


Comandos Sistema Operativo Linux

Roberto Gómez Cárdenas
rogomez@itesm.mx
<http://homepage.cem.itesm.mx/rogomez>

Lámina 1 Dr. Roberto Gómez C.



El shell

El shell es un programa que lee líneas de comando de una terminal o archivo, las interpreta, y le indica a UNIX que hacer

- Interface entre los usuarios y UNIX
- Interprete de comandos
- Es un lenguaje de programación
- El shell no es parte del sistema operativo

Lámina 2 Dr. Roberto Gómez C.



Completando nombres largos

- No aplica para todos shells, solo los mas modernos
 - bash, tcshell
- Cuando se escribe el nombre de un archivo y se desea terminar de escribirlo “automáticamente” y suponiendo que no exista ambigüedad (es decir que no haya más de un archivo que empiece por los caracteres ya escritos)
 - al pulsar dos veces la tecla ESC el shell se encargará de completar el resto del nombre
 - en algunos Unix es ALT ESC y en algunos Linux es TAB
- Util para trabajar con un archivo de nombre

`text-utils.8.4.1.documentation.linux.tar.gz`

Lámina 3

Dr. Roberto Gómez C.



La línea de comandos

- La línea de comandos empieza en el prompt del Shell, hasta el <RETURN>
- La primera palabra es el nombre de un archivo ejecutable, o de un comando interno del Shell
- Sintaxis base de la línea comandos incluye
`comando opcion(es) argumento(s)`
- Es necesario dejar un espacio:
 - entre el nombre del comando y las opciones y/o los argumentos
 - entre las opciones y los argumentos
 - entre los argumentos

Lámina 4

Dr. Roberto Gómez C.



Elementos línea comandos

- El comando: ¿qué hacer?
 - es la primera palabra de la línea
 - corresponde al nombre de un archivo ejecutable
 - es posible utilizar un archivo de acceso relativo o absoluto
- Las opciones: ¿cómo hacerlo?
 - siguen al comando (separado por un espacio)
 - generalmente precedida por un '-' (a veces un '+')
- Los argumentos: ¿sobre quién actuar?
 - generalmente uno, o varios, nombres de archivos

Lámina 5

Dr. Roberto Gómez C.



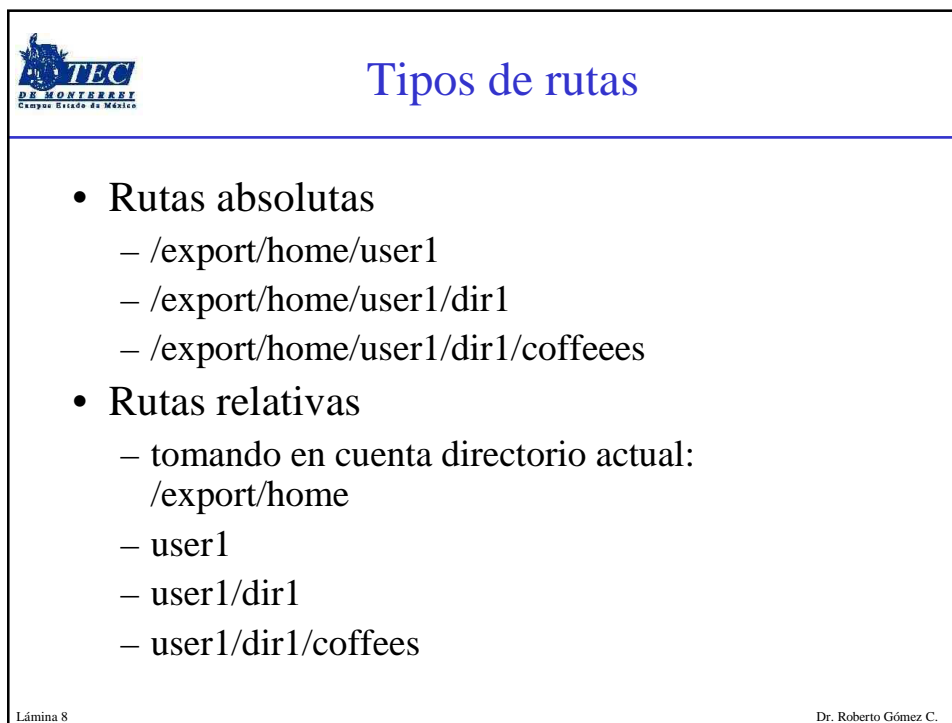
El comando pwd

- Print Working Directory
- Despliega el path completo del directorio actual
- Uno de los comandos más simples de unix
- Ejemplo

```
$ pwd  
/home/toto  
$
```

Lámina 6

Dr. Roberto Gómez C.





Convenciones de nombres

- Hasta 255 caracteres
 - caracteres alfanuméricos y no alfanuméricos
- No se deben usar caracteres especiales como
 - asterisco
 - ampersands
 - Pipes “|”
 - comillas
 - signos de pesos
- Espacios tampoco deben usarse
- Como regla, nombres directorios y archivos no contienen extensiones
 - Aunque se pueden usar

Lámina 9

Dr. Roberto Gómez C.




El comando ls

- Lista información acerca de los archivos (el directorio de trabajo por defecto)
- Acomoda la salida alfabéticamente i no se le indica lo contrario
- Algunas opciones
 - a: despliega archivos ocultos
 - F: clasifica archivo con caracteres */=@|
 - l: listado en formato largo
 - d: lista nombres directorios en lugar de contenidos
 - r: acomoda la salida en orden inverso
 - t: acomoda la salida de acuerdo a su tiempo de modificación
 - x: lista la salida por líneas en lugar de columnas

Lámina 10

Dr. Roberto Gómez C.



Ejemplo comando ls (opciones a y F)

```


$ ls
Algo fecha hola ls passwd prbUmask TestUmask
datos firulais hola.c Mod.php probando salida
david galloclaudio jitomate numeros prueba saludos
dos gc kilo papas prueba.c test

$ ls -a
. dos hola Mod.php passwd salida
.. fecha hola.c numeros probando saludos
Algo firulais jitomate .oculto1 prueba test
datos galloclaudio kilo .oculto2 prueba.c TestUmask
david gc ls papas prbUmask

$ ls -F
Algo/ fecha hola* ls passwd prbUmask TestUmask/
datos firulais hola.c Mod.php probando salida*
david galloclaudio jitomate numeros prueba* saludos*
dos gc kilo* papas prueba.c test

$
    
```

Lámina 11
Dr. Roberto Gómez C.



Opción larga comando ls

- Opciones comunes
 - a, F, l, d, t, r
- Tipos archivos
 - d: directorio, -: regular,
 - c: especial, s: socket
 - f: fifo
- Ejemplo

```

$ ls
total 4
drwxr-xr-x 1 cachafas 512 Oct 12 10:13 Sundraw
drwxr-xr-x 1 cachafas 512 Dec 11 20:13 Sunpaint
-rwxr-xr-x 1 cachafas 512 Sep 15 18:13 toto
-rw-r-x--x 1 cachafas 512 Jan 12 1999 curso.html

$
            
```

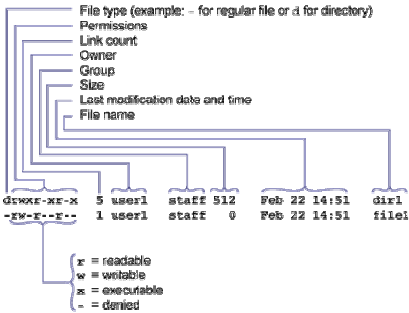




Lámina 12
Dr. Roberto Gómez C.



El comando echo

- Sintaxis
echo [OPCIONES] ... [STRING] ...
- Dos acciones
 - Despliega una línea de texto
 - Despliega el contenido de una variable de ambiente
 - Necesario anteceder el nombre de la variable de un carácter “\$”.

Lámina 13 Dr. Roberto Gómez C.



Ejemplo echo

```
$ echo Hola Mundo
Hola Mundo
$ echo PWD
PWD
$ echo $PWD
/home/toto/Pasados
$ echo $TERM
xterm
$
```

Lámina 14 Dr. Roberto Gómez C.



Usar echo y > para crear archivo

- Parámetro de echo y el carácter > permiten crear un archivo con lo que se le pase de parametro a echo
- Sintaxis

```
echo [string a almacenar] > [nombre archivo]
```

- Ejemplo

```
$ ls
$ echo hola mundo > hello
$ ls
hello
$ more hello
hola mundo
$
```

Lámina 15

Dr. Roberto Gómez C.



El ambiente Unix

- Cuando uno entra al sistema UNIX, el sistema proporciona un **ambiente** propio.
- El **ambiente** del usuario contiene toda la información necesaria, así como ciertas variables.
- Ambiente definido por variables locales y externas.
- Las variables locales sólo son conocidas por el shell que las creó o modificó.
- Las variables exportadas pueden ser vistas por todos los sub-shells.
- Usuario puede definir las variables locales y externas

Lámina 16

Dr. Roberto Gómez C.



El comando set y variables ambiente

- Comando set permite examinar variables de ambiente
- Ejemplos variables
 - USER y USERNAME cuyo valor es el login del usuario
 - UID con el número que identifica al usuario
 - TERM mantiene el nombre de la terminal que está usando
 - SHELL la ruta y nombre del intérprete de comandos
 - PWD el nombre del directorio de trabajo
 - HOME el nombre del directorio personal del usuario
 - PS1 y PS2 indican a bash como presentar prompts

Lámina 17

Dr. Roberto Gómez C.



Ejemplo salida comando set

```

$ set
BASH=/bin/bash
HOME=/home/rogomez
HOSTNAME=localhost
HOSTTYPE=i386
PS1='[\u@\h \W]\$ '
PS2='> '
PS4='+ '
PWD=/home/rogomez
SESSION_MANAGER=local/localhost:/tmp/.ICE-unix/3521
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor
SHLVL=2
SSH_AGENT_PID=3564
SSH_AUTH_SOCK=/tmp/ssh-XXqnlCD6/agent.3521
SUPPORTED=en_US.UTF-8:en_US:en
TERM=xterm
UID=501
USER=rogomez
$

```

Lámina 18

Dr. Roberto Gómez C.



Lo primero es lo primero: el prompt

- Símbolo que indica que el shell esta listo para recibir instrucciones.
- Existen prompts por default, dependiendo del shell y tipo de unix utilizado
 - \$ bourne shell o korn shell
 - % c shell o tc shell
 - # representa que el usuario es root
- Es posible que un usuarios defina su propio prompt a través de la variable de ambiente correspondiente
 - En el caso de Linux es PS1, por ejemplo el valor


```
PS1 = "\u@h \W]$ '
```
 - produce el siguiente prompt


```
[toto@localhost bin]$
```

Lámina 19

Dr. Roberto Gómez C.




Opciones para configurar el prompt

Carácteres	Significado
\u	Nombre usuario
\W	Directorio trabajo
\w	Ruta completa de trabajo
\t	La hora actual
\d	La fecha actual
\s	El nombre del shell
\h	El nombre de la máquina actual
\#	El número de comando
\!	La posición en el history
\\$	Carácter de prompt según
\nnn	Carácter nnn (en octal)

Lámina 20

Dr. Roberto Gómez C.




La documentación en línea: man

- Dividido en secciones:
 1. Comandos de usuarios
 2. Llamadas de sistema
 3. Subrutinas
 4. Dispositivos
 5. Formatos de archivos
 6. Juegos
 7. Varios
 8. Administración de Sistema
- Sintaxis del comando man


```
man [ opcion ] [ seccion ] titulo(s)
```

Lámina 21
Dr. Roberto Gómez C.



Primer ejemplo man

```

$ man tty

Reformatting page. Wait... Done

TTY(1)          USER COMMANDS          TTY(1)

NAME
tty - display the name of the terminal

SYNOPSIS
tty [-s]


DESCRIPTION
tty prints the pathname of the user's terminal unless the -s
(silent) option is given. In either case, the exit value is zero
if the estándar input is a terminal, and one if it is not.

OPTIONS
-s Silent. Does not print the pathname of the user's
terminal.

Sun Release 4.1 Last change: 9 September 1987

```

Lámina 22
Dr. Roberto Gómez C.



Segundo ejemplo man

```

$ man -s4 tty
Reformatting page. Wait... Done
TTY(4)  DEVICES AND NETWORK INTERFACES  TTY(4)

NAME
tty - controlling terminal interface

DESCRIPTION
The file /dev/tty is, in each process, a synonym for the
controlling terminal of that process, if any. Is is useful.....

IOCTLS
In addition to the ioctl() requests supported by the device that tty refers
to, the following ioctl() request is supported:


    TIOCNOTTY  Detach the current process from its controlling
                terminal, and remove it from its.....

FILES
/dev/tty

SEE ALSO
termio (4)

```


Lámina 23
Sun Release 4.1 Last change: 9 September 1987
Dr. Roberto Gómez C.



Scrolling Keys del comando more

Scrolling Keys	Propósito
Spacebar	Scrolls to the next screen
Return	Scrolls one line at a time
b	Moves back one screen
f	Moves forward one screen
h	Displays a help menu of features
q	Quits and returns to the shell prompt
/ string	Searches forward for string
n	Finds the next occurrence of string

Lámina 24
Dr. Roberto Gómez C.




Opción `-k` man y comando `catman`

- Posible invocar man con `-k <keyword>`
 - lista de comandos relevantes y relacionados con el keyword
 - por default esta opción no esta activada
 - el administrador debe activarla a través del comando `catman`

```
# catman -w
#
```

```
$ man -k calendar
cal      cal (1)      - display a calendar
calendar calendar (1) - reminder service
diftim   difftime (3c) - computes the difference
           between two calendar times
mktime   mktime (3c) - converts a tm structure to a
           calendar time
$
```


Lámina 25
Dr. Roberto Gómez C.



Comando `touch`

- Actualiza los tiempo de acceso y modificación de cada archivo pasado como argumento al tiempo actual
- En caso de que el archivo no exista, crea un archivo “vacío”.
- Algunas opciones:
 - `a`: solo cambia el tiempo de acceso
 - `d, --date=STRING`: utiliza `STRING` en lugar del tiempo actual
 - `m`: solo cambia el tiempo de modificación
 - `r, --reference=FILE`: utiliza tiempo del archivo en lugar del tiempo actual

Lámina 26
Dr. Roberto Gómez C.




Ejemplo comando touch

```

$ ls -l numeros
-rw-rw-r-- 1 toto  toto    189 Jul 21 19:24 numeros
$ date
Wed Jul 21 19:26:31 CDT 2004
$ touch numeros
$ ls -l numeros
-rw-rw-r-- 1 toto  toto    189 Jul 21 19:26 numeros
$

```

Lámina 27 Dr. Roberto Gómez C.




Ejemplo comando file

```

$ ls
ComandosUnix.pdf  fig-s1.gif  hello
cve_sans.gif      fig-s2.gif  hello.c
dollarlogo_20x30.gif  fig-s3.gif  linuxpenguinlogo_30x30.gif
fig1.gif          fig-s4.gif  openbsdheadlogo_30x30.gif
fig2.gif          fig-s5.gif  son4.txt
fig3.gif          fig-s6.gif  winlogo_30x30.gif
$ file ComandosUnix.pdf
ComandosUnix.pdf: PDF document, version 1.2
$ file hello
hello: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.2.5, dynamically linked (uses shared libs), not stripped
$ file hello.c
hello.c: ASCII text
$ file fig1.gif
fig1.gif: GIF image data, version 87a, 444 x 391
$

```

Lámina 28 Dr. Roberto Gómez C.



Ejemplo comando strings

```

$ cat hello.c
main()
{
  printf("Hola Mundo \n");
}
$ gcc hello.c -o hello
$ ./hello
Hola Mundo
$


```

```

$ strings hello
/lib/ld-linux.so.2
libc.so.6
printf
_IO_stdin_used
__libc_start_main
__gmon_start__
GLIBC_2.0
PTRh|
QVh(
Hola Mundo
$

```

Lámina 29
Dr. Roberto Gómez C.



Ejemplo comando more

```

$ more son4.txt

\documentstyle[indentfirst,epsf,10]{article}


% Formateando los margenes de la pagina

\textheight 23cm
\textwidth 15.25cm
\hoffset -1.25cm
\voffset -2.75cm

\begin{document}
% \fhyph
% \pagestyle{empty}
\def\tabesp{xxxxxxxxxxxx\=xxxxxxxxxxxxxxxx\=xxxxx\=\kill
--More--(3%)

```


Lámina 30
Dr. Roberto Gómez C.



Scrolling Keys del comando more

Scrolling Keys	Propósito
Spacebar	Scrolls to the next screen
Return	Scrolls one line at a time
b	Moves back one screen
f	Moves forward one screen
h	Displays a help menu of features
q	Quits and returns to the shell prompt
/ string	Searches forward for string
n	Finds the next occurrence of string

Lámina 31 Dr. Roberto Gómez C.



Ejemplo comando cat

```

$ cat a1
Soy el archivo a1
$ cat a2
Soy el archivo a2
$ cat a1 a2 > a3
$ cat a3
Soy el archivo a1
Soy el archivo a2
$
  
```

Lámina 32 Dr. Roberto Gómez C.



El comando date

- Sintaxis
 - date [-u --utc --universal] [MMDDhhmm[[CC]YY][.ss]]
- Despliega el tiempo actual del sistema en un determinado formato, o define la fecha y hora del sistema
 - posible definir el formato de salida
- Algunas opciones
 - d, --date=STRING: despliega tiempo descrito por STRING
 - r, --reference=FILE: despliega tiempo última
 - s, --set reference=STRING: asigna tiempo descrito por STRING
 - u: despliega tiempo UTC (Universal Time Coordinated)

Lámina 33

Dr. Roberto Gómez C.



Ejemplo comando date


```

$ date
Fri Jun 29 12:00:44 CDT 2007
$ date -u
Fri Jun 29 17:00:50 UTC 2007
$ date -r a1
Fri Jun 25 11:09:50 UTC 2007
$ date -d "11/20/2003 12:08:01"
Fri Jun 25 11:09:50 UTC 2007
$ date
Fri Jun 29 12:23:57 CDT 2007
$ date '+DATE: %m-%d-%y%nTIME: %H:%M:%S'
DATE: 06-29-07
TIME: 12:24:00
$ date '+DATE: %d.%m.%y%nTIME: %H:%M:%S'
DATE: 29.06.07
TIME: 12:24:03
$

```

Lámina 34


Dr. Roberto Gómez C.



Ejemplos uname y cal

```
$ uname
SunOS
$ uname -a
Linux localhost 2.4.20-8 #1 Thu Mar 13 17:54:28 EST 2003
i686 GNU/Linux
$ cal
May 2002
S M Tu W Th F S
    1 2 3 4
 5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
$
```


Lámina 35 Dr. Roberto Gómez C.



Comando mkdir

- Crea directorio(s) si no existe(n)
- Opciones
 - m: asigna permisos
 - p: no error si existen, crea directorios padres si es necesario
 - v: crea mensaje por cada directorio creado

Lámina 36 Dr. Roberto Gómez C.




Ejemplo opción `-p` comando `mkdir`

- Permite crear varios niveles de directorios al mismo tiempo
- Ejemplo:


```
$ ls -F
Reports/ dante* dante1* dir1/ dir2/ dir3/ dir4/ file.1*
file.2* file.3* file1* file2* file3* file4* fruit* fruit2*
logfile practice/ tutor.vi
$ mkdir -p practice2/dir1/admin
$ ls -F
Reports/ dante* dante1* dir1/ dir2/ dir3/ dir4/ file.1*
file.2* file.3* file1* file2* file3* file4* fruit* fruit2*
logfile practice/ practice2/ tutor.vi
$ ls -F practice2
dir1/
$ cd practice2
$ ls -F dir1
admin/
$
```


Lámina 37 Dr. Roberto Gómez C.



Comando `cd`

- Cambio de directorio
- Comando tipo built-in
- Parámetros especiales
 - “.” “ directorio actual
 - “..” “ directorio padre
 - “~” “ directorio hogar

Lámina 38 Dr. Roberto Gómez C.



Ejemplo pwd, mkdir y cd

```

$ pwd
/home/toto
$ ls
a1      a3      b1      b2
$ mkdir B
$ cp b1 b2 B/
$ ls
a1      a3      b1      b2
$ cd B
$ ls B
b1      b2
$ pwd
/home/toto/B
$


```

```

$ cd ..
$ pwd
/home/toto
$ mkdir A
$ mv a1 a3 A
$ ls
$ ls A
a1      a3
$

```


Lámina 39
Dr. Roberto Gómez C.



Comandos cp y mv

- Comando cp
 - Copia archivos y directorios
 - Algunas opciones
 - f: si archivo destino existente no puede ser abierto, lo borra e intenta de nuevo
 - i: pregunta antes de borrarlo
 - R, r: copia recursivamente
- Comando mv
 - renombra o mueve los archivos
 - Algunas opciones
 - u: solo lo mueve si el archivo fuente es más nuevo que el archivo destino o si este último no existe
 - f: sobre escribe si archivo destino existe
 - i: pregunta antes de sobre escribir


Lámina 40
Dr. Roberto Gómez C.



Ejemplo comando cp y mv

```
$ ls
a1      a2      a3
$ cp a1 b1
$ ls
a1      a2      a3      b1
$ mv a2 b2
$ ls
a1      a3      b1      b2
$
```


Lámina 41 Dr. Roberto Gómez C.



Borrando archivos: rm

- Sintaxis rm:
 - rm [-firv] archivo(s)
- borra el nombre de un archivo
- si ese nombre fuera el último (numero de ligas = 1), el archivo será "físicamente" suprimido.
- opciones:
 - f: ignora archivos no existentes y nunca previene
 - i: interactivo, previene al usuario
 - r: borra contenido de directorios recursivamente
 - v: imprime nombre archivo antes de borrarlo

Lámina 42 Dr. Roberto Gómez C.




Borrando directorios

- Sintaxis:
 - rmdir directorios
- Borra un directorio.
- No borra el directorio si este no se encuentra vacío.

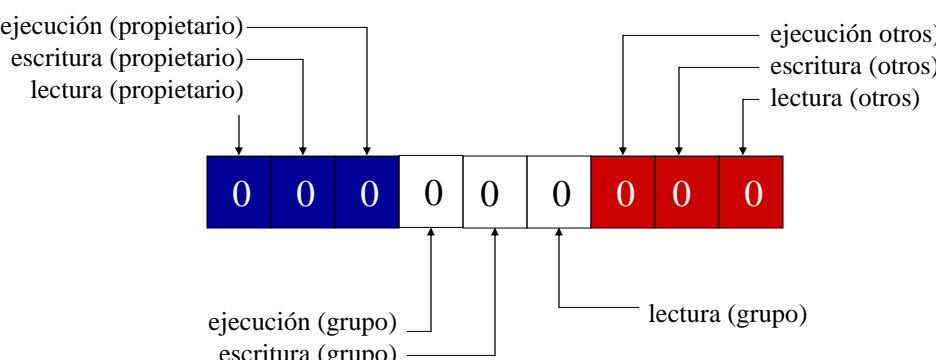
Lámina 43

Dr. Roberto Gómez C.



Los permisos de los archivos

- Archivos cuentan con permisos, el significado varia un poco entre archivos y directorios



ejecución (propietario)
escritura (propietario)
lectura (propietario)


ejecución otros
escritura (otros)
lectura (otros)

ejecución (grupo)
escritura (grupo)

lectura (grupo)

Lámina 44


Dr. Roberto Gómez C.



Significado permisos en directorios

- **r:**
 - autorización de leer el directorio (comando **ls**)
- **w:**
 - autorización de escribir en el directorio
 - (creación, modificación o supresión de archivos)
- **x:**
 - autorización para posesionarse en el directorio (comando **cd**)

Lámina 45
Dr. Roberto Gómez C.



Comando chmod

- Cambiar los permisos de un archivo
- Sintaxis


```
chmod permisos archivo
```
- Definiendo permisos
 - octal: permiso otorgado es un 1, sin otorgar es un cero
 - simbólico:

{

u
g
o
a

+ -

r
w
x

u: propietario (creador)


g: grupo

o: resto del mundo

a: todos

rwxrwxrwx	777
rwxrwxrw-	776
rwxrwxr--	774
rwxrwx---	770
rwxrwxr-x	775
rwxrwx--r	771


Lámina 46
Dr. Roberto Gómez C.



Ejemplos chmod

```
$ ls -l e1
-rw-rw-rw- 1 toto 0 Oct 12 18:20 e1
$ chmod 755 e1
$ ls -l e1
-rwxr-xr-x 1 toto 0 Oct 12 18:20 e1
$ chmod a-x e1
$ ls -l e1
-rw-r--r-- 1 toto 0 Oct 12 18:20 e1
$
```


Lámina 47
Dr. Roberto Gómez C.



Ejemplo ln

```
$ ls
pln1
$ more pln1
Esto
es
una
prueba
$ ls -l pln1
-rw-rw-r-- 1 toto toto 21 Sep 18 10:29 pln1
$ ln pln1 pln2
$ ls -l pln2
-rw-rw-r-- 2 toto toto 21 Sep 18 10:29 pln2
$ more pln2
Esto
es
una
prueba
$ echo this is a test >> pln2
```

Lámina 48
Dr. Roberto Gómez C.




Ejemplo ln (cont)

```

$ echo this is a test >> pln2
$ more pln2
Esto
es
una
prueba
this is a test
$ more pln1
Esto
es
una
prueba
this is a test
$ rm pln2
$ ll pln1
-rw-rw-r-- 1 toto  toto    36 Sep 18 10:31 pln1
$

```

Lámina 49 Dr. Roberto Gómez C.




Ejemplo opción -s de comando ln (liga simbólica)

```

$ ls
a1
$ more a1
Esto es una prueba
$ ln -s a1 a1.liga
$ ls -l
Total 4
-rw-r--r-- 1 rogoomez gomez  9 Feb 12 10:09  a1
lrwxrwxrwx 1 rogoomez gomez  4 Feb 23 17:24  a1.liga -> a1
$ echo This is a test >> a1.liga
$ more a1.liga
Esto es una prueba
This is a test
$ more a1
Esto es una prueba
This is a test
$

```

Lámina 50 Dr. Roberto Gómez C.



Ejemplo opción -s de comando ln (liga simbólica)


```

$ ls
a1 a1.liga
$ rm a1
$ ls -l
Total 0
lrwxrwxrwx 1 rogomez gomez 4 Feb 23 17:24 a1.liga -> a1
$ more a1.liga
a1.liga: Not such file or directory
$

$ ln -s /home/erick/grades /tmp/grades.old
$ cd /tmp/grades.old
$ pwd
/home/erick/grades
$

```

Lámina 51 Dr. Roberto Gómez C.




Ejemplo who, whoami, id

```

$ who
toto :0 Jul 21 18:48
toto pts/0 Jul 21 18:49 (:0.0)
cachafas pts/1 Jul 21 18:51 (:0.0)
solovino pts/2 Jul 21 18:51 (:0.0)
toto pts/3 Jul 21 18:53 (:0.0)
$ whoami
toto
$ id
uid=503(toto) gid=503(toto) groups=503(toto)
$ su rogomez
Password
$ id
uid=501(rogomez) gid=501(rogomez) groups=503(rogomez), 502(jesu)
$

```

Lámina 52 Dr. Roberto Gómez C.




Cambiando el password

- Abrir una terminal virtual
- En una terminal virtual ejecutar el comando `passwd`

```
$ passwd  
passwd: Changing password for user1  
Enter login password:  
New password:  
Re-enter new password:  
passwd (SYSTEM): passwd successfully changed for user1  
$
```

Lámina 53 Dr. Roberto Gómez C.



El archivo `/etc/passwd`

- Archivo ASCII manipulable con un editor
- Debe poder ser leído por todos los usuarios para ciertos comandos
- A cada usuario le corresponde una entrada
- Los programas realizan una búsqueda secuencial de las entradas (no vale la pena ordenar las entradas)
- Los campos de cada entrada están separados por carácter de dos puntos (:)

Lámina 54 Dr. Roberto Gómez C.



Campos de las entradas

- El identificador del usuario
- El password del usuario
- En algunas versiones: información sobre fecha del último cambio del password y sobre el periodo para realizar dichos cambios; dicha información separada por comas
- Un valor numérico, de 0 a 6000 que representa el UID del usuario

Lámina 55

Dr. Roberto Gómez C.



- Otro valor numérico, que no pase de 600, que representa el GID
- Un campo de comentarios conocido como GECOS
- El directorio hogar
- El shell de inicio

Lámina 56

Dr. Roberto Gómez C.



Ejemplo archivo /etc/passwd

```
xetaboada:ypK2awu1hBqGs:1326:41:Eunice Taboada Ibarra:/home/dacs/xetaboada:/bin/csh
dgonzalez:dU8Ml0KM7Af8Y:10106:41:John Lucien Gonzalez:/home/dacs/dgonzalez:/bin/csh
abermude:Fe5I/SHg53HM:2404:43:Adriana Diaz B Pagos:/home/prepa/abermude:/bin/csh
sa448020:iqC7X.6SUEASE:1832:215:David Bernal G Di Soporte:/home/sap/sa448020:/bin/csh
rcaballe:j3KODtAuQ8uEQ:8773:41:Ricardo Caballero Valdes:/home/dacs/rcaballe:/bin/csh
csanchez:YYoHIXDeYHanM:1212:43:Concepcion Sanchez:/home/prepa/csanchez:/bin/csh
sduenas:lube95PeMQZOQ:10140:41:Lic. Sergio F Rodriguez:/home/dacs/sduenas:/bin/csh
rperrin:rKWggQip3DIHQ:10021:44:Rafael Fausto:/home/dae/rperrin:/bin/csh
gperrin:Bj87cqMfSXzmc:10012:44:Graciela Patricia:/home/dae/gperrin:/bin/csh
rvilla:4McraxhY8AVB6:8839:43:Rafael Villa:/home/prepa/rvilla:/bin/csh
lvelio:lifTeZS98v/H.:1248:41:Lucrecia Velio-mejia:/home/dacs/lvelio:/bin/csh
tpacheco:UNbyYZ.dNCY3.:10275:510:Tito Omar:/home/dia/tpacheco:/bin/csh
jorozco:QOdtJnflY.1.s:3656:206:Jorge Orozco S:/home/unicom/jorozco:/bin/csh
bmerced:FzniebygQZSRs:1613:510:Bernando Isidro Merced S Dia :/home/dia/bmerced:/bin/csh
amoreno:lvoQAFGgLpxWg:5161:40:Asuncion Moreno 3122:/home/dia/amoreno:/bin/ksh
mahernan:AxBSyYy/tiHM6:1166:203:Magdalena Hernandez S:/home/dsa/mahernan:/bin/csh
aantunan:nhd5kmfXoGVP.:8937:41:Alma L Antunano Arias Dacs:/home/dacs/aantunan:/bin/csh
```

Lámina 57

Dr. Roberto Gómez C.




shutdown, init y runlevel

- Un sistema Linux no se arranca o detiene, sino que simplemente se cambia su nivel de ejecución.
- Durante un arranque normal, el sistema se coloca en el nivel 3 (multiusuario con red) o en el nivel 5 (análogo al 3 pero con el sistema de ventanas activo desde el inicio).
- Comandos:
 - shutdown -h now cambia el nivel actual al nivel 0 (halt).
 - shutdown -r now cambia el nivel actual al nivel 6 (reboot).
 - /sbin/init nivel cambia al nivel especificado
 - /sbin/runlevel indica el nivel de ejecución previo y el actual

Lámina 58

Dr. Roberto Gómez C.




El archivo inittab

- Es el primer archivo que es leído al arranque del sistema, contiene especificaciones sobre otros archivos deben de ser ejecutados, el nivel de arranque del sistema
- Administrado por init,
- Formato del archivo: **id : nivel : acción : procesos**

Campo	Significado
id	identificador línea en archivo
nivel	nivel ejecución (0-6)
acción	como ejecutar el proceso (respawn, wait, once, boot...)
procesos	proceso a ejecutar

Lámina 59
Dr. Roberto Gómez C.



Ejemplo archivo /etc/inittab

```

# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)

id:S:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud:once:/sbin/update

# Trap CTRL-ALT-DELETE
ca:ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left. Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf:powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# spawn a login shell into irda device
Tl:2345:respawn:/sbin/mgetty -r irc0m0


# Run xdm in runlevel 5
# xdm is now a separate service
x:S:respawn:/etc/X11/xdm -nodaemon
    
```

nivel en el que
arrancara por
default

indica que la entrada al sistema,
donde pide el login y el password,
se hará en forma gráfica,
y que el programa que lo
realiza es el xdm

Al acceder a nuestro sistema,
aparte de la pantalla de login
gráfica, hay 6 consolas en
modo texto (a las que se puede
acceder pulsando
CONTROL+ALT-(desde la
tecla F1 hasta la tecla F6, la
tecla F7 vuelve a acceder al
sistema gráfico) ejecutándose
en segundo plano.


Lámina 60



Niveles ejecución Linux

0	Halt	Nivel detiene el sistema
1	Single User	Modo administración
2	Multiuser	Modo funcionamiento normal sin algunos servicios de red
3	Multiuser + network	Como mod 2 pero con todos los servicios de red
4		Generalmente no utilizado
5	Modo gráfico multiusuario completo	Con una pantalla de inicio de sesión basada en X
6	Reboot	Se reinicia el sistema
s,S	Emergency single user	Igual al nivel 1 pero sin acceder a los archivos de configuración de inicio.

Lámina 61 Dr. Roberto Gómez C.



Los administradores arranque Linux

- **LILO**
 - Linux LOader
 - administrador de arranque que permite iniciar un sistema operativo (SO) cargando el sector de arranque de una partición del disco duro (o de un disquete).
- **GRUB**
 - GNU GRand Unified Boot loader
 - programa que permite al usuario seleccionar qué sistema operativo instalado deseamos arrancar en el momento de arranque del sistema.
 - también permite que el usuario pase argumentos al kernel.

Lámina 62 Dr. Roberto Gómez C.



Scripts inicialización

- Cuando arranca nivel ejecución 5,
 - programa init ejecuta el script `/etc/rc.d/rc 5`.
 - script consulta el directorio `/etc/rc.d/rc5.d/` para determinar qué procesos iniciar o parar.
- De forma general, existe un directorio `/etc/rc.d/rc<x>.d/`, por cada nivel de ejecución definido por el sistema,
 - se encuentran los servicios que deberán ser lanzados y parados en ese nivel de ejecución.

Lámina 63

Dr. Roberto Gómez C.




Listado directorio `/etc/rc.d/rc3.d`

```
$ pwd
/etc/rc.d
$ ls -l rc3.d/
init.d rc rc0.d rc1.d rc2.d rc3.d rc4.d rc5.d rc6.d rc.local rc.sysinit
$ ls -l rc3.d/
total 0
lrwxrwxrwx 1 root root 13 Apr 1 1998 K15gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 13 Apr 1 1998 K60lpd -> ../init.d/lpd
lrwxrwxrwx 1 root root 15 Apr 1 1998 K95nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 17 Apr 1 1998 S01kerneld -> ../init.d/kerneld
lrwxrwxrwx 1 root root 17 Apr 1 1998 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 16 Apr 1 1998 S20random -> ../init.d/random
lrwxrwxrwx 1 root root 16 Apr 1 1998 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 13 Apr 1 1998 S40atd -> ../init.d/atd
lrwxrwxrwx 1 root root 15 Apr 1 1998 S40crond -> ../init.d/crond
lrwxrwxrwx 1 root root 18 Apr 1 1998 S75keytable -> ../init.d/keytable
lrwxrwxrwx 1 root root 11 Apr 1 1998 S99local -> ../rc.local
```

Lámina 64

Dr. Roberto Gómez C.




Comandos which y whereis

```

$ date
Wed Jul 21 19:44:33 CDT 2004
$ which date
/bin/date
$ whereis date
date: /bin/date /usr/share/man/man1/date.1.gz
$

```

Lámina 65 Dr. Roberto Gómez C.




Ejemplo alias

```

$ alias
alias h='history'
alias l='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mc='./usr/share/mc/bin/mc-wrapper.sh'
alias vi='vim'
$ more a1
Soy el archivo a1
$ alias m=more
$ m a1
Soy el archivo a1
$ unalias m
$ m a1
bash: m: command not found
$

```

Lámina 66 Dr. Roberto Gómez C.



Ejemplo history


```

$ ps
  PID TTY          TIME CMD
 1225 pts/1    00:00:00 bash
 1373 pts/1    00:00:00 ps
$ who
toto  :0          Jul 26 21:02
toto  pts/0      Jul 26 21:02 (:0.0)
$ date
Mon Jul 26 21:09:48 CDT 2004
$ history
 1 ps
 2 who
 3 date
 4 history
$ !d
date
Mon Jul 26 21:11:06 CDT 2004
$

```

Lámina 67

Dr. Roberto Gómez C.



Ejemplo ps


```

$ ps
  PID TTY          TIME CMD
 1225 pts/1    00:00:00 bash
 1397 pts/1    00:00:00 ps
$ ps -e
  PID TTY          TIME CMD
  1 ?          00:00:04 init
  2 ?          00:00:00 keventd
  3 ?          00:00:00 kapmd
  4 ?          00:00:00 ksoftirqd_CPU0
  :
  :
1195 pts/0    00:00:00 bash
1225 pts/1    00:00:00 bash
1268 pts/0    00:00:00 vim
1400 pts/1    00:00:00 ps
$
$ ps -f
UID      PID PPID  C STIME TTY          TIME CMD
toto    1225 1193  0 21:02 pts/1    00:00:00 bash
toto    1399 1225  0 21:13 pts/1    00:00:00 ps -f

```

Lámina 68


Dr. Roberto Gómez C.



Ejemplo kill

```
$ netscape &
[1] 1321
$ ps
PID TTY      TIME CMD
1226 pts/1    00:00:00 bash
1321 pts/1    00:00:00 netscape-commun
1345 pts/1    00:00:00 netscape-commun
1348 pts/1    00:00:00 ps
$ kill 1321
$ ps
PID TTY      TIME CMD
1226 pts/1    00:00:00 bash
1349 pts/1    00:00:00 ps
$
```


Lámina 69
Dr. Roberto Gómez C.



El comando jobs

Comando	Valor
jobs	despliega los trabajos (jobs) que se encuentran actualmente corriendo
bg %n	pone el trabajo n en background
fg %n	pone el trabajo n “al frente” (foreground)
^Z	detiene el trabajo “del frente” (foreground)
stop %n	detiene el trabajo n en background

Lámina 70
Dr. Roberto Gómez C.



El comando top

- Despliega la actividad de los procesos ejecutándose en el CPU a tiempo real
- Proporciona interfaz para manipular procesos.

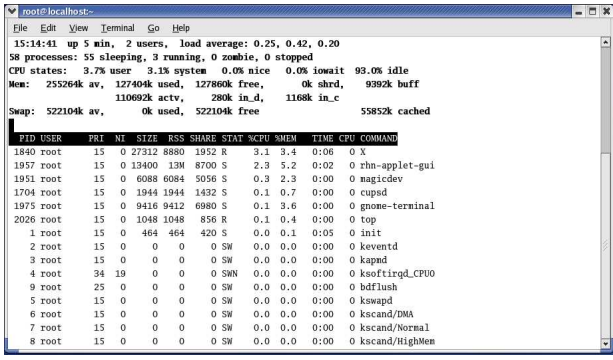



Lámina 71 Dr. Roberto Gómez C.



Ejemplo tty

\$ tty

/dev/pts/4

\$

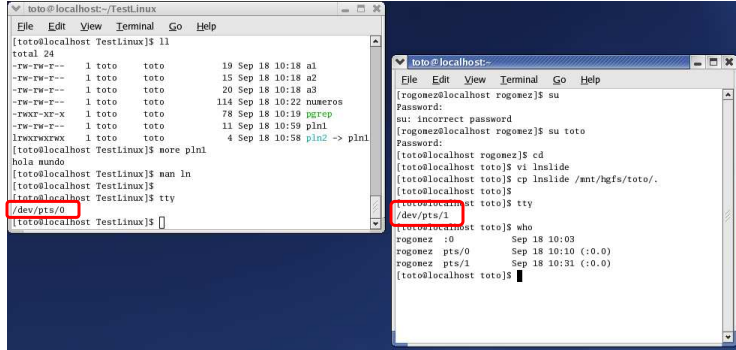


Lámina 72 Sun Sep 18 11:02 AM



Comando ifconfig

- Comando de manipulación de la tarjeta de red.
- Solo root lo puede ejecutar directamente.
- Para ejecutarlo como usuario sin privilegios es necesario introducir el path completo.

`/sbin/ifconfig`

- Sin parámetros despliega la dirección física y lógica asociada a cada uno de los puertos de red de la máquina.

Lámina 73

Dr. Roberto Gómez C.



Ejemplo salida ifconfig


```
$ /sbin/ifconfig
eth0 Link encap:Ethernet HWaddr 00:0C:29:B9:F9:82
      inet addr:10.48.50.250 Bcast:10.48.55.255 Mask:255.255.248.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:29 errors:0 dropped:0 overruns:0 frame:0
      TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:2897 (2.8 Kb) TX bytes:684 (684.0 b)
      Interrupt:10 Base address:0x1080

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:19051 errors:0 dropped:0 overruns:0 frame:0
      TX packets:19051 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:1300986 (1.2 Mb) TX bytes:1300986 (1.2 Mb)

$
```

Lámina 74

Dr. Roberto Gómez C.



Ejemplos grep


```

$ more /etc/group
root::0:root
bin::2:root,bin,daemon
sys::3:root,adm,daemon
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
daemon::12:root,daemon
$ grep adm /etc/group
sys::3:root,adm,daemon
adm::4:root,adm,daemon
tty::7:root,tty,adm
lp::8:root,lp,adm
$
$ grep -n root /etc/group
1:root::0:root
2:bin::2:root,bin,daemon
4:sys::3:root,adm,daemon
5:adm::4:root,adm,daemon
6:uucp::5:root,uucp
7:mail::6:root
8:tty::7:root,tty,adm
9:lp::8:root,lp,adm
10:nuucp::9:root,nuucp
12:daemon::12:root,daemon
$ cd /etc
$ ls
group hosts passwd
$ grep -l root group passwd hosts
group
passwd
$
$ grep -v root group
other::1:
staff::10:
sysadmin::14:
nobody::60001:
noaccess::60002:
nogroup::65534:
...
$ grep -c root group
10
$ grep -i MAIL group
mail::6:root
$

```

grep = Global Regular Expression and Print


Lámina 75
Dr. Roberto Gómez C.



Opciones comando grep

Opción	Definición
-i	ignora mayúsculas, minúsculas y mayúsculas son considerados idénticos
-l	solo lista los nombres de los archivos que coincidan con lo buscado
-n	precede cada líneas con su número línea relativo
-v	invierte búsqueda, solo despliega líneas que NO concuerdan con lo buscado
-c	solo imprime el número de líneas que concuerdan
-w	realiza la búsqueda como una palabra, ignora aquellas concordancias que son substings de palabras más grandes

Lámina 76
Dr. Roberto Gómez C.



Los pipes

- Una tubería o pipe es una combinación de varios comandos que se ejecutan simultáneamente, donde el resultado del primero se envía a la entrada del siguiente. Este tipo de comunicación entre procesos se especifica mediante el carácter barra vertical, '|’.
- Sintaxis

`comando1 | comando2 | ... | comandon`






Lámina 77
Dr. Roberto Gómez C.



Redireccionamiento

- El redireccionamiento significa hacer que la shell cambie lo que está considerado como entrada estándar o el lugar donde va a parar la salida estándar.
- Para redireccionar la salida estándar, se usa el símbolo `>`.
 - si el archivo existe lo reemplaza con el resultado del comando
- El símbolo `>>` permite reañadir a la salida estándar
- El símbolo `2>` permite redireccionar la salida errores estándar.

Lámina 78
Dr. Roberto Gómez C.



Ejemplos redireccionamientos


```

$ ls
a1 a2 a3
$ ls > salida
$ ls
a1 a2 a3 salida
$ more salida
a1
a2
a3
$ echo hola mundo
hola mundo
$ echo hola mundo >> salida
$ more salida
a1
a2
a3
hola mundo
$
                    
```

```

$ ls b*
ls: b*: No such file or directory
$ ls b* > salidita
ls: b*: No such file or directory
$ ls
a1 a2 a3 salida salidita
$ more salidita
$ ls b* 2> error
$ ls
a1 a2 a3 error salida salidita
$ more error
ls: b*: No such file or directory
$ echo uno dos tres > a1
$ more a1
uno dos tres
$ more < a1
uno dos tres
$
                    
```

Lámina 79
Dr. Roberto Gómez C.



El comando tee

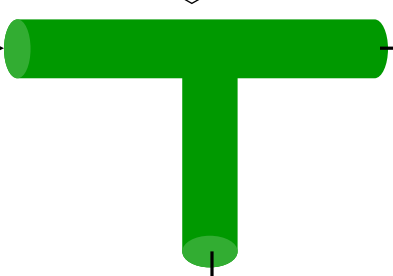
comando

tee filename

comando

stdin

→




→

stdout

↓

filename

Lámina 80
Dr. Roberto Gómez C.




Ejemplo comando tee

```

$ ls | tee salida
dante dante1 dir1  dir2  dir3  dir4  file.1  file.2  file.3
file1  file2  file3  file4  fruit  fruit2  practice tutor.vi
$ more salida
dante dante1 dir1  dir2  dir3  dir4  file.1  file.2  file.3
file1  file2  file3  file4  fruit  fruit2  practice tutor.vi
$ cal | tee -a resultado
      August 2000
S  M   Tu  W  Th  F   S
    1  2  3  4  5
6   7  8  9  10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
$

```


Lámina 81 Dr. Roberto Gómez C.



Localizando archivos comando find

- Formato comando
 - find pathname(s) expression(s) action(s)
 - pathname
 - path del directorio donde empezara la busqueda
 - expression
 - criterio de busqueda
 - si la expresión es verdadera, la acción especificada se llevará a cabo


Lámina 82 Dr. Roberto Gómez C.



Expresiones comando find

Expresión	Busca archivos que
-name filename	concurden con el nombre
-size [+ -] n	mayores que +n, menores -n o iguales a n
-atime [+ -] n	accedidos mas de +n días, menores -n días y exactamente n días
-mtime [+ -] n	modificados mas de +n días, menores -n días y exactamente n días
-user loginID	tengan propietario a loginID
-type	concurden con un tipo archivo (f,d,s)
-perm	cuenten con ciertos permisos

Lámina 83 Dr. Roberto Gómez C.



Acciones comando find

Acción	Definición
-exec command { } \;	ejecuta command a cada archivo encontrado. Los corchetes {}, delimita donde se pasa el archivo como argumento. Espacio, backslash y punto y coma (;) delimita el final del comando
-ok command { } \;	especifica la forma interactiva de -exec. Requiere entrada antes que find aplique el command al archivo,
-print	imprime el path completo en la salida estándar, es el default
-ls	imprime el pathname con todas sus características

Lámina 84 Dr. Roberto Gómez C.



Ejemplos uso comando find

- Búsqueda archivos llamado core, desde directorio raíz
`$ find / -name core`
- Búsqueda archivos llamados core, desde directorio hogar y borrarlos cuando se encuentran
`$ find ~ -name core -exec rm {} \;`
- Archivos, desde directorio trabajo, que no han sido modificados en los últimos 90 días
`$ find . -mtime +90`
- Archivos mayores que 57 bloques (512-byte blocks) a partir directorio hogar
`$ find ~ -size +57`

Lámina 85

Dr. Roberto Gómez C.




Más ejemplos

- Archivos cuyo nombre termina con “tif”, a partir del directorio /usr
`$ find /usr -name `*tif``

Lámina 86

Dr. Roberto Gómez C.



Ejemplo wc

```


$ cat numeros
uno      un       one
dos      deux    two
tres     trois   three
cuatro   quatre  four
cinco    cinq    five
seis     six     six
siete    sept    seven
ocho     huit    eighth
nueve    neuf    nine
diez     dix     ten

$ wc numeros
  10  30  189 numeros

$

```


Lámina 87 Dr. Roberto Gómez C.



Ejemplo head

<pre> \$ cat meses enero febrero marzo abril mayo junio julio agosto septiembre octubre noviembre diciembre \$ </pre>		<pre> \$ head meses enero febrero marzo abril mayo junio julio agosto septiembre octubre \$ </pre>
---	--	--

Lámina 88 Dr. Roberto Gómez C.



Ejemplo tail

```

$ cat meses
enero
febrero
marzo
abril
mayo
junio
julio
agosto
septiembre
octubre
noviembre
diciembre
$


```

```

$ tail meses
marzo
abril
mayo
junio
julio
agosto
septiembre
octubre
noviembre
diciembre
$

```

Lámina 89
Dr. Roberto Gómez C.



Ejemplo sort

```

$ cat arch1
Anne      48486
Jamie    48481
Fred      48487
Sondra    48483
Janet     48482
$ sort arch1
Anne      48486
Fred      48487
Jamie    48481
Janet     48482
Sondra    48483
$


```

```

$ sort +1n arch1
Jamie    48481
Janet    48482
Sondra   48483
Anne     48486
Fred     48487
$

```


Lámina 90
Dr. Roberto Gómez C.



Opciones comando sort

Opción	Definición
-n	realiza un sort numérico
(+ -) -n	empieza (+n) o termin (-n) con campo que sigue a n
-r	invierte el orden del sort
-f	ignora mayúsculas al acomodar
-M	sorts los primeros tres caracteres del campo como el nombre de mes abreviado
-d	usa orden diccionario, solo letras, dígitos y espacios son comparados, el resto es ignorado.
-o filename	imprime resultados en el filename
-b	ignora blancos en posiciones iniciales/finales llave sort
-t char	utiliza char como el carácter separador de campos

Lámina 91
Dr. Roberto Gómez C.




Ejemplo cut

```

$ cat passwd
root:x:0:0:root:/root:/bin/bash
murie:x:500:500:M Muriel Cordero:/home/murie:/bin/bash
practica:x:501:501:Usuario practicas:/home/practica:/bin/ksh
wizardi:x:502:502:Wizard nethack:/home/wizard:/bin/bash
$ cut -f1,7 -d: passwd
root:/bin/bash
murie:/bin/bash
practica:/bin/ksh
wizardi:/bin/bash
$

```

Lámina 92
Dr. Roberto Gómez C.



Ejemplo paste

```

$ cat num-esp
uno
dos
tres
cuatro
cinco
$ cat num-fra
un
deux
trois
quatre
cinq
$


```

```

$ paste num-esp num-fra
uno un
dos deux
tres trois
cuatro quatre
cinq cinq
$

```

Lámina 93
Dr. Roberto Gómez C.



Ejemplo uniq

```

$ uniq nums
uno
dos
tres
cuatro
cinco
uno
$ sort nums > ordena
$ more ordena
cinco
cuatro
dos
tres
uno
uno
$


```

```

$ uniq ordena
cinco
cuatro
dos
tres
uno
$

```

Lámina 94
Dr. Roberto Gómez C.




Ejemplo tr

```

$ cat prueba
Esto es una PRUEBA
$ tr 'a-z' 'A-Z' < prueba
ESTO ES UNA PRUEBA
$ tr 'A-Z' 'a-z' < prueba
esto es una prueba
$ cat toto
No es lo mismo @ que *
$ tr '@' '*' < toto
No es lo mismo * que *
$ tr '*' '@' < toto
No es lo mismo @ que @
$

```

Lámina 95
Dr. Roberto Gómez C.




Ejemplo df

```

$ df
Filesystem      1k-blocks  Used Available Use% Mounted on
/dev/sda2        4538156 2635296 1672328 62% /
/dev/sda1         101089    9425   86445 10% /boot
none             127632     0   127632 0% /dev/shm
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       4.4G  2.6G  1.6G  62% /
/dev/sda1       99M   9.3M   85M  10% /boot
none            125M   0   125M  0% /dev/shm
$

```

Lámina 96
Dr. Roberto Gómez C.




Ejemplo du

```

$ du
4  ./gnome2/accels
8  ./gnome2/share/fonts
8  ./gnome2/share/cursor-fonts
:
:
3216 ./Cursos/AlgosDist/Apuntes/Libro
480 ./Cursos/AlgosDist/Apuntes/Policopias
4380 ./Cursos/AlgosDist/Apuntes
100 ./Cursos/AlgosDist/Examenes
92  ./Cursos/AlgosDist/Figuras
60  ./Cursos/AlgosDist/Lynch
4   ./gqview/thumbnails
4   ./gqview/collections
24  ./gqview
172840 .
$

```


Lámina 97 Dr. Roberto Gómez C.



Comandos comparación archivos

- **cmp**
 - compara dos archivos e indica , si la hay , el lugar donde se produce la primera diferencia (número de caracter y línea de la diferencia)
- **comm**
 - su salida se produce en 3 columnas. La primera contiene las líneas únicas del primer archivo, la segunda las únicas del segunda y la tercera las comunes.
 - parámetros -1 , -2 y -3 indican que no visualizen la primera, segunda y tercera columna.
 - problema que tiene comm es que espera que las líneas estén ordenadas alfabéticamente, o en la misma posición dentro del archivo
- **diff**
 - compara el archivo original y el nuevo línea a línea e imprime el resultado en la salida estándar en un formato específico
 - formato: comandos vi para igualar archivos

Lámina 98 Dr. Roberto Gómez C.




Ejemplo comm antes sort

```

$ more coches1
Renault
Peagout
Lamborgini
Ford
Ferrari
BMW
Mercedes
$ more coches2
Feat
Cooper
Peagout
Lamborgini
Chevrolet
Chrysler
Tzuru
BMW
$

$ comm coches1 coches2
Feat
Cooper
Peagout
Lamborgini
Chevrolet
Chrysler
Renault
Peagout
Lamborgini
Ford
Ferrari
BMW
Mercedes
Tzuru
BMW
$
    
```

Lámina 99 Dr. Roberto Gómez C.



Ejemplo comm después sort


```

$ sort coches1 > c1
$ sort coches2 > c2
$ more c1
BMW
Ferrari
Ford
Lamborgini
Mercedez
Peagout
Renault
$

$ more c2
BMW
Chevrolet
Chrysler
Cooper
Feat
Lamborgini
Peagout
Tzuru
$

$ comm c1 c2
BMW
Chevrolet
Chrysler
Cooper
Feat
Ferrari
Ford
Mercedez
Renault
Tzuru
$
    
```

Lámina 100 Dr. Roberto Gómez C.



Ejemplo comando cmp

```

$ more n1
uno
dos
tres
cuatro
cinco
$ more n2
uno
dos
tres
cuatre
cinq
$ cmp n1 n2
n1 n2 differ: byte 14, line 4
$


```

```

$ cat a.fil
Este archivo con tres líneas.
Es casi igual al otro archivo,
pero alguna palabra es diferente.
$ cat b.fil
Este archivo con tres líneas.
Es casi igual al otro archivo,
pero alguna palabra es cambiada.
$ cmp a.fil b.fil
a.fil b.fil differ: char99, line 3
$

```

Lámina 101
Dr. Roberto Gómez C.



Ejemplo diff

```

$ more frutas1
manzana
naranja
nuez
$ more frutas2
manzana
naranja
uva
$ more frutas3
naranja
nuez
melon
$ diff frutas1 frutas2
3c3
< nuez
-----
> uva
$

```

```

$ diff -e frutas1 frutas2
3c
uva
.
$ diff frutas1 frutas3
1d0
< manzana
3a3
> melon
$ diff -e frutas1 frutas3
3a
melon
.
1d
$

```

Lámina 102
Dr. Roberto Gómez C.



Segundo ejemplo diff

\$ diff antigua nueva

3c3

< El Hobbit

> El Señor de los Anillos

78a79,87

> Tres anillos para los Reyes Elfos bajo el cielo.

> Siete para los Señores Enanos en casas de piedra.

> Nueve para los Hombres Mortales condenados a morir.

> Uno para el Señor Oscuro, sobre el trono oscuro

> en la tierra de Mordor donde se extienden las Sombras.

> Un Anillo para gobernarlos a todos. Un Anillo para encontrarlos,

> un anillo para atraerlos a todos y atarlos a las tinieblas

> en la tierra de Mordor donde se extienden las Sombras.

\$

3c3 expresa que se debe cambiar la línea 3, quitando "El Hobbit" y substituyendolo por "El Señor de los Anillos". 78a79,87 indica que se debe insertar unas nuevas líneas 79 a la 87.

Lámina 103

Dr. Roberto Gómez C.




Comando awk

- Busca y procesa un patrón en un archivo
- Busca en uno o más archivos para ver si contienen líneas que coincidan con patrones específicos y después realiza ciertas acciones
- Puede usarse para generar reportes o filtrar textos
- Trabaja de la misma forma para números y texto, cuando se mezclan, awk no tiene problemas
- Dos formatos:
 - awk [-Fc] -f archivo-programa [lista-archivos]
 - usa un archivo-programa donde se almacena el programa awk
 - lista-archivos: pathnames que awk va a procesar
 - awk programa [lista-archivos]
 - el programa awk se pasa a través de la línea de comandos
 - posible escribir programas awk simples y cortos

Lámina 104

Dr. Roberto Gómez C.



Ejemplo awk

```


$ cat carros
plym  fury           77    73           2500
chevy nova            79    60           3000
ford  mustang        65    45          10000
volvo gl              78   102           9850
ford  ltd             83    15          10500

$ awk '{ print $3 $1}' carros
77 plym
79 chevy
65 ford
78 volvo
83 ford

$ awk '{ print $3 $0}' carros
77 plym      fury           77    73           2500
79 chevy     nova            79    60           3000
65 ford      mustang        65    45          10000
78 volvo     gl              78   102           9850
83 ford      ltd             83    15          10500

```


Lámina 105 \$ Dr. Roberto Gómez C.



Comando sed

- Edita un archivo de forma no interactiva
- Comandos almacenados en un archivo tipo script
 - posible introducir comandos desde línea comandos
- Por default copia líneas de la lista de archivos a la salida estándar, editando dichas líneas durante el proceso
- Sintaxis: sed [-n] -f script-file [lista-archivos]
 - sin la opción -f sed usa el primer argumento de la línea de comandos como script-file
 - f: provoca sed lea el script-file del archivo especificado como primer argumento
 - n: no print. La utileria sed no copia argumentos a la salida estándar excepto cuando se especifica por la instrucción Print (p)
- script de sed: líneas con formato
[direc[,direc]] instrucción [lista-argumentos]

Lámina 106 Dr. Roberto Gómez C.



Ejemplo sed

```

$ cat new
Line one.
The second line.
The third.
This is line four.
Five.
This is the sixth sentence.
This is line seven.
Eighth and last.
$ sed -n '/line/ p' new
The second line.
This is line four.
This is line seven.
$


```

```

$ sed -n '3,6 p' new
The third.
This is line four.
Five.
This is the sixth sentence.
$ sed '5 q' new
Line one.
The second line.
The third.
This is line four.
Five.
$

```

Lámina 107
Dr. Roberto Gómez C.



Metacaracteres del shell


- Carácter tilde: ~
 - directorio hogar
 - usando ~ username
 - usando ~+ y ~-
- Carácter dash: -
 - cambio entre directorios específicos
- Carácter asterisco: *
- Carácter signo interrogación: ?
- Los corchetes: []

```

$ pwd
/export/home/user1
$ cd /tmp
$ pwd
/tmp
$ cd -
/export/home/user1
$ cd -
/tmp
$

```

Lámina 108
Dr. Roberto Gómez C.



Ejemplos (~+) y (~-)


- El tilde y el signo de mas (~+) se refiere al directorio de trabajo actual
- El tilde y el signo de menos (~-) se refiere al directorio de trabajo anterior

```

$ pwd
/export/home/user1
$ cd dir 2
$ ls ~+
beans notes recipes
$ cd ~-
$ pwd
/export/home/user1
$ cd ~-
$ pwd
/export/home/user1/dir2
$

```

Lámina 109 Dr. Roberto Gómez C.



Ejemplo []

```

$ ls [a-f]*
dante          file.1  file.3  file2   file4   fruit2
dante_1        file.2  file1   file3   fruit

dir1:
cofees


dir2:
beans noit

dir3:
planets

dir4:
flowers
$ ls [af]*
file.1  file.2  file.3  file1  file2  file3  file4  fruit  fruit2
$

```


Lámina 110 Dr. Roberto Gómez C.



Ejemplo varios comandos en una línea con metacaracter: ;

```
$ date  
Thu May 30 15:24:03 MDT 2002  
$ uname  
SunOS  
$ date; uname  
Thu May 30 15:24:03 MDT 2002  
SunOS  
$
```

Lámina 111 Dr. Roberto Gómez C.



Permisos por default y umask

- Abreviación de user file creation mode mask
- El número octal de cuatro digitos que Unix usa para determinar los permisos de los nuevos archivos creados.
- Especifica los permisos que *no se quiere* que tengan los nuevos archivos y directorios.
- Comando trabaja haciendo un AND de bits con el complemento de umask
- Por default los archivos se crean con permiso 666 y los directorios con 777

Lámina 112 Dr. Roberto Gómez C.



El comando umask

- Es un comando interno (built-in) de sh, ksh y csh
- Los valores más comunes son 022, 027 y 077
- Un valor de 022 permite que lea y escriba todos los archivos recién creados, y el resto solo puede leerlos.
 - 0666 (modo de creación por default)
 - 0022 (valor de umask)
 - 0644 (modo resultante)
- Con 077 solo el propietario puede leer y escribir los archivos creados
 - 0666 (modo de creación por default)
 - 0077 (valor de umask)
 - 0600 (modo resultante)

Lámina 113

Dr. Roberto Gómez C.



El comando umask (cont)

- Una forma simple de calcular valores de umask es recordar que:
 - valor 2: apaga el permiso de escritura
 - valor 7: apaga permisos lectura, escritura y ejecución
- Si se está usando ksh, se puede asignar el valor de umask de forma simbólica, tal y como se hace con el comando chmod

```
emata@francia:7> umask u=rwx, g=x
emata@francia:8> umask 067
```

Lámina 114

Dr. Roberto Gómez C.



El sticky bit

- Aplicable en archivos ejecutables
- Le indica a Unix que deje el ejecutable en memoria después de que esta haya terminado su ejecución
- Dejando el programa en memoria, reduce el tiempo para otros usuarios (en teoría)
- Fue una interesante idea hace tiempo, pero es obsoleta hoy en día
 - técnicas memoria virtual la hacen innecesaria
 - paginación hace que ya no se use

Lámina 115

Dr. Roberto Gómez C.




Sticky bit y los directorios

- Si un usuario tiene permiso escritura en un directorio puede renombrar o borrar archivos en él (aunque no le pertenezcan)
- Varias nuevas versiones de Unix tiene una forma de impedir lo anterior
- El propietario del directorio puede activar el sticky bit
- Los usuarios que pueden renombrar o borrar archivos en dicho subdirectorio son:
 - el propietario del archivo
 - el propietario del directorio
 - el superusuario

Lámina 116

Dr. Roberto Gómez C.



Ejemplo uso sticky bit en directorios

```


egarcia> mkdir proyecto
egarcia> chmod 777 proyecto
egarcia> ls -ld
drwxrwxrwx  2  egarcia profes      32 Sep 23  19:30 proyecto

/* usuario jvazquez borra un archivo que no le pertenece */

jvazquez> cd /home/usr/egarcia/proyecto
ls -lg
total 3
-rw-r--r--  1  rogomez  profes  120 Sep 23  19:23 data.rogomez
-rw-r--r--  1  jvazquez profes 3421 Sep 24  20:03 data.jvazquez
-rw-r--r--  1  egarcia   profes  728 Sep 25  01:34 data.egarcia
-rw-r--r--  1  aortiz   profes  716 Sep 27  12:52 data.aortiz
jvazquez> rm data.aortiz

```

Lámina 117 Dr. Roberto Gómez C.



```

jvazquez> ls -lg
total 2
-rw-r--r--  1  rogomez  profes  120 Sep 23  19:23 data.rogomez
-rw-r--r--  1  jvazquez  profes  3421 Sep 24  20:03 data.jvazquez
-rw-r--r--  1  egarcia   profes  728 Sep 25  01:34 data.egarcia

egarcia> chmod 1777 proyecto
egarcia> ls -ld
drwxrwxrwx  2  egarcia profes      32 Sep 23  19:30 proyecto

jvazquez> rm data.rogomez
data.rogomez: 644 mode ? y
rm: data.rogomez not removed
Permission denied
jvazquez>

```

Lámina 118 Dr. Roberto Gómez C.



Los usuarios y los procesos

- Procesos pertenecen a un solo y único usuario
- El propietario es el que lanzó el proceso
 - puede enviarle señales y, en consecuencia, matarlo
- Para lanzarlo debe poseer los permisos de ejecución del archivo que contiene el código binario

Lámina 119

Dr. Roberto Gómez C.



- La “propiedad” del archivo del código no influye en la del proceso
 - usuario toto ejecuta código de un archivo que pertenece a cachafas
 - el proceso pertenece a usuario toto
- Esto es limitativo
 - se desea permitir a un usuario modificar el contenido de un archivo sin darle derecho de escritura en él
 - ejemplo archivo /etc/passwd, un usuario debe poder cambiar su password sin poder modificar el archivo que lo contiene

Lámina 120

Dr. Roberto Gómez C.



El bit Set UID (SUID)

- Derecho complementario de un proceso que condiciona la propiedad del proceso que ejecuta su código
- Retomando el ejemplo anterior:
 - si usuario cachafas activa el bit SUID del archivo
 - el usuario toto es el propietario del archivo, pero el propietario efectivo es cachafas
 - toto adquiere los derechos de cachafas durante el tiempo que dure la ejecución del proceso

Lámina 121

Dr. Roberto Gómez C.



Cuidados del bit SUID

- El bit SUID puede representar un hoyo en la seguridad del sistema
- Es necesario minimizar el número de archivos que pertenezcan al super-usuario y que tengan activado el bit SUID
- Algunas versiones de Unix ignoran el bit SUID y SGID en scripts, solo programas compilados pueden tenerlo activo

Lámina 122

Dr. Roberto Gómez C.



El bit Set Group ID (SGID)

- Mismo principio que SUID pero para grupos
- Ejecutar un archivo con bit SGID activo asigna el ID de grupo del usuario al mismo que el del archivo ejecutado, durante el tiempo que dura la ejecución de este
- Archivos con SGID o SUID activo pierden sus propiedades especiales cuando son copiados

Lámina 123

Dr. Roberto Gómez C.




Ejemplo bits SUID y SGID

```
rogomez@armagnac:3> ls -l /usr/bin/passwd /usr/bin/login
                        /usr/bin/mailx /etc/passwd
-rw-r--r--  1 root      752 Oct 22 1998 /etc/passwd
-r-sr-xr-x  1 root      29192 Jul 15 1997 /usr/bin/login*
-r-x--s-x   1 bin       127540 Jul 15 1997 /usr/bin/mailx*
-r-sr-sr-x  3 root      96796 Jul 15 1997 /usr/bin/passwd*
rogomez@armagnac:4>
```

Lámina 124

Dr. Roberto Gómez C.




Comando chmod: SGID, SUID, sticky bit

`chmod n777 a1`

Valor n	Efecto	Ejemplo	Resultado ls -l a1
1	Activar sticky bit	<code>chmod 1777 a1</code>	<code>-rwxrwxrwt</code>
2	Activar SGID	<code>chmod 2777 a1</code>	<code>-rwxrwsrwt</code>
4	Activar SUID	<code>chmod 4777 a1</code>	<code>-rwsrwxrwt</code>
6	Activar SUID y SGID	<code>chmod 6777 a1</code>	<code>-rwsrwxswt</code>
0	Desactivar sticky bit, SUID y SGID	<code>chmod 0777 a1</code>	<code>-rwxrwxrwx</code>


Lámina 125
Dr. Roberto Gómez C.



Las listas de control de acceso (acl)

- Algunos sistemas que cumplen con el libro Naranja, han cambiado el sistema de protección de permisos al de listas de control de acceso.
- Se basa en el concepto de derechos sobreentendidos
- Se trata de afinar la noción de permiso a usuarios o grupos específicos
- Se puede dotar de permisos de rwx a un determinado usuario o grupo de usuarios
- Existen en Unix desde hace más de diez años
- No todas las versiones de Unix lo soportan

Lámina 126
Dr. Roberto Gómez C.



Comandos


- Dos comandos
 - getfacl archivo
 - despliega el ACL del archivo
 - setfacl archivo
 - opción -m: modificación del ACL


```
user:<user name>:rwx
group:<group name>:rwx
other:rwx
```
 - opción -x: borrar entradas ACL


```
setfacl -x g:staff file
```
- Necesario montaje apropiado


```
mount -o remount,acl /
```

Lámina 127
Dr. Roberto Gómez C.



Ejemplo uso ACLs

Desplegando permisos de forma normal:

```
root@cachafas:2# ls -l /usr/local/sshd
- -rwx----- 1 root bin 2616160 Apr 28 1997 /usr/local/sshd
root@cachafas:3#
```

Verificando permisos con comando getfacl:

```
root@cachafas:3# getfacl /usr/local/sshd
# file: /usr/local/sshd
# owner: root
# group: bin
user::rwx
group::---
mask:---
other:---
root@cachafas:4#
```

Lámina 128
Dr. Roberto Gómez C.



Extendiendo los permisos

```
root@cachafas:4# setfacl -m user:toni:r-x /usr/local/sshd
root@cachafas:5# getfacl /usr/local/sshd
# file: /usr/local/sshd
# owner: root
# group: bin
user::rwx
user:toni:r-x
group:---
mask:---
other:---
root@cachafas:6#
```

Lámina 129

Dr. Roberto Gómez C.




Advertencia ACLs

- ACLs no son un estándar entre las versiones de Unix.
- Es posible que no funcionen en un ambiente de sistema de archivos en red.
- Sun planea soportar ACLs a través de extensiones en NFS3, más que implementar ACLs en su implementación.
- Hay que asegurarse que cualquier sistema que se exporte esta protegido adecuadamente por los permisos UNIX por default y las condiciones del propietario

Lámina 130


Dr. Roberto Gómez C.



Respaldos (backups)

- Es una copia de los datos escrita en cinta u otro medio de almacenamiento duradero.
- De manera rutinaria se recuerda a los usuarios de computadoras que respalden su trabajo con frecuencia.
- Los administradores de sitios pueden tener la responsabilidad de respaldar docenas o incluso de cientos de máquinas
- Comandos en Unix:
 - comandos de compresión
 - comando tar


Lámina 131 Dr. Roberto Gómez C.



Comandos comprimir

Formato archivo	Comando para comprimir/decomprimir	Comentarios
.Z	zcat / uncompress	Lempel -Ziv code
.z	pcat / unpack	Huffman Code
.zip	zip / unzip	Lempel-Ziv code
.gz	gzip/gunzip	versión GNU de zip
.rar	rar / unrar	unrar e -r archivo.rar unrar x -r
.bzip	bzip / bunzip2	bzip2 archivo unzip2 archivo.bz2


Lámina 132 Dr. Roberto Gómez C.



Ejemplo comandos compresión

```
toto@cachafas:2>ls -l
total 1080
-rw-r--r-- 1 rogozmez 394751 Jan 26 2001 arch1.pdf
-rw-r--r-- 1 rogozmez 372670 Jan 26 2001 arch2.pdf
-rw-r--r-- 1 rogozmez 300325 Jan 26 2001 arch3.pdf
toto@cachafas:3>gzip arch1.pdf
toto@cachafas:4>ls -l
total 1064
-rw-r--r-- 1 rogozmez 382986 Jan 26 2001 arch1.pdf.gz
-rw-r--r-- 1 rogozmez 372670 Jan 26 2001 arch2.pdf
-rw-r--r-- 1 rogozmez 300325 Jan 26 2001 arch3.pdf
toto@cachafas:5>gunzip arch1.pdf.gz
toto@cachafas:6>ls -l
total 1080
-rw-r--r-- 1 rogozmez 394751 Jan 26 2001 arch1.pdf
-rw-r--r-- 1 rogozmez 372670 Jan 26 2001 arch2.pdf
-rw-r--r-- 1 rogozmez 300325 Jan 26 2001 arch3.pdf
toto@cachafas:7>
```

Lámina 133 Dr. Roberto Gómez C.




Comando tar

- Crea archivos de cinta, añade o extrae archivos.
- El comando archiva o extrae varios archivos en un solo archivo con extensión .tar
- Sintaxis:

```
tar c [bBefPhiloPvwX [ 0-7]] [block ] [tarfile ][exclude-file ]
[ -l include-file ] [-C directory file | file ]...
tar r [bBefPhilvw [ 0-7]] [block ] [ -l include-file ] [-C directory file | file ]...
tar u [bBefPhilvw [ 0-7]] [block ] [tarfile ] file ...
tar t [bBefPhilvX [ 0-7]] [block ] [tarfile ][exclude-file ] [ -l include-file | file]...
tar x [BefPhilmopvwX [ 0-7]] [tarfile ][exclude-file ] [ -l include-file ] [ file ]...
```

Lámina 134 Dr. Roberto Gómez C.



Opciones comando tar

- c: crear archivo
- r: reemplazar
- u: actualizar. Los archivos son añadidos al archivo tar si no existen
- x: extraer/ restablecer
- v: verbose, da información de lo que el comando tar está haciendo

Lámina 135 Dr. Roberto Gómez C.




Ejemplo comando tar

```

toto@cachafas:7> ls -l
-rw-r-r--    1    al444561    96    Jul  1 17:08 arch1
-rw-r-r--    1    al444561   456    May  1 18:09 arch2
-rw-r-r--    1    al444561    56    May  5  2:69 arch3
toto@cachafas:8> tar -cfv nuevo arch1 arch2
toto@cachafas:9> ls -l
-rw-r-r--    1    al444561    96    Jul  1 17:08 arch1
-rw-r-r--    1    al444561   456    May  1 18:09 arch2
-rw-r-r--    1    al444561    56    May  5  2:69 arch3
-rw-r-r--    1    al444561   680    Jul  1 17:10 nuevo.tar
toto@cachafas:10> rm arch1
toto@cachafas:11> rm arch2
toto@cachafas:12>

```


Lámina 136 Dr. Roberto Gómez C.



Ejemplo comando tar (cont)

```
toto@cachafas:12> ls -l
-rw-r-r--      1    al444561  56  May 5  2:69 arch3
-rw-r-r--      1    al444561 680  Jul  1 17:10 nuevo.tar
toto@cachafas:13> tar -xvf nuevo
toto@cachafas:14> ls -l
-rw-r-r--      1    al444561  96  Jul  1 17:08 arch1
-rw-r-r--      1    al444561 456  May  1 18:09 arch2
-rw-r-r--      1    al444561  56  May  5  2:69 arch3
-rw-r-r--      1    al444561 680  Jul  1 17:10 nuevo.tar
toto@cachafas:15>
```


Lámina 137 Dr. Roberto Gómez C.



Compresión y empaquetamiento

- Comando zip comprime y empaqueta,
 - gzip ó bzip2 sólo comprimen archivos ,no directorios
- Para comprimir y empaquetar en un archivo al mismo tiempo hay que combinar el tar y el gzip o el bzip2
- Archivos .tar.gz (tgz)
 - Empaquetar y comprimir: tar -zvcf archivo.tgz directorio
 - Desempaquetar y descomprimir: tar -zvxf archivo.tgz
 - Ver contenido: tar -zvtf archivo.tgz
- Archivos tar.bz2 (tbz)
 - Empaquetar y comprimir: tar -jvcf archivo.tbz directorio
 - Desempaquetar y descomprimir: tar -jvxf archivo.tbz
 - Ver contenido: tar -jvtf archivo.tbz


Lámina 138 Dr. Roberto Gómez C.



El comando dump

- Construye una lista de los archivos que han sido modificados desde el último dump.
 - Posibilidad de acceder a múltiples cintas
 - Archivos de cualquier tipo (aún dispositivos) pueden ser respaldados y restablecidos
 - permisos, propiedades y estampillas tiempo son conservados
 - respaldos pueden hacerse de forma incremental
- No es conveniente usar dump en un sistema de archivo montado con actividad.
 - la forma `_canónica_` de usar dump es respaldar sistemas de archivo `_desmontados_`; si no se puede desmontar, entonces hacerlo a una hora en la cual no haya movimiento.

Lámina 139 Dr. Roberto Gómez C.




Ejemplo comando dump

- Realizar respaldo de una partición `/dev/sda1` en `/dev/rmt0`

```
# dump 0sfu 3600 /dev/rmt0 /dev/sda1
# dump 0sfu mis02: /dev/rmt0 /dev/sda1
```
- Segunda orden permite copia de seguridad de un disco sobre un dispositivo remoto
 - por ejemplo situado aquí sobre la máquina "mis02".
- Algunas opciones de dump:
 - **0 a 9** : nivel de copia de seguridad. 0 corresponde a una copia de seguridad completa, mientras que los otros niveles n corresponden a la copia de seguridad de archivos que fueron modificados desde la enésima copia de seguridad;
 - **s** : tamaño de la cinta en pies;
 - **f** : archivo. Puede estar compuesto de máquina:archivo;
 - **u** : escritura de la fecha y del nivel de copia de seguridad en el archivo `/etc/dumpdate`

Lámina 140 Dr. Roberto Gómez C.




Comando restore

- Para restaurar la cinta es necesario:

```
# restore -if /dev/rmt0  
# restore -if mis02:/dev/rmt0
```
- Para restaurar completamente una cinta:

```
# restore -if /dev/rmt0  
# restore -if mis02:/dev/rmt0
```


Lámina 141 Dr. Roberto Gómez C.



Los scripts

- Archivos que contienen comandos a ser ejecutados por el shell.
- Puede ser cualquier comando que pueda teclearse a partir del prompt:
 - comando que invoque una utilidad Unix, (vi, netscape, etc)
 - un programa compilado
 - otro script
- Aparte de estos comandos existe un grupo de comandos, (los *comandos de control de flujo*), que fueron diseñados para ser usados en scripts.

Lámina 142 Dr. Roberto Gómez C.



Ejemplo scripts

```

toto@cachafas:1>cat quienesta
date
echo Usuarios actualmente conectados
who
toto@cachafas:2> quienesta
quienesta: execute permission denied
toto@cachafas:3> ls -lg quienesta
-rw-r--r-- 1 toto pubs 42 Jun 17 10:55 quienesta
toto@cachafas:4> chmod +x quienesta
-rwxr--r-- 1 toto pubs 42 Jun 17 10:55 quienesta
toto@cachafas:5> quienesta
Fri Jun 17 10:59:40 PDT 1994
Usuarios actualmente conectados
toto      console Jun 17 08:26
cachafas  tty02   Jun 17 10:04
dongato   tty06   Jun 17 08:51
toto@cachafas:6>


```

```

toto@cachafas:6> cat prueba
echo " palabra 1 : \c"
read word1
echo " palabra 2 : \c"
read word2
if test "$word1" = "$word2"
then
    echo Concuerdan
fi
echo Fin del programa
toto@cachafas:7>

```

Lámina 143
Dr. Roberto Gómez C.



Principales comandos scripts

- Entrada salida
 - echo
 - read
- Variables predefinidas

Variable	Significado
\$?	Valor de salida del último comando, 0 si todo salió bien
\$0	nombre del script
\$1 a \$9	argumentos que se pasaron al script
\$#	numero argumentos pasados al script
\$*	lista de argumentos a partir de \$1
\$\$	numero pid del proceso actual
\$_	número pid del proceso hijo

Lámina 144
recordar que para desplegar su valor es necesario usar el comando *echo*
Dr. Roberto Gómez C.



Ejemplo de uso variables parámetros

```

$ more s1
echo Nombre del script: $0
echo Numero argumentos: $#
echo Lista de argumentos: $*
echo pid del proceso actual: $$
echo pid del proceso hijo: $!
$ ./s1 uno dos tres
Nombre del script: ./s1
Numero de argumentos: 3
Lista de argumentos: uno dos tres
pid del proceso actual: 3818
pid del proceso hijo:
$

```

Lámina 145

Dr. Roberto Gómez C.




Comandos de control

ciclos	<pre> cont=5 while [\$cont -gt 0] do echo vuelta \$cont cont=\$(expr \$cont - 1) done for cont in \$(seq 1 3) do echo vuelta \$cont done </pre>	case	<pre> echo -n "Dame un valor numérico: "; read opt case \$opt in 1) echo porque escogiste UNO? ;; 5) echo por el **** teee... ;; *) echo \$opt no me vale :(;; esac </pre>
	if-then-elif-else-fi <pre> if [condicion] then #procesos a realizar elif [condicion] then #procesos alternativos else #mas procesos alternativos fi </pre>		

Lámina 146

Dr. Roberto Gómez C.



Ejemplo uso variables control

```
$ more s1
for i in $*
do
  echo parametro $i
  if [ $i == "uno" ]
  then
    echo se introdujo parametro uno
    echo creando archivo uno
    echo esto es una prueba > uno
    ls -l uno
    echo contenido archivo uno
    more uno
    echo borrando archivo uno
    rm uno
    echo ls -l uno
    ls -l uno
  fi
done
$
```

```
$ s1 1328 83 tres uno k
parametro 1328
parametro 83
parametro tres
parametro uno
se introdujo parametro uno
creando archivo uno
-rw-rw-r-- 1 rogoomez rogoomez 19 Mar 3 14:10 uno
desplegando contenido archivo uno
esto es una prueba
borrando archivo uno
ls -l uno
parametro k
$
```

Lámina 147 Dr. Roberto Gómez C.