencable que crea un entorn multimèdia.
 - o Knoppix Mame: Per jugar a jocs d'arcade.

Escala de desenvolupament de linux

Un estudi fet l'any 2000 sobre la distribució debian va demostrar que:

- conte 55 milions de línies e codi.
- Fent servir un desenvolupament convencional aquest S.O. hagués costat 1.900 milions de dòlars.

Execució d'ordres

Prompt: usuari@esi1a-230.~\$

Atenció: Linux diferencia entre majúscules i minúscules.

ls (≠Ls) (≠LS)

ls (= dir)

ls -l → Informació ampliada

ls -l -i => ls -li

Ordres bàsiques:

- **exit** Finalitza la sessió. És aconsellable desconnectar sempre per qüestions de seguretat. Els terminals no estan assignats, per tant podem entrar amb la mateixa identitat des de terminal diferents.
- **who** [am i]: ens informa dels usuaris que estan connectats actualment al sistema. mcastro@esi1a-230~\$ who
jfuertes pts/3 2008-01-11 13.32 (192.168.0.23)
- **write**: es fa servir per comunicar-nos amb els altres usuaris que estan connectats en aquest moment.
Per executar-ho write nom_usuari
Escriuiu el missatge i per finalitzar feu CTRL +D.

- **mesg**: Activa o desactiva la possibilitat de rebre missatges
 mesg y → Activar la recepció
 mesg n → desactivar la recepció
- **cal**: ens mostra un calendari. Sintaxi: cal [mes] [any]
 Forma de fer-ho servir.
 - o **cal**: calendari mes i any actual
 - o **cal any**: calendari de l'any
 - o **cal mes any**: Calendari del mes de l'any
- **date**: Informa de la data i la hora actual
 \$ date
 Vie ene 11 13.55.00 CET 2008
- **uname**: Aquesta comanda es fa servir per a obtenir informació del sistema.
 Ex: tipus de maquina i processador, S.O., data instal·lació del S.O., ...
 Sintaxi: uname [-a] [-m] [-v] [-s] [-n] [-r]
 - o -m : tipus de hardware i processador
 - o -n : nom del host
 - o -v : dona informació de quan va ser instal·lat el sistema
 - o -s : dona el nom del sistema
 - o -r : dona la versió del kernel
 - o -a : dona tota la informació anterior a la vegada
- **passwd**: ens permet modificar la passwrod. ATENCIÓ:
 - o Les passwords no es visualitzen a l'entorn textual.
 - o Ubuntu
 - Password >= 6 characters – login
 - No pot coincidir amb - login
 - password anterior
- **clear**: Serveix per netejar la pantalla.
- **script**: ens permet emmagatzemar en un arxiu tot el que l'usuari teclegi a partir del moment de la seva activació.
 Sinaxi: script [-a] [nom_arxiu]
 - o **script** : emmagatzema tota la sessió en un arxiu anomenat typescript.
 - o **Script nom_arxiu** : emmagatzema tota la sessió en un arxiu anomenat nom_arxiu.
 - o **script -a nom_arxiu**: emmagatzema tota la sessió en un arxiu anomenat nom_arxiu sense que aquest perdi la informació prèvia
 - o **exit**: per sortir de l'script.
- **mail**: Es un mecanisme de correu electrònic per a enviar missatges a la xarxa local. No és necessari que el receptor estigui connectat en aquest moment.
 Per executar-ho fem:
 - { **mail** → Anem a la safata d'entrada i es mostreu els missatges nous
 - { **mail nom_usuari** → Enviar email a l'usuari especificat.
 - o Per a passar un missatge a un altre fem intro
 - o Per a esborrar un missatge fem d
 - o Per a imprimir un missatge fem p
 - o Per a sortir q
 - o Per a finalitzar un missatge fem CTRL+D
- **man**: [secció] ordre AJUDA

Sistema d'arxius

Es defineix **arxiu** com un conjunt de dades que té un nom associat. Emmagatzemats en memòries secundàries. (HD/ CD-ROM/DVD/...) La utilitat dels arxius es emmagatzemar a llarg termini.

Linux tracta els arxius com una seqüència de bytes. El kernel no imposa cap estructura sobre els arxius.

Sistema d'arxius: Part del S.O. responsable de l'administració de fitxers.

Les característiques del sistema d'arxius de linux són:

- Els usuaris poden crear, modificar i esborrar fitxers
- Cada arxiu té definit tres tipus de permisos d'accés
 - o De lectura (r)
 - o D'escriptura (w)
 - o D'execució (x)
- Els tres tipus d'accés es poden estendre a:
 - o El propietari del arxiu
 - o El grup al que pertany el propietari
 - o La resta d'usuaris
- Proporciona possibilitat de xifrat i còpies de seguretat.

A linux els fitxers estan organitzats en directoris en forma d'arbre.

Tots els directoris i fitxers depenen del directori arrel es denota per la barra normal (/). Els diferents dispositius d'emmagatzemament (HD, CD-ROM, DVD, Pen drive, ...) també depenen del directori arrel (tenen els seus subdirectoris dins de la carpeta /media

Directoris especials

- . → Directori actual
- .. → directori pare

Directoris importants

- / (Directori arrel) : Només hi ha un i es denota per / No té directori pare. S'emmagatzema un arxiu binari que és el nucli d'arrencada de linux.
- /bin : Conte moltes de les ordres executables que es fan servir a linux com per exemple ls, etc.
- /home : Pugen tots els directoris de feina de tots els usuaris. Els usuaris poden fer el que vulguin en el seu directori.
- /usr/bin : emmagatzema arxius binaris d'ús menys freqüent i més grans.
- /usr/lib : Conte les biblioteques que fan servir els compiladors. (c , c++ , perl , etc)
- /usr/man : Conte les pàgines del manual.
- /usr/local/bin : Arxius executables que no pertanyen a linux estàndard creats per els administradors.
- /etc : Conte arxius de text de configuració del sistema
 - o /etc/apache → Configuració del servidor web apache
 - o /etc/php5 → Configuració del script php
- /dev : conte els arxius dels dispositius d'E/S.

- /boot : Configuració de la arrancada de Linux.
 - o /boot/grub/menu.lst (Configuració del S.O. de arrancada)

Nom d'arxius i directoris

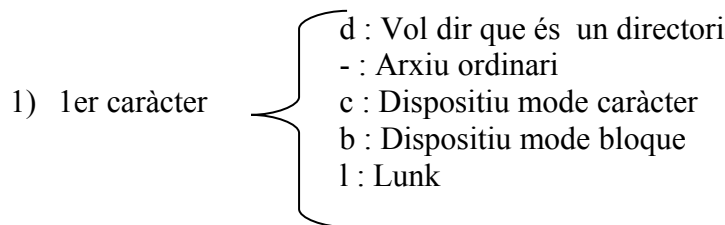
Els noms d'arxius i directoris han de tenir les següents de característiques:

- 255 caràcters
- Es poden fer servir tots els caràcters encara que hi ha alguns no recomanats com per exemple: * ? < > ! ; [] \ \$ “ ()
- A linux no existeixen les extensions. Podeu posar tantes com vulgueu o no posar cap.

Manipulació d'arxius i directoris

- ls : Llistar els arxius continguts en un determinat directori.
 Sintaxi: ls [-lFad] [directoriu][arxius]
 - o ls : Dóna com a sortida tots els noms de fitxers i directoris del directori actual.
 - o ls -F : Dóna com a sortida la mateixa llista anterior però afegint h caràcter @ per a cada directori i un caràcter * per a cada arxiu executable.
 - o Ls -l : Dóna com a sortida una informació el mes extensa possible.

```
$ls -l
-rwxr-xr-x 1 djimenez esi1a 6265 2008-01-07 texto.txt
```



1r Grup de 3 caràcters = Permisos de lectura, escriptura i execució per a l'usuari propietari.

2on Grup de 3 caràcter = Permisos de grup

3er Grup de 3 caràcters = Permisos de la resta d'usuaris

- 2) Ens indica el nombre d'enllaços que te l'arxiu. (Accés directe)
 - 3) Si l'element es un directori aquest nombre indica el número de subdirectoris.
 - 4) El grup al que pertany l'usuari propietari.
 - 5) La mida de l'arxiu però amb bytes.
 - 6) Data i hora de la darrera modificació.
 - 7) Nom de l'arxiu.
- ls -a : Ens permet veure tots els arxius i directoris, fins i tot els ocults → son fitxers que comencen per .(punt)

```
ls Examples
ls -l Examples
```

- **ls -ld directori** : Permet veure totes les propietats d'un directori i no el seu contingut.
- **pwd** : Ens mostra el directori actual en forma de trajectòria absoluta.
/home/djimenez
- **cd** : ens permet moure'ns entre els directoris. El camí pot ser absolut o relatiu. Si no especifiquem cap paràmetre ens envia al nostre HOME.
Sintaxi: cd [directori]
- **mkdir** Aquesta comanda ens permet crear directoris. Es poden fer servir rutes relatives i absolutes i només podrem crear directoris a el nostre home.
 - o mkdir /home/djimenez/src → creu el directori src a la carpeta /home/djimenez
 - o mkdir djimenez/src
- **rmdir** : Esborrem directoris si estan buits.

Editor de text VI

Com editar o crear un arxiu de text?

\$vi nom_arxiu

Si l'arxiu existeix → l'obre i es col·loca a l'inici

Si l'arxiu no existeix → el crea

Per col·locar-nos a una línia en concret fem \$vi +n nom_arxiu
(a on n es el n^a de línia)

Per a col·locar-nos a l'última línia fem \$vi + nom_arxiu

Estructura de les ordres vi

vi te dos modes de treball:

- Mode comanda: Ens permet donar ordres pròpies de l'editor
- Mode Edició: Permet introduir text

Quan entrem al vi estem en mode comanda

Mode comanda =>(i) mode edició

Mode edició =>(ESC) Mode comanda

Ordres més comunes en mode comanda

- Sortir → :q
- Sortir sense guardar → :q!
- Guardar → :w
- Guardar i sortir → :wq
- Guardar en un lloc i nom concret → :w /home/pflorez/nom_arxiu

Ordres comunes del vi

- o (minúscula) : Obria línia posterior a on es troba el cursor
- O (majúscula) : Obria línia anterior a on es troba el cursor
- dd : Es borra la línia en la que es troba el cursor
- u (Minúscula) : Desfer l'últim canvi que hem fet.

Cercar Patrons

- /patro : Cerca el patro des d'on es troba el cursor en endavant
- ?patro : Cerca el patro des d'on es troba el cursor en enderrere
- n : Repeteix la cerca.

Moure o copiar blocs de text

- Moure
 - o Col·loquem el cursor a la 1^a línia del bloc que volem moure
 - o Fem ndd (a on n es el nombre de línies que volem moure)
 - Esborra n línies i les guarda al buffer
 - o Per enganxar el que hem retallat fem p
- Copiar
 - o Col·loquem el cursor a la 1^a línia del bloc que volem copiar
 - o Fem nyy (a on n es el nombre de línies que volem copiar)
 - copia n línies al buffer
 - o Per enganxar el que hem copiat fem p

Comandes de manipulació de fitxers

- cat : visualitza el contingut d'arxius ascii
Sintaxi: cat [arxiu]
- more: Imprimeix per pantalla un arxiu pantalla a pantalla
Sintaxi: more [arxiu]
Per moure's per pantalla farem:
 - Barra espaciadora : Salta a la pròxima pantalla
 - Enter : Salta una línia
 - q : sortir
- head : Ens permet visualitzar les primeres línies d'un fitxer.
Sintaxi: head [-N] arxiu (N és el número que indica el nombre de línies de l'inici que volen veure) Si no ho posem es mostren les 10 primeres línies
- tail : Ens permet visualitzar les últimes línies d'un fitxer.
Sintaxi: tail [-N] arxiu (N és el número que indica el nombre de línies del final que volen veure) Si no ho posem es mostren les 10 últimes línies
- cp: Aquesta comanda ens permet copiar arxius entre diferents directoris del sistema
Sintaxi: cp arxiu(s) directori
Exemples:
 - cp serv.c /home/djimenez/src/ → Copia serv.c del directori actual al directori /home/djimenez/src/ amb el mateix nom
 - cp serv.c /home/djimenez/src/abc.dat → Copia serv.c del directori actual al directori /home/djimenez/src/ amb el nom abc.dat
 - cp /etc/apache.conf /home/djimenez/dat/hola.txt
- mv: Ens permet moure arxius d'un lloc a un altre. També ens permet renombrar arxius i directoris
Sintaxi: mv arxiu(s) desti
Exemples:
 - mv fotots/img.jpg tmp/
 - mv /home/djimenez/hola.txt /home/pflorez/adeu.txt
- rm : Ens permet esborrar arxius i directoris
Sintaxi: rm [-irf] arxiu

-f (Force) Força l'esborrament d'arxius de sol lectura

-i (interactiu) Pregunta abans d'esborrar.

-r (recursiu) Esborra tots els arxius de subdirectoris dependents

Exemples:

rm hola.txt

rm /home/pflorez/hola.txt

rm -r /home/djimenez/*

- file: Linux no imposa cap estructura sobre els fitxers. Per a saber de quin tipus és el fitxer fem servir file

Sintaxi: file fitxer

Exemples:

\$file archivo

archivo: Microsoft Office Document

\$file archivo2

Archivo2: zip archive data, at least v 2.0

Us d'arxius i permisos

Permisos associats a:

- Usuari Propietari
- Grup de l'usuari propietari
- Resta d'usuaris

Determinació dels permisos

r – lectura

w – escriptura

x – Execució

- - Sense permís

- **chmod** : Ens permet modificar els permisos d'arxius i directoris

Sintaxi: chmod mode arxiu(s)

El permís es pot especificar de 2 formes:

Exemple r w x r - x r - -

	Usuari	grup	Altres
Permisos	r w x	r - x	r - -
Binari	1 1 1	1 0 1	1 0 0
Números	7	5	4

chmod 754 arxiu

Exemple amb lletres

u = usuari g = grup o = altres

chmod g+w arxiu

Això canvia els permisos del grup i ara podra escriure

Permisos per defecte: r w - r - - r - -

Per canviar els permisos per defecte utilitzarem umask

- **umask:** Quan creem un arxiu te uns permisos per defecte. Si volem canviar aquests permisos farem servir umask.

Sintaxi: umask [mascara]

Si no especifiquem la mascara us informará dels permisos per defecte que son actius.

Els permisos màxims que pot tenir un arxiu son: r w – r w – r w –

En decimal es 666

Si nosaltres volem que els permisos per defecte siguin r w – r - - - - -

En decimal 640

666

-640

026

umask 26

- **whereis:** Els paràmetres que accepta aquesta comanda son ordres del S.O. Linux. I ens retorna a el directori a on es recideix l'ordre i la pagina del manual.

Exemple: whereis ls
 /bin/ls /usr/share/man/man1/ls.1.gz

Sintaxi: whereis [-bms] ordre

- o -b: Nomes cerca l'arxiu binari
- o -m: Nomes cerca la pagina del manual
- o -s: Nomes cerca el codi font.

- **id:** Retorna l'identificador d'usuari i de grup.
nº que fa servir linux per a saber a qui a d'aplicar els permisos

Sintaxi: id [-ug] [usuari]

- Si fm ed sense paràmetre ens mopstra el nostre id
- -u : mostra l'identificador d'usuari
- -g: mostra l'identificador de grup.

- **newgrp:** Aquesta comanda ens permet canviar l'identificador de grup.

Sintaxi: newgrp nom_grup

- **su:** Ens permet canviar el nostre identificador d'usuari i deixar-nos actuar com un altre usuari (evidentment hem de conèixer la password de l'altre usuari)

Sintaxi: su [-] [usuari]

- Si no especifiqueu paràmetres es com si ens identifiquem com a root (Administrador)
- Si posem – prenem els paràmetres d'inici del nou usuari.
- Per sortir d'una sesio, exit.

- **sudo:** Ens permet executar una comanda com si fosim l'administrador.

Sintaxi: sudo ordre

Intèrpret d'ordres

Es confon amb el S.O. Quan iniciem Linux el sistema executa automàticament un programa anomenat Shell (closca) encarregat d'interpretar totes les ordres que introdueix l'usuari. El Shell indica que està present mostrant el prompt. Farem servir el Shell Bash desenvolupat per la FSF.

Hardware → S.O. → Aplicacions → Shell

El Shell porta incorporades algunes ordres dins del seu propi codi. Són les ordres intrínseques del Shell.

Modes d'invocació una ordre

- ordre & → Ens permet executar una ordre en segon pla
- ordre1; ordre2; ordre3; ... Permet executar varies ordres invocades des d'una única línia.
\$date; ls; cal 7 1980
- ordre1 | ordre2 (Canonades - Pipelines)
- ordre1 `ordre2` Substitució d'ordres. La sortida de l'ordre2 s fa servir com argument de l'ordre1.
\$ls `pwd`
- ordre1 && ordre2 Executa l'ordre1 i si finalitza amb èxit llavors executa l'ordre2
\$ cp && date

Historic d'ordres

Les ordres que anem executant queden emmagatzemades al sistema i per a veure-les fan la comanda history.

Les ordres apareixen llistades i numerades, per executar una ordre en concret farem !n on n és el número associat a l'ordre concret.

Metacaràcters

Ens permeten crear patrons de noms d'arxius

? * []

- ? → Es fa servir per a substituir qualsevol caràcter, però només un, excepte el primer punt pels arxius ocults
- [] → Defineixen una classe de caràcters. Es pot fer servir el :
 - o - → Determinar tots els caràcters d'un rang
 - o ! → Per a negar un conjunt de caràcters

Exemples: ls hol[ao]
 ls hol[acfo]
 ls hol [a-n]
 ls hol [a-nA-N]

- * → Substitueix zero o més caràcters, excepte un primer punt pels arxius ocults

Exemples: ls *.c
 ls r*

Alies

Es fan servir per a invocar ordres amb un nom diferent al que fem servir normalment.

Exemple: Volem executar ls -l fent servir l
 alias l="ls -l" Activa el alias l
 Els alias estan actius fins que finalitza la sessió.
 unalias l Desactiva el alias l

Redireccions d'entrada i sortida

Quan iniciem l'interpret d'ordres es creen arxius que fan referència a tres dispositius diferents :

- stdin l'arxiu d'entrada estàndard en general es el teclat
- stdout arxiu de sortida estàndard en general el monitor
- stderr arxiu de sortida estàndard de error en general a la pantalla

Redirecció d'entrada

- < Pren l'entrada d'un altra dispositiu diferent del dispositiu estandar

Redirecció de sortida

- > Dona la sortida d'un proces a un altre dispositiu diferent del monitor > crea o sobrescriu el arxiu existent
- >> Adereix al final del fitxer allo que escrivim

Redirecció de errors

A linux les comandes generen diagnostics de error per redireccionar aquest errors fem servir els símbols 2> o 2>>

Filtres

- sort: serveix per a ordenar línies d'un arxiu formades per camps separats per algun caràcter per defecte el caràcter de separació es espi en blanc o tabulació

Sintaxi: sort [-ntx] [-k group] arxiu

- n ordena numericament
 - tx x fa referencia a un caràcter es fa servir per especificar un caràcter de separació entre els camps dels fitxers
 - k camp (camp es un numero) aquet numero indica sobre quin camp volem ordenar comença a numerar els camps per 1
- wc: Contador de paraules, línies i caràcters
- Sintaxi: ws [-lwc] arxiu
- -l : Ens dona nomes el numero de línies
 - -w : Ens dona nomes el numero de paraules
 - -c: Ens dona nomes el numero de caràcters
 - Si no posem parametres ens dona tota la informació

- grep : Ens permet cercar cadenas de caracters (patrons) en arxius. Si el patro es de mes de una paraula llavors l'hem de incloure entre cometes dobles.

Sintaxi: grep [-inv] "Patro" arxiu

- -i : No fa distincions entre majúscules i minúscules
- -v : Visualitza les línies que no contenen el patró
- -n : Mostra el numero de línia a on es troba el patró

Exemple:

grep "posem" prova

Pipeline (Canonades)

En ocasions resulta convenient que la sortida d'una ordre actua com entrada d'una altre. Això s'aconsegueix amb les pipelines simbolitzades per |

\$who >> usuarios

\$xc -l usuarios

\$ rm usuarios

\$who | wc -l

Esquema de canonada



Esquema general

\$ ordre1 | ordre2 | | ordreN a on s'ha de verificar que:

- l'ordre situada a l'esquerra de | genera una stdout
- l'ordre situada a la dreta de | pren l'entrada de stdin

Exemple1 Volem veure els arxius que van ser modificats al Gener d'aquest any (tota la informació)

\$ ls -l | grep "2008-01"

Exemple2 Igual que l'exemple1 però ara els arxius s'han d'ordenar per la seva mida en bytes

\$ ls -l | grep "2008-01" | sort -n -k5

Expressions regulars

Que és? Una expressió regular es un patro que defineix un conjunt de cadena de caracters.

Elements que es poden incloure en una expresio regular:

- c : Qualsevol caracer en concret
- \c : Qualsevol caracer especial
- . (Punt) : Fa referencia a qualsevol caracer pero només un
- ^x : La línia comença per x
- x\$: La línia finalitza amb x
- x* : x Es repeteix 0 o més vegades
- x+ : x Es repeteix una o més vegades
- x? : x Es repeteix 0 o 1 vegada
- [c1c2c3...cn] : Representa un caracer que coincideix amb c1 o c2 o c3 o cn

- [^c1c2c3... cn] : Representa un caracter que no coincideix amb c1 o c2 o c3 o cn
- [c1-c2] : Representa un caracter que coincideix entre els caracters c1 i c2
- [^c1-c2] : Representa un caracter que no coincideix entre els caracters c1 i c2
- (x) : Serveix per agrupar.
- x {m,n} : x Apareix entre m i n vegades
- x {n} : x Apareix n vegades
- x {m,} : x Apareix m vegades com a minim

Exemple:

EXEMPLE	AMB QUE COINCIDEIX
gat	Qualsevol linia que contingui gat
^gat	Que la linia comença per gat
gat\$	Totes les linies que acabin per gat
^gat\$	Totes les linies que comencen i acaben per gat
gat [ao]	Totes les linies que contenen gato o gata
gat[^aeiou]o	Totes les linies que contenen gat seguit d'una consonant i seguit de o
^....\$	Totes les linies que tinguin 4 caracters
^\.	Totes les linies que comencin per punt
^[^.]	Totes les linies que no comencin per punt
gats*	Totes les linies que contenen gat o gats o gatss o gatsss etc.
gats+	Totes les linies que contenen gats o gatss o gatsss
(gats)+	Totes les linies que contenen gats o gats...gats
gats{5}	Totes les linies que contenen gatsssss
gats{3,5}	Totes les linies que contenen gatsss o gatssss o gatsssss
gats{3,}	Totes les linies que contenen gatsss o gatsss....s
Gat?o	Totes les linies que contenen gato o gao
\(gat\)	Totes les linies que contenen (gat)

dd-mm-aaaa ^[0-3][0-9]\-[0-1][0-9]\-[0-9]{4}\$

- **cut**: Serveix per tallar i pasar a la sortida estandard les columnes o camps d'un arxiu especificat

Sintaxi: cut -c llista arxiu(s) → Es fa servir per tallar columnes
 cut -f llista [-dX] arxiu(s) → Es fa servir per tallar camps
 o dX La X especifica el caracter de separacio entre els camps

Per especificar quins camps o columnes volen tallar fem servir la llista de la següent manera:

- A-B (A,B numeros): Camps o columnes de la A fins la B
- A- : Camps o columnes de la A fins el final
- A, B Camps o columnes A i B
- ACamps o columnes A

Exemple:

Tenim aquest arxiu anomenat persones:

SSP : 458

PPS : 542

JAD : 854

POL : 125

Per llistar els primers camps

cut -f 1 -d: persones

Per llistar els primers camps ordenats

cut -f 1 -d: persones | sort

Per llistar per columnes

cut -c 1-3 persones

- **tr** : Llegueix les dades de l'entrada estandard (teclat) i tradueix en funcio de les equivalencies que especifiquem

Sintaxi: tr [-dsc] Cadena1 Cadena2 < fitxer

- Si no posem parametres haurem d'especificar cadena1 i cadena2 i substitueix les coincidencies de cadena1 per cadena2
- -d : Nomes hem d'especificar cadena1. Esborra les aparicions de cadena1 al fitxer.
- -s: Hem d'especificar la cadena1. Esborra els caràcters repetits de la cadena1 al fitxer.
- -c: Hem d'especificar cadena1 i cadena2. Substituir les NO coincidencies de cadena1 per cadena2.

Exemples

Aquest es un arxiu de text.

QUE CONTE LLETRES MAJÚSCULES I minúscules

tr [a-z] [A-Z] < fich → converteix text a majúscules

tr [A-Z] x < fich → tot pasa a X

tr -d [A-Z] < fich → borrar totes le majúscules

tr -c [A-Z

] " " < fich → Totes les que no siguin de la A-Z substituira per espai en blanc

tr -s [a-zA-Z] < fich → Borra el que estigui repetit

find: Serveix per cercar fitxers dins el sistema

Sintaxi: find camí expressió

- Camí: lloc en el qual volen cercar (directorís)
- Expressió:
 - o -name NOM: Permet cercar els fitxers de nom NOM
 - o -regexp NOM: Permet cercar els fitxers que tenen un nom que coincideix amb l'expressió regular NOM
 - o -user NOM: Permet cercar els fitxers de l'usuari NOM
 - o -group NOM: Permet cercar els fitxers grup NOM
 - o -mtime N: Arxius modificats fa exactament N dies

- `-mtime -N`: Arxius modificat fa menys de N dies
- `-mtime +N`: Arxius modificat fa més de N dies.
- `-size -N [ckMG]`: Arxius que ocupen menys de N
- `-size +N [ckMG]`: Arxius que ocupen més de N
 - c → bytes
 - k → kilobytes
 - M → Megabytes
 - G → Gigabytes
- Type X: Arxius de tipus X a on X és:
 - l: link
 - d: directoris
 - f: arxius regular

Exemples:

`find / -name ifconfig 2> null` → Busca directori arrel un arxiu anomenat ifconfig (2>null evita els errors, els envia al dispositiu nul)

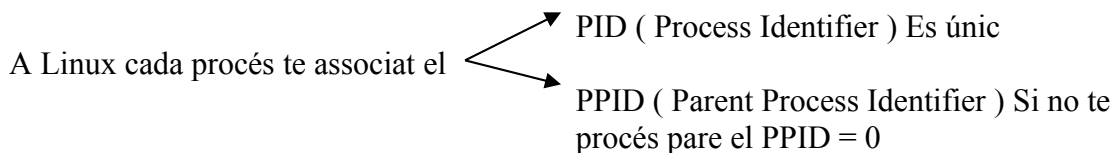
`find /home/pflorez/ -size +2M -mtime -5` → Busca arxius en el directori pflorez de mes de 2 Mb i que tinguin menys de 5 dies

`find /user/bin/ -name bt* -type l -size +25c` → Busca en la carpeta Bin nombres amb bt... de tipus link i qu ocupin mes de 25 Bytes

Programes i processos

Programa: Conjunt d'instruccions i dades.

Procés: Programa en execució



Un proces pot estar en tres estats basics:

- Sèsta executant
- Esta dormint (no pot continuar l'execució per que està a l'espera d'una operació d'entrada/sortida)
- No disposa del processador
- ps : Llista tots els processos que s'estan executan
 - Sintaxi: ps [-ef]
 - ps: sense opcions mostra tots els processos del nostre terminal
 - -e : Tots els processos del sistema
 - -f : Llista completa amb informacio adicional

Informació que ens dona:

UID → Usuari Propietari

PID → Identificador de Procés

PPID → Identificador Pare

C → Quantitat de recursos de CPU que el procés ha fet servir recentment.

STIME → Instant de començament del procés

TTY → Terminal associat al procés, si el procés no esta associat a cap terminal surt un interrogant.

TIME → Temps de CPU assignat

COMMAND → Nom del programa que ha generat el procés

- kill: Aquesta comanda permet enviar senyals (interrupció software) als diferents processos identificats pel seu PID.

Què fa un procés?

- Ignorar-la
- Tractar-la amb una rutina del kernel
- Tractar-la amb una rutina propia

Senyals:

9.SIGKILL: Provoca l'acabament d'un programa. No es pot ignorar.

15. SIGTERM: Indica a un procés que ha d'acabar la seva execució.

No pot ser ignorada. Aquesta es la senyal per defecte.

Sintaxi: kill [-senyal] PID

Exemple

Kill -9 8312

- time: retrona els temps relatius a l'execució d'un procés.

Sintaxi: time ordre [arguments]

Exemple:

sleep 5

time sleep 5

real 0m 5.003s

user 0m 0.000s

sys 0m0.004s

real: temps que passa des de que executen ordre fins que finalitza

user: Quantitat de temps que consumeix el procés executant el seu propi codi.

sys: temps que triga linux que fa servir l'ordre

- w : Aquesta comanda ens mostra informacio relacionada amb els usuaris connectats en aquest moment i del seus procesos.

Sintaxi: w [-hs] [NomUsuari]

- -h : Elimina les dues primeres files informatives
- -s : Us dona la informació en format curt
- Si especifiquem usuari ens dona informacio de aquest usuari en concret i si no especifiquem l'usuari dona l'informació de tots els procesos de tots els usuaris

Informacio que ens dona:

- La primera linia ens dona:
 - o Hora actual
 - o Temps que porta el sistema actiu
 - o Nombre d'usuaris connectats
 - o Carrega mitjana del sistema els ultims 1, 5 o 15 minuts
- La segona linia ens dona:
 - o USER: Nom d'usuari
 - o TTY: Terminal
 - o FROM: Ordinadors remots
 - o LOGIN: Hora de connexió al sistema del usuari

- IDLE: Temps desocupat
- JCPU: Temps de processador associat a tots els processos del terminal
- TPCPU: Temps de processador del procés indicat en el WHAT
- WHAT: Procés que s'executa en aquest moment per aquest usuari

Programació de l'interpret d'ordres

Els interprets d'ordres es comporten com a veritables llenguatges de programació.

Per poder executar els arxius s'executen així: `./Prueba`

Per posar comentaris s'utilitza el símbol `#`

Funció d'impressió de dades per pantalla

Es la funció `echo`. `echo [-ne] "Cadena de text"`

- Sense parametres → Imprimeix la cadena de text i salta a la línia següent.
- `-n` → Imprimeix la cadena de text sense fer un salt de línia
- `-e` → Activa l'interpretació de caràcters especials dins la cadena de text a mostrar

Caràcters especials

- `\a` → Alarma (beep)