Administración Linux

Sistema de ficheros

ls: listar archivos

-**l** Mostrar información adicional de cada archivo

-a Mostrar todos los archivos, incluidos los ocultos

-r Listar en orden inverso

-t Listar en orden cronológico ascendente

-h Mostrar tamaños de ficheros en KB o MB en vez de bytes

-larth muestra toda la información de los archivos de un directorio ordenados de manera cronológica y con tamaños en KB o MB

pwd Mostrar directorio actual.

cd Cambiar directorio.

mkdir Crear directorio.

cp Copiar archivos

mv Mover archivos (o renombrar) mv <ruta-fichero-origen> <destino>

rm Borrar archivos (o directorios con el parámetro -r)

ln Crear enlaces a archivos (-s para simbólico) ln -s <ruta-fichero-origen> <ruta-enlace>

whereis Ubicación de un comando

find Buscar archivos find <carpeta-base> -name <nombre-archivo>

cat/more/less Mostrar contenidos de un archivo

wc Contar palabras (o líneas con el parámetro -l)

head/tail Mostrar N primeras/últimas líneas (utilizando -n)

grep Buscar patrón de texto en una archivo grep casa miArchivo.txt

cut Muestra secciones concretas de un archivo cut -d " " -f2 a.txt

tar Comprimir/descomprimir archivos/carpetas

Comprimir tar cfvz carpeta.tgz miCarpeta

Descomprimir: tar xfvz carpeta.tgz

sort Ordena las líneas de un archivo alfabética

whoami Muestra el nombre del usuario actual

who Muestra los usuarios conectados al sistema

w Equivalente a who, algo más de información

passwd Cambiar contraseña del usuario actual

write Escribir un mensaje otro usuario

useradd Crear usuario en el sistema

adduser Script “asistente” para crear un usuario

chmod Modificar permisos de un fichero o carpeta

chown/chgrp Modificar UID/GID de un fichero

top Muestra el estado del sistema (carga de CPU, memoria,…)

ps Muestra información sobre los procesos activos

kill Enviar una señal a un proceso kill -9 miProceso fuerza su terminación

pstree Muestra el árbol de procesos

sed modificar un fichero dado como entrada.sed <opciones> <instrucciones> <fichero>

-i El fichero de entrada es sobrescrito, en lugar de usar stdout

sed -i ‘s/casa/coche/g’ a.txt Reemplazar “casa” por “coche” en a.txt

sed -i ‘/cadena/d’ archivo Eliminar toda ocurrencia de cadena en a.txt

sed ‘2,3 p’ * Mostrar líneas 2 y 3 de todos los ficheros

Awk procesado de texto awk: patrón { acción }

awk '{print $1,$4}’ data.txt Muestra la 1ª y 4ª columna de data.txt

awk '{print $1,$NF}’ data.txt Muestra la 1ª y última columna de data.txt

awk 'END { print NR }’ data.txt Cuenta el nº de líneas de data.tx

Shell scripting

Si el script no tiene permisos: bash miScript

modificar los permisos del script y ejecutarlo como un binario: chmod +x miScript; ./miScript

read Para leer del teclado

echo/printf Para escribir por pantalla (stdout)

Los parámetros que reciba se convierten en variables:

$1 .. 9 Los parámetros, el número indica su posición

$0 Contiene el nombre del script

$# Número de parámetros recibidos

$? El PID del proceso que está ejecutando el script

$* Contiene todos los parámetros ($1 + $2 + …)

```
#!/bin/bash
if [ `whoami` = "unai" ]; then
echo "El usuario actual es unai"
else
echo "El usuario actual no es unai"
fi
```

Strings	Numéricos	Verdadero si
x = y	x -eq y	x es igual a y
x != y	x -ne y	x es diferente a y
x < y	x -lt y *	x es menor que y
	*: -gt, -le, -ge	equivalen: >, <=, >=

Operador	Verdadero si
-d carpeta	Carpeta existe
-e fichero	Fichero existe
-r fichero -w fichero	Usuario tiene permiso de lectura / escritura en el fichero
-s fichero	Fichero existe y no está vacío

- Sentencia **case**

```
usu=$1
case $usu in
  "unai")  echo "El usuario es Unai";;
  "mikel") echo "El usuario es Mikel";;
  "jon")   echo "El usuario es Jon";;
esac
```

FOR

- Archivos:

```
for archivo in `ls`; do
  echo $archivo;
done
```

- Número o elementos concretos:

```
for i in 1 3 5 9 11; do
  echo $i;
done
```

- Secuencias de números:

```
for i in {1..8};do
  echo $i;
done
```

- Expresiones regulares
  - Utilizadas para buscar texto que corresponda a un patrón
  - Un patrón se construye con literales y caracteres especiales
  - Ejemplos de patrones:
    - "casa"
      - Encontraría "casa"
    - "c([a-z])sa"
      - "casa", "cbsa", "ccsa",...
    - "c([a-z]+)sa"
      - "casa", "caasa", "cxyzsa"
    - "c(\d*)sa"
      - "csa", "c101sa", ...

Símbolo	Significado
.	Cualquier carácter
[]	Caracteres definidos entre []
\d	Cualquier dígito
*	Una, ninguna o más del elemento precedido
+	Uno o más del elemento precedido

- Funciones
  - Se pueden crear funciones para ordenar el código
  - Se le pueden pasar parámetros
    - Se recogen en la función como **$1**, **$2**, ... en base al orden
  - Para devolver un valor de la función, utilizar **echo** y no return
    - return se interpreta como el comando de finalizar el programa
  - Ejemplo:

```
function suma()
{
  OP1=$1
  OP2=$2
  echo $(($OP1+$OP2))
}

A=33
B=44
res=$(suma $A $B)
echo "La suma de $A y $B es $res"
```