Prácticas de Fundamentos del Software

Módulo I. Órdenes UNIX y Shell Bash

Sesión Nº2: Órdenes Básicas de UNIX/Linux

3-Oct-2011

1 Introducción

Esta parte se dedica al manejo del Shell de forma interactiva e introduce un conjunto básico de órdenes para comenzar a trabajar con el sistema operativo.

2 Objetivos principales

- Manejo de las órdenes básicas de una shell.
- Utilización del manual de uso en línea.
- Manipulación básica de archivos y directorios.

Se presentarán los metacaracteres que nos ayudan a escribir muchas de las órdenes y daremos algunos consejos para su uso. En esta sesión veremos los metacaracteres de nombres de archivos.

Se verán las siguientes órdenes:

Órdenes Linux			
`man/info/help`	`who`	`rm`	`more`
`touch`	`file`	`mkdir/rmdir`	`head/tail`
`ls`	`cp`	`cd`	`sort`
`pwd`	`mv`	`cat`	`clear`

Tabla 1. Órdenes de la sesión.

3 Los intérpretes de órdenes

Un intérprete de órdenes, o shell en la terminología UNIX, está construido como un programa normal de usuario. Esto tiene la ventaja de que podemos cambiar de intérprete de órdenes según nuestras necesidades o preferencias. Existen diferentes shells: Bourne Again Shell (bash), TC Shell (tcsh) y Z Shell (zch). Estos shells no son exclusivos de Linux ya que se distribuyen libremente y pueden compilarse en cualquier sistema Linux. Podemos ver los shell de los que dispone nuestro sistema mirando en el archivo `/etc/shells`.

El shell, además de ejecutar las órdenes de Linux, tiene sus propias órdenes y variables, lo que lo convierte en un lenguaje de programación. La ventaja que presenta frente a otros lenguajes es su alta productividad -una tarea escrita en el lenguaje del shell suele tener menos código que si está escrita en un lenguaje como C-.

3.1 Orden man

La orden **man** es el paginador del manual del sistema. Las páginas usadas como argumentos al ejecutar man suelen ser normalmente nombres de programas, útiles o funciones. La página de manual asociada con cada uno de esos argumentos es buscada y presentada. Si la llamada da también la sección, man buscará sólo en dicha sección del manual. Normalmente, la búsqueda se lleva a cabo en todas las secciones de manual disponibles según un orden predeterminado, y sólo se presenta la primera página encontrada, incluso si esa página se encuentra en varias secciones. Para más ayuda escribir en la consola:

```
man man
```

man utiliza las siguientes teclas para buscar texto o desplazarse por el mismo:

- Control + F: avanzar página.
- Control + B: retrasar página.
- Control + P: ir arriba un espacio.
- Control + N: ir abajo un espacio.
- / texto: buscar el texto que se indica como argumento (hacia delante).
- ? texto: buscar el texto que se indica como argumento (hacia atrás).
- n: siguiente elemento en la búsqueda.
- N: elemento previo en la búsqueda.
- v: lanza (si es posible) el editor por defecto para editar el fichero que estamos viendo.
- q: sale del man.

Nota: Usa la orden man cuando no sepas las opciones correctas de una orden que necesitas para la realización de un ejercicio.

Como recomendación general, indicar que es conveniente usar el shell cuando necesitemos hacer algo con muchos archivos, o hacer la misma tarea de forma repetida. No lo deberías usar cuando la tarea sea compleja, requiera gran eficiencia, necesite de un entorno hardware diferente o requiera diferentes herramientas software.

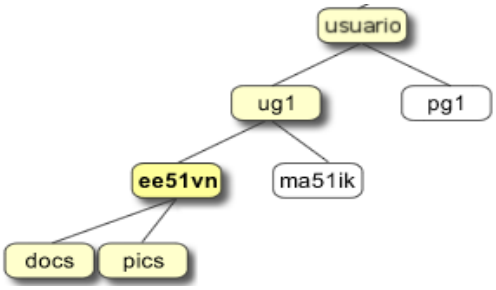
3.1 Órdenes Linux relacionadas con archivos y directorios

La tabla siguiente recoge las órdenes más frecuentes relacionadas con archivos y directorios (carpetas):

Órdenes	**Descripción**
`ls [directorio]`	Lista los contenidos de un directorio
`cd [directorio]`	Orden que nos cambia de directorio de trabajo. Las abreviaciones `.` y `..` se pueden utilizar como referencia de los directorios actual y padre, respectivamente.
`pwd`	Imprime el camino absoluto del directorio actual
`mkdir directorio`	Crea un directorio a partir del nombre dado como argumento
`rmdir directorio`	Borra un directorio existente (si está vacío)
`cat [archivo(s)]`	Orden multipropósito: muestra el contenido de un archivo o varios, concatena archivos, copia un archivo, crea un archivo de texto o muestra los caracteres invisibles de control.

cp archivo1 archivo2	Copia el archivo1 en el archivo2. Si archivo2 no existe, se crea.
mv fuente destino	Renombra archivos (el archivo fuente puede ser archivo o directorio, al igual que el destino) y puede mover de lugar un archivo o directorio
more archivo(s)	Visualiza un archivo fraccionándolo una pantalla cada vez (existen otros paginadores como `page`, `pg`, etc.)
rm directorio_archivos	Borra archivos y directorios con contenido
touch archivo(s)	Si existen los archivos dados como argumentos se modifican su fecha y hora. En caso contrario, se crean con la fecha actual del sistema.
file archivo(s)	Muestra el tipo de archivo dado como argumento
clear	Borra el contenido del terminal actual
tail [archivo(s)]	Muestra la parte final del contenido de un archivo dado como argumento
head [archivo(s)]	Muestra la parte inicial del contenido de un archivo dado como argumento
sort [archivo(s)]	Ordena, según un criterio elegido, el contenido de los archivos dados como argumentos

Ejercicio 1. Crea el siguiente árbol de directorios a partir de tu directorio de usuario (donde usuario representa tu nombre de usuario o login en el sistema operativo). Crea también los ficheros vacíos **`pg1`** y **`ma51ik`** en los directorios que se indican en la ilustración:



Ejercicio 2. Mueve el directorio **`pics`** dentro del directorio **`ug1`**. Renombra finalmente dicho directorio a **`imagenes`**. Copia el directorio **`docs`** dentro de **`ug1`**.

La orden **`ls`** (list sources) muestra los archivos contenidos en el directorio que se especifica (si no se especifica nada, tomará como directorio el directorio actual).

`ls [-option]` … **`[FILE]...`**

Algunas de las opciones más corrientes de esta orden son:

`-a` lista los archivos del directorio actual, incluidos aquellos cuyo nombre comienza con un punto, ".".
`-C` lista en formato multicolumna.
`-l` formato largo (ver descripción a continuación).
`-r` lista en orden inverso.
`-R` lista subdirectorios recursivamente, además del directorio actual.
`-t` lista de acuerdo con la fecha de modificación de los archivos.
Por ejemplo, con la opción **`-l`** veremos los archivos del directorio en formato largo, que da información detallada de los objetos que hay dentro del directorio. Para cada entrada del directorio, se imprime una línea con la siguiente información.

```
ls -l
-rw-r--r--  1 x00000 alumnos   23410 Mar 15 2009 programa.c
drwxr-xr-x  2 x00000 alumnos    1024 Aug 11 09:11 practfs
-rwxr-xr-x  1 x00000 alumnos   20250 Mar 15 2009 a.out
```

El significado de la información se irá viendo a lo largo de las sesiones. De momento, comentar dos de los campos informativos:

- **Tipo de objeto:** Un carácter indica el tipo de objeto que representa esa entrada: un **–** (guión) indica que es un archivo plano o regular; una **d** que es un directorio; la **c** indica que es un archivo de dispositivo orientado a carácter; la **b**, dispositivo orientado a bloques; la **l**, archivo de tipo enlace simbólico.
- **Bits de protección:** Parte de la protección en Linux descansa en la protección de los archivos. Cada archivo tiene asociados 12 *bits de protección* que indican qué operaciones podemos realizar sobre él y quien puede hacerlas. La orden **ls** muestra sólo 9 bits a través de una cadena de la forma genérica `rwxrwxrwx` donde cada letra representa un bit (un permiso). La pertenencia a un grupo la establece el administrador del sistema cuando lo crea (en nuestro sistema, todos los alumnos pertenecen al mismo grupo).

Los permisos se indican de la siguiente forma:

Permiso	Archivos	Directorios
r	Lectura (Read)	Se puede listar su contenido
w	Escritura (Write)	Podemos modificarlo
x	Ejecución (eXecute)	Podemos acceder a él
-	No hay permiso	

Existen tres grupos de permisos

rwx	rwx	rwx
Propietario del archivo (owner)	Grupo de usuarios (group)	Resto de usuarios (others)

Algunos ejemplos de las otras órdenes antes citadas:

- Cambiarse al directorio home para trabajar en él.

```
cd
```

- Copiar los archivos del directorio `x3333`, que está en el mismo nivel que en el que estamos situados, en el directorio `midir`.

```
cp ../x3333/ejemplo ../x3333/mensaje midir
```

- Asigna el nuevo nombre `megustamas` a un archivo existente, denominado `viejonombre`:

```
mv viejonombre megustamas
```

- Creamos un directorio para prácticas de fundamentos del software y borramos un directorio `temporal` que no nos sirve:

```
mkdir FS
rmdir temporal
```

- Ver el tipo de contenido de uno o varios archivos para, por ejemplo, visualizar el contenido de aquellos que son texto:

```
file practica1 programa.c a.out

Practica1: ASCII text
Programa.c: ISO-8859 c PROGRAM TEXT
a.out:      ELF 32-bit LSB executable, Intel 80386, version 1 …

cat practica1 programa.c
```

En ocasiones, podemos por error visualizar un archivo con la orden **`cat`**. Si el archivo contiene caracteres no imprimibles, por ejemplo, un archivo ejecutable, la visualización del mismo suele dejar el terminal con caracteres extraños. Podemos restablecer el estado normal del terminal con la orden siguiente:

```
setterm -r
```

Ejercicio 3. Liste los archivos que estén en su directorio actual y fíjese en alguno que no disponga de la fecha actualizada, es decir, el día de hoy. Ejecute la orden `touch` sobre dicho archivo y observe qué sucede sobre la fecha del citado archivo cuando se vuelva a listar.

Ejercicio 4. La organización del espacio en directorios es fundamental para poder localizar fácilmente aquello que estemos buscando. En ese sentido, realice las siguientes acciones dentro de su directorio **`home`** (es el directorio por defecto sobre el que trabajamos al entrar en el sistema):

a) Obtener en nombre de camino absoluto (pathname absoluto) de tu directorio home. ¿Es el mismo que el de tu compañero/a?
b) Crear un directorio para cada asignatura en la que se van a realizar prácticas de laboratorio y, dentro de cada directorio, nuevos directorios para cada una de las prácticas realizadas hasta el momento.
c) Dentro del directorio de la asignatura fundamentos del software (llamado **FS**) y dentro del directorio creado para esta sesión, copiar los archivos **`host`** y **`passwd`** que se encuentran dentro del directorio **`/etc`**.
d) Mostrar el contenido de cada uno de los archivos.

4 Metacaracteres de archivo

Cuando se pretende aplicar determinadas acciones a un conjunto de archivos y/o directorios cuyos nombres contienen secuencias de caracteres comunes (por ejemplo, `tema1.pdf`, `tema2.pdf` y `tema3.pdf`), es muy útil poder nombrarlos a través de una expresión que los pudiera identificar a todos (patrón de igualación). El patrón nos permite pasarle a la orden (sustitución de nombres) que vayamos a utilizar una única expresión para poder manipular todo el conjunto. Por ejemplo, para listar los tres archivos anteriores podemos utilizar la orden `ls tema?.pdf`

Metacarácter	Descripción de la función
?	Iguala cualquier carácter simple
*****	Iguala cualquier secuencia de cero o más caracteres
[ ]	Designan un carácter o rango de caracteres que, como una clase, son igualados por un carácter simple. Para indicar un rango, mostramos el primer y último carácter separados por un guión "-".
{ }	Sustituyen conjuntos de palabras separadas por comas que comparten partes comunes.
~	(se escribe pulsando AltGr 4) Se usa para abreviar el camino absoluto (path) del directorio home.

Como ejemplo de uso de los metacaracteres de archivos pueden ser los siguientes:

- Igualando un carácter simple con **?**

```
ls -l Sesion.1 Sesion.2 Sesion.3
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.1
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.2
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.3
```

```
ls -l Sesion.?
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.1
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.2
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.3
```

- Igualando cero o más caracteres con *****

```
ls -l Sesion.*
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.1
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.2
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.3
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 Sesion.apendices *.
```

- Igualando cero o más caracteres con **[]**

```
ls -l *.[ch]
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 hebras.c
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 hebras.h
-rw-r--r-- 1 jrevelle prcicyt 0 oct  7 11:06 pipes.c
```

- Abreviando el directorio home con **~**

```
pwd
/tmp
```

```
cd ~/fs
```

```
pwd
/home/x01010101/fs
```

Ejercicio 5. Desplacémonos hasta el directorio **`/bin`**, genere los siguientes listados de archivos (siempre de la forma más compacta y utilizando los metacaracteres apropiados):

a) Todos los archivos que contengan sólo cuatro caracteres en su nombre.
b) Todos los archivos que comiencen por los caracteres **d**, **f**.
c) Todos los archivos que comiencen por las parejas de caracteres **sa**, **se**, **ad**.
d) Todos los archivos que comiencen por **t** y acaben en **r**.

Ejercicio 6. Liste todos los archivos que comiencen por **tem** y terminen por **.gz** o **.zip** :

a) de tu directorio home
b) del directorio actual
c) ¿hay alguna diferencia en el resultado de su ejecución? Explique el por qué si o el por qué no.

Ejercicio 7. Muestre del contenido de un archivo regular que contenga texto:

a) las 10 primeras líneas.
b) las 5 últimas líneas.

Ejercicio 8. Ordene el contenido de un archivo de texto según diversos criterios de ordenación.

Ejercicio 9. ¿Cómo puedes ver el contenido de todos los archivos del directorio actual que terminen en **.txt** o **.c**?