

Fundamentos: Sistemas y arquitecturas
Práctica 1
Sistema operativo Linux

José Manuel Bustos Muñoz

Ejercicio 1:

Haciendo uso de la utilidad `man` encontrar y documentar las siguientes opciones para los comandos:

- `ls`:
 - opción `-t`
 - opción `-c`
- `cat`:
 - opción `-n`
 - opción `-T`

En la consola utilizamos el comando “`man`” para ver la ayuda de “`ls`” → *man ls*. Una vez en la ayuda bajamos para observar las opciones `-t` y `-c` indicadas.

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
  fied.

  Mandatory arguments to long options are mandatory for short options
  too.
```

ls -t: Ordena los archivos por fecha de modificación. Los más nuevos primero.

```
-t      sort by modification time, newest first
```

ls -c: muestra los elementos en orden según la fecha de última modificación.

```
-c      with -lt: sort by, and show, ctime (time of last modification of
        file status information) with -l: show ctime and sort by name
        otherwise: sort by ctime, newest first
```

Si probamos en un directorio con varios elementos a listar con “`ls`” y con “`ls -c`” por ejemplo, podemos ver como se muestran los elementos en diferente orden.

```
bigdata@bigdata:~$ cd Desktop
bigdata@bigdata:~/Desktop$ ls
firefox.desktop  lxterminal.desktop  wireshark.desktop
leafpad.desktop  pcmanfm.desktop
bigdata@bigdata:~/Desktop$ ls -c
wireshark.desktop  pcmanfm.desktop  lxterminal.desktop
firefox.desktop    leafpad.desktop
```

En la consola utilizamos el comando “**man**” para ver la ayuda de “**cat**” → *man cat*. Una vez en la ayuda bajamos para observar las opciones -n y -T indicadas.

```
CAT(1)                                User Commands                                CAT(1)

NAME
  cat - concatenate files and print on the standard output

SYNOPSIS
  cat [OPTION]... [FILE]...

DESCRIPTION
  Concatenate FILE(s), or standard input, to standard output.

  -A, --show-all
      equivalent to -vET

  -b, --number-nonblank
      number nonempty output lines, overrides -n

  -e
      equivalent to -vE

  -E, --show-ends
      display $ at end of each line
```

cat -n: Numera las líneas mostradas (Desde 1).

```
-n, --number
    number all output lines
```

cat -T: Para visualizar las tabulaciones.

```
-T, --show-tabs
    display TAB characters as ^I
```

Ejercicio 2:

Dado el fichero nombres.txt obtener un listado de los 5 nombres más repetidos junto con su número de apariciones en el fichero. Indicar el comando (o comandos usados).

1. Ordenamos el fichero de nombres para que las coincidencias estén seguidas, y con el uso de uniq eliminamos los duplicados y a cada elemento añadimos un contador del número de ocurrencias. Almacenamos el resultado en otro fichero.
"sort nombres.txt | uniq -c > fichero.txt"
2. Ordenamos dicho fichero de mayor a menor por el número de ocurrencias, lo almacenamos en otro fichero.
"sort -r fichero.txt > fich.txt"
3. Mostramos sólo los 5 primeros elementos del fichero final con head.
"head -5 fich.txt"

```
bigdata@bigdata:~/Desktop/material_practica$ ls
calculaFG.bash  factorial.bash  nombres.txt
bigdata@bigdata:~/Desktop/material_practica$ sort nombres.txt | uniq -c > fichero.txt
bigdata@bigdata:~/Desktop/material_practica$ sort -r fichero.txt > fich.txt
bigdata@bigdata:~/Desktop/material_practica$ head -5 fich.txt
  1742 Victoria
  1681 Maria
  1678 Alberto
  1645 Juan
  1631 Ana
bigdata@bigdata:~/Desktop/material_practica$
```

Ejercicio 3:

¿En qué directorio (o directorios) se encuentra el archivo firefox.desktop? Indicar el comando o comandos usados para encontrarlo.

El archivo se encuentra en el directorio **"Desktop"** con la ruta *"/home/bigdata/Desktop"*.

Para verlo se pueden usar los comandos:

"find -name firefox.desktop" para hacer la búsqueda desde el directorio actual,

o

"find / -name firefox.desktop" para hacer la búsqueda desde el directorio raíz.

```
bigdata@bigdata:~$ find -name firefox.desktop
./Desktop/firefox.desktop
bigdata@bigdata:~$ sudo find / -name firefox.desktop
/home/bigdata/Desktop/firefox.desktop
```

Ejercicio 4:

¿Qué comando debe usarse para obtener la fecha actual con el formato año-mes-día hora:minuto:segundo?

Para obtener la fecha se debe utilizar el comando `date`, que con distintas opciones podemos formatear para obtener la fecha como deseemos. En este caso sería con el siguiente comando:

`date +"%Y-%m-%d %H:%M:%S"`

```
bigdata@bigdata:~$ date +"%Y-%m-%d %H:%M:%S"
2017-10-03 22:28:04
bigdata@bigdata:~$
```

Ejercicio 5:

Crear un script shell que muestre:

- Modelo de procesador
- Cantidad de memoria libre (en GB)
- Cantidad de disco libre (en GB) para la partición /

El script “**ejercicio5.bash**” tiene los comandos utilizados para este ejercicio. Al ejecutarlo se obtiene la información:

```
bigdata@bigdata:~/Desktop$ ./ejercicio5.bash
La información sobre el procesador es la siguiente:
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 58
model name    : Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz
stepping      : 9
microcode     : 0x19
cpu MHz       : 2494.316
cache size    : 3072 KB
physical id   : 0
siblings      : 1
core id       : 0
cpu cores     : 1
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level   : 13
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
               nat nse36 clflush rtmx fxsr sse sse2 syscall nx rdtscp lm constant tsc rep good r
```

```

cpuid level      : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx rdtscp lm constant_tsc rep_good
opl xtopology nonstop_tsc pn1 pclmulqdq monitor ssse3 cx16 sse4_1 sse4_2 x2apic
popcnt aes xsave avx rdrand hypervisor lahf_lm
bugs            :
bogomips        : 4988.63
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:

La cantidad de memoria libre en GB:

```

	total	usado	libre	compart.	buffers	almac.
Mem:	1	0	1	0	0	0
-/+ buffers/cache:		0	1			
Intercambio:		0	0	0		

```

La cantidad de disco libre en GB para la partición /:
S.ficheros      Tamaño Usados  Disp Uso% Montado en
/dev/sda1       19G   2,4G   16G  14% /

```

Ejercicio 6:

Crear un script shell que reciba como argumento una cadena de texto con una extensión (por ejemplo, txt) y muestre el número de ficheros con dicha extensión en el directorio actual.

*El script “**ejercicio6.bash**” contiene la solución al ejercicio planteado. Al ejecutar el script se obtiene la siguiente información:*

```

bigdata@bigdata:~/Desktop$ ls
ejercicio5.bash  leafpad.desktop  material_practica.tar.gz
ejercicio6.bash  lxterminal.desktop  pcmanfm.desktop
firefox.desktop  material_practica  wireshark.desktop
bigdata@bigdata:~/Desktop$ ./ejercicio6.bash a.desktop
El número de archivos con esa extensión es: 5
bigdata@bigdata:~/Desktop$ ./ejercicio6.bash a.bash
El número de archivos con esa extensión es: 2
bigdata@bigdata:~/Desktop$

```

Ejercicio 7:

Obtener el nombre del usuario que ha lanzado el proceso `rsyslogd`. Indicar el comando o comandos usados.

Con el comando “`ps`” podemos ver los procesos del sistema. Con las opciones “`-aux`” sale el listado de todos los procesos con distintas información.

```
bigdata@bigdata:~$ ps -aux
USER      PID  %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0  0.2 33640  4260 ?        Ss   20:58   0:01 /sbin/init
root         2   0.0  0.0      0     0 ?        S    20:58   0:00 [kthreadd]
root         3   0.0  0.0      0     0 ?        S    20:58   0:00 [ksoftirqd/0]
root         5   0.0  0.0      0     0 ?        S<   20:58   0:00 [kworker/0:0H]
root         7   0.0  0.0      0     0 ?        S    20:58   0:00 [rcu_sched]
root         8   0.0  0.0      0     0 ?        S    20:58   0:00 [rcu_bh]
root         9   0.0  0.0      0     0 ?        R    20:58   0:00 [rcuos/0]
root        10   0.0  0.0      0     0 ?        S    20:58   0:00 [rcuob/0]
root        11   0.0  0.0      0     0 ?        S    20:58   0:00 [migration/0]
root        12   0.0  0.0      0     0 ?        S    20:58   0:00 [watchdog/0]
root        13   0.0  0.0      0     0 ?        S<   20:58   0:00 [khelper]
root        14   0.0  0.0      0     0 ?        S    20:58   0:00 [kdevtmpfs]
root        15   0.0  0.0      0     0 ?        S<   20:58   0:00 [netns]
root        16   0.0  0.0      0     0 ?        S<   20:58   0:00 [perf]
root        17   0.0  0.0      0     0 ?        S    20:58   0:00 [khungtaskd]
```

El usuario que ha lanzado el proceso “`rsyslogd`” sería: “***syslog***”, como puede verse en la imagen.

```
message+ 567 0.0 0.1 39848 3124 ?        Ss   20:58   0:00 dbus-daemon --s
root     583 0.0 0.0 15280  224 ?        S    20:58   0:00 upstart-file-br
syslog   587 0.0 0.1 255844 2892 ?        Ssl  20:58   0:00 rsyslogd
root     596 0.0 0.1 19300  2548 ?        Ss   20:58   0:00 /usr/sbin/bluet
avahi    608 0.0 0.1 32348 3044 ?        S    20:58   0:00 avahi-daemon: r
```

Ejercicio 8:

Dados los scripts `calculaFG.bash` y `factorial.bash`. Calcular el tiempo que tarda en ejecutar el script `calculaFG.bash`. Realizar 4 repeticiones y mostrar la media. ¿Qué comando o comandos ha usado?

Podemos utilizar el comando **“time”** para calcular los tiempos al ejecutar el script. El real es el tiempo transcurrido, podría sumarse los 4 tiempos obtenidos y hacer la media. **“time ./calculaFG.bash”**

```
bigdata@bigdata:~/Desktop/material_practica$ time ./calculaFG.bash
real    0m7.060s
user    0m0.272s
sys     0m0.488s
bigdata@bigdata:~/Desktop/material_practica$ time ./calculaFG.bash
real    0m6.952s
user    0m0.240s
sys     0m0.512s
bigdata@bigdata:~/Desktop/material_practica$ time ./calculaFG.bash
real    0m7.088s
user    0m0.244s
sys     0m0.524s
bigdata@bigdata:~/Desktop/material_practica$ time ./calculaFG.bash
real    0m7.072s
user    0m0.256s
sys     0m0.512s
bigdata@bigdata:~/Desktop/material_practica$
```

Con el siguiente comando puede recogerse sólo el tiempo real en lugar de los 3 anteriores.

‘sudo time -f “%e” ./calculaFG.bash’

```
bigdata@bigdata:~/Desktop/material_practica$ sudo time -f "%e" ./calculaFG.bash
6.19
bigdata@bigdata:~/Desktop/material_practica$
```

Realizamos un script **“ejercicio8.bash”** que ejecutándolo en el directorio del script al que queremos calcular el tiempo de ejecución ejecutamos 4 veces el comando y calculamos la media.

```
bigdata@bigdata:~/Desktop/material_practica$ ./ejercicio8.bash
Los tiempos de cada una de las 4 ejecuciones han sido: 6.15 6.35 7.44 6.13
La media ha sido: 6.51750000000000000000
bigdata@bigdata:~/Desktop/material_practica$
```


Ejercicio 9:

Modifique el script calculaFG.bash para que calcule los factoriales en paralelo. Calcular el tiempo consumido ahora por el script. ¿Observa alguna diferencia?.

Utilizamos **&** para ejecutar en paralelo los procesos y con **wait** esperamos a que terminen.

```
#!/bin/bash

./factorial.bash 1000 > /dev/null &
./factorial.bash 1001 > /dev/null &
./factorial.bash 1002 > /dev/null &
./factorial.bash 1003 > /dev/null &

wait %1 %2 %3 %4
```

Al ejecutar tarda algo más pero es prácticamente lo mismo, no hay gran diferencia.

```
bigdata@bigdata:~/Desktop/material_practica$ ./ejercicio8.bash
Los tiempos de cada una de las 4 ejecuciones han sido: 6.64 7.66 7.69 7.61
La media ha sido: 7.40000000000000000000000000000000
bigdata@bigdata:~/Desktop/material_practica$
```

Ejercicio 10:

¿Cómo posicionaría la ejecución del script anterior en el core número 1? Indicar el comando o comandos.

Con el uso del comando **taskset**, opción **-c** y se le indica el número o números del core/s donde queremos ejecutar los procesos, y le indicamos el script o proceso con la opción **-p** y pasándole el PID.

“taskset -cp 1 PID”

Ejercicio 11:

¿Cómo limitaría la cantidad máxima de memoria al lanzar el comando cat? Indicar el comando o comandos necesarios.

Tenemos el comando **ulimit** para limitar ciertos parámetros, como la cantidad máxima de memoria utilizable y también la virtual, con las opciones -m y -v.

“**ulimit -v 10**” así limitamos la memoria virtual a utilizar y afectaría al lanzar un cat.

Ejercicio 12:

¿Qué variable de entorno indica el usuario actual? ¿Qué valor tiene? Indicar cómo lo ha obtenido.

Para ver todas las variables de entorno se utiliza el comando “**printenv**”.

```
bigdata@bigdata:~$ printenv
XDG_VTNR=7
SSH_AGENT_PID=1352
XDG_SESSION_ID=c2
SAL_USE_VCLPLUGIN=gtk
CLUTTER_IM_MODULE=xim
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/bigdata
TERM=xterm
SHELL=/bin/bash
XDG_MENU_PREFIX=lxde-
GNOME_KEYRING_CONTROL=/run/user/1000/keyring-exyFk0
USER=bigdata
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd
```

De la lista obtenida la variable de entorno que indica el usuario actual es “**USER**”.

```
GNOME_KEYRING_CONTROL=/run/user/1000/keyring-exyFk0
USER=bigdata
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd
```

Con “**echo nombre_variable_entorno**” se visualiza el valor de dicha variable.

```
bigdata@bigdata:~$ echo $USER
bigdata
```

Ejercicio 13:

¿Qué línea debemos utilizar en cron para ejecutar el script del ejercicio 5 de forma semanal los viernes a las 14? ¿Cómo almacenaríamos la salida de dicho script de manera incremental en un archivo llamado stats.txt?

Para planificar la ejecución de procesos utilizamos el comando **"crontab"**. Con la opción **-e** se abre el fichero de cron con algún editor de texto y se pueden planificar procesos. En la línea a añadir al archivo para planificar la tarea se pasan los diferentes argumentos que indican hora, frecuencia, etc. Y por último el comando o script a ejecutar con su ruta absoluta. El usuario debe tener permisos para ejecutar el script.

"crontab -e"

"00 14 * * 5 * /home/bigdata/Desktop/ejercicio5.bash"

Enviamos la salida a un fichero stats.txt:

**"00 14 * * 5 * /home/bigdata/Desktop/ejercicio5.bash >>
/home/bigdata/Desktop/stats.txt"**

Ejercicio 14:

Comprimir en un archivo con extensión (.tar.bz2) el documento de respuesta a esta práctica y cualquier otro material derivado de esta práctica (scripts o documentos extras).

Para comprimir en un archivo con extensión ".tar.bz2" utilizamos el siguiente comando:

"tar -cvjf nombreArchivo.tar.bz2 lista_de_ficheros_a_comprimir"

**'tar -cvjf practica1josemanuelbustos.tar.bz2 ejercicio5.bash ejercicio6.bash
ejercicio8.bash "Practica 1 Sistemas y Arquitectura - Jose Manuel Bustos Munoz.pdf"'**

Como resultado obtenemos el archivo que será la entrega de la práctica:

"practica1josemanuelbustos.tar.bz2".