

WUOLAH



Nessy

www.wuolah.com/student/Nessy



3499

Ejercicios .pdf

Material Examen Módulo I



2º Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

MÓDULO I: Administración de Linux (Ejercicios)

Sesión 1: Herramientas de administración básicas

1.1 Repaso de scripts de bash

```
#!/bin/bash
cp kernel/*.gz /tmp
gunzip /tmp/*.gz
/tmp/kernel123-3.0.4 ubda=/tmp/Fedora14-x86-root_fs mem=1024m
```

1.2 Valores por omisión para nuevas cuentas

```
cat /etc/default/useradd (etc_default_useradd.txt)
cat /etc/login.defs (etc_login.defs.txt)
adduser Dani
cat /etc/passwd (etc_passwd.txt)
cat /etc/group (etc_group.txt)
ls -l /home (drwx----- 2 Dani Dani 4096 Sep 24 17:10 Dani)
```

1.3 Creación de usuarios

```
sudo useradd -d /home/dani -m -s /bin/bash dani
```

- -d: se utiliza /home/dani como valor para el directorio de inicio de sesión del usuario. Si no está creado NO lo crea.
- -m: crea el directorio home del usuario, si este no existe.
- -s: se selecciona el Shell de inicio de sesión del usuario.

```
gedit nuevos_usuarios &
otro:x:1002:1002:otro::/home/diferente:/bin/sh (o bash)
```

```
newusers nuevos_usuarios
```

1.4 Archivos /etc/passwd

```
Contenido: etc_passwd.txt
Propietario: root
Permisos: -rw-r--r--
(ls -l /etc/passwd)
```

1.5 Archivo /etc/shadow

El archivo solo tiene permiso de lectura para el root, por lo que, para poder visualizarlo, debe consultarse desde usuario root.

1.6 Creación de grupos

1.
`groupadd nombre_grupo`
`gpasswd -a usuario grupo`
2.
Root: uid=0(root)
gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
Usuario: uid=1000(laura) gid=1000(laura)
grupos=1000(laura),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)

1.7 Archivo del kernel de Linux

Find / -name Fedora

1.8 Organización del SA

**** En el directorio /sbin si es un programa fundamental y si no en su directorio (/root)

1.9 Información de los SAs

/etc/fstab: lista de discos y particiones disponibles. Se indica como montar cada dispositivo y qué configuración utilizar.

/etc/mtab: es un archivo de información del sistema. Lista todos los sistemas de ficheros montados, junto con sus opciones de inicialización.

1.10 Información de los SAs

1.11 Archivos de información para los SAs

- **/etc/fstab:** lista de discos y particiones disponibles. Se indica cómo montar cada dispositivo y qué configuración utilizar.
- **/etc/mtab:** lista todos los sistemas de ficheros montados actualmente junto con sus opciones de inicialización.
- **/proc/filesystems:** muestra una lista de los tipos de SA soportados actualmente por el kernel
- **/proc/mounts:** proporciona una lista de todos los montajes en uso por el sistema. La salida de datos que encontramos aquí se parece a /etc/mtab, solo que más actualizada.

Sesión 2: Herramientas de administración del SA

2.1 Creación de SA

```
mke2fs -t ext3 -L LABEL_ext3 /dev/loop0  
mke2fs -t ext4 -L LABEL_ext4 /dev/loop1
```

- **-t**: tipo de archivo que se va a asignar.
- **-L**: etiqueta que se le quiere establecer.

2.2 Personalización de los metadatos del SA

- a) ¿Cómo conseguir que siempre se ejecute **e2fsck** al arrancar el programa?
Poniendo el **-c max-mount-counts** como **0** o **-1**.
- b) ¿Cómo reservar un número de bloques del SA para un usuario?
Con la opción **-u**, especificando el usuario que podrá usar los bloques reservados, y con **-r**, especificando el número de bloques a reservar.

2.4 Montaje de SA

- a) El SA **LABEL_ext3** en **/mnt/SA_ext3** en modo solo lectura.

```
mkdir /mnt/SA_ext3  
mount -r -L LABEL_ext3 /mnt/SA_ext3
```
- b) El SA **LABEL_ext4** en **/mnt/LABEL_ext4** con sincronización de E/S de modificación de directorios.

```
mount -o dirsync -L LABEL_ext3 /mnt/LABEL_ext4
```

2.5 Automontaje de SA

- a)

```
LABEL=LABEL_ext3 /mnt/SA_ext3 ext3 ro 0 0
```
- b)

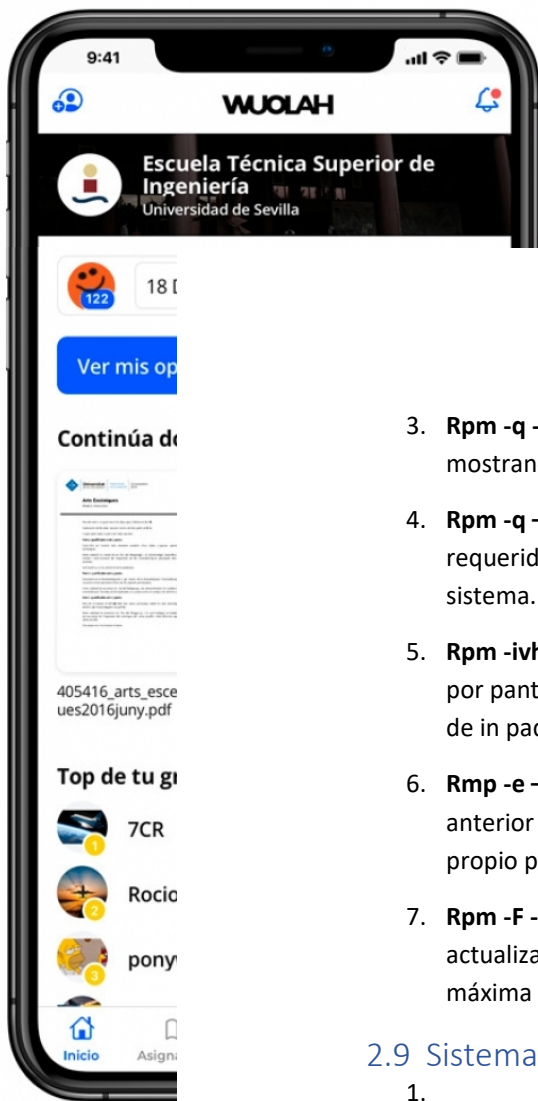
```
LABEL=LABEL_ext4 /mnt/LABEL_ext4 ext4 dirsnc 0 0
```

2.7 Trabajo con el gestor de paquetes YUM

- **yum list installed**: lista todos los paquetes instalados en el sistema.
- **Yum remove <paquete>**: elimina un paquete instalado.
- **Yum install <paquete>**: instala un paquete.

2.8 Trabajo con el gestor de paquetes rpm

1. **Rpm -qi <nombre-paquete-instalado>**: mostrar información sobre uno o más paquetes ya instalados.
2. **Rpm -qli <nombre-paquete-instalado>**: como 1.- pero mostrando solo los archivos de configuración que contiene.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



3. **Rpm -q -c -i <nombre-paquete-instalado> // o "-c"?:** como el anterior, pero mostrando solo los archivos de configuración que contiene el paquete
4. **Rpm -q --whatrequires <nombre-paquete-instalado>:** muestra los paquetes requeridos por un paquete determinado que se encuentra instalado en el sistema.
5. **Rpm -ivh </fénix/depar/lsi/so/paquetes/"nombre del paquete">:** muestra por pantalla la máxima información posible acerca del proceso de instalación de in paquete a partir del directorio punto de montaje.
6. **Rmp -e --nodeps -v <nombre-paquete>:** desinstala el paquete del punto anterior mostrando por pantalla la máxima información posible acerca del propio proceso de eliminación del paquete.
7. **Rpm -F -v </fénix/depar/lsi/so/paquetes/"nombre del paquete">:** paso 4 con actualización del paquete instalado a la versión más reciente. Muestra la máxima información por pantalla.

2.9 Sistema de cuotas para el SA tipo ext3

1.

```
LABEL=LABEL_ext3 /mnt/SA_ext3 ext3 usrquota 0 0
```
2.

```
mount -o remount /mnt/SA_ext3
```
3.

```
[apt-get install quota]
quotacheck -nm /mnt/SA_ext3
```
4.

```
quotaon -a
```
5.

```
edquota <nombre_usuario>
// Otra idea....
#!/bin/bash
for [ linea in `cat /etc/passwd` ]
do
usuario = `cut -d : -f1`
setquota -u $usuario
done
```
6.

```
edquota -t
```

2.10 Establecer límites sobre recursos de un SA

- - setquota -u <nombre_usuario> -b nombre 7 10 7 10
 - -u: especifica el usuario para el que se le van a editar
 - -b: lee la información desde la línea, con el siguiente formato:
name block-softlimit block-hardlimit inode-softlimit inode-hardlimit

Sesión 3: Monitorización del Sistema

3.1 Consulta de estadísticas del sistema

- a)
- uptime**
up 6:29, es decir, lleva 6 horas y media el sistema en marcha.
- b)
- uptime**
2 users, es decir, hay 2 usuarios trabajando.
- c)
- uptime**
load average: 0,31, 0,43, 0,44, es decir, en los últimos 15 minutos, la carga media del sistema es de 0,44.
(1 minuto, 5 minutos, 15 minutos).

3.2 Prioridad de los procesos

- a)
- ```
prueba_procesos <num_iteraciones>
#!/bin/bash
iteraciones=$1 # Coge el valor del segundo argumento
x=0 # Se inicializa una nueva variable y su valor
for ((COUNT=0; COUNT<iteraciones; COUNT++));
do
x=$(($x + 1))
done
printf "El valor de la variable es $x\n"
```
- b)
- ps -l** : muestra los procesos en ejecución con sus prioridades (columna **NI**)  
**renice <valor> <PID>** : cambia la prioridad de un proceso en ejecución. Si se quiere disminuir esta prioridad, debe hacerse en súper-usuario.

c)

Para obtener los tiempos, lanzaría los procesos con la orden **time ./prueba\_procesos 10000 &.**

### 3.3 Jerarquía e información de procesos

“?” en TTY quiere decir que ese proceso no está ligado a la terminal interfaz (terminal interface).

### 3.4 Estadística de recursos del sistema

a)

**mpstat** ; se ha gastado 0.0 (%irq)

b)

**mpstat** ; se ha gastado 0.03 (%soft)

c)

**top** ; 427500 libre (free) y 56300 ocupada (used)

### 3.5 Consulta de metadatos de archivo

**Ls -la** : listar todos los archivos del directorio en formato largo.

**b**: dispositivo de bloques.

**c**: dispositivo de caracteres.

### 3.6 Listado de metadatos de archivos: ls

- “Access time”: **ls -ltu** (ordenados por este campo).
- “ctime”: **ls -ltc** (ordenados por ese campo).

### 3.7 Metadatos del SA: df y du

1. **df -h /**

2. **df -i /**

3. **du -h /etc**

**du -h /var**

### 3.8 Creación de enlaces con la orden ln

**touch archivo.txt**

**touch target\_hardLink2.txt**

**ln -s archivo.txt softLink** (Crear un soft link)

**ln -P archivo.txt hardLink** (Crear un hard link)

**ln -P target\_hardLink2.txt hardLink2**



- El contador del archivo archivo.txt vale 2 porque solo cuenta los enlaces duros, no los simbólicos.

### 3.9 Trabajo con enlaces

- **Ls -lL** (muestra la información de los archivos a los que referencian los enlaces simbólicos)
- **Ls -l**

### 3.10 Creación de archivos especiales

- **sudo mknod bloques b 4 64**
- **sudo mknod caracteres c 5 20**
- **ls -li**  

|        |            |   |      |      |       |              |            |
|--------|------------|---|------|------|-------|--------------|------------|
| 274296 | brw-r--r-- | 1 | root | root | 4, 64 | oct 26 00:13 | bloques    |
| 275509 | crw-r--r-- | 1 | root | root | 5, 20 | oct 26 00:13 | caracteres |

## Sesión 4: Automatización de Tareas

### 4.1 Consulta de información sobre procesos demonio`

- **ps aux | grep atd**  
 Se ve que no tiene terminal asociado, '?' donde debería poner pts/\*, y su usuario es daemon. Buscamos cual es su padre con la orden **ps -p PPID -f**. SU padre sería 1 (init).

```
laura@laura-VirtualBox:~$ ps aux | grep atd
daemon 3528 0.0 0.1 28332 1952 ? Ss 06:55 0:00 /usr/sbin/atd -f
laura 7102 0.0 0.1 21536 1060 pts/0 S+ 10:10 0:00 grep --color=auto atd
laura@laura-VirtualBox:~$ ps -p 3528 -f
UID PID PPID C STIME TTY TIME CMD
daemon 3528 1 0 06:55 ? 00:00:00 /usr/sbin/atd -f
```

- **ps aux | grep cron**  
 No tiene terminal asociado, su padre es 1 (init) y el usuario, root.

```
laura@laura-VirtualBox:~$ ps aux | grep cron
root 522 0.0 0.1 38428 1772 ? Ss 04:42 0:00 /usr/sbin/cron -f
laura 7136 0.0 0.1 21536 1064 pts/0 S+ 10:15 0:00 grep --color=auto cron
laura@laura-VirtualBox:~$ ps -p 522 -f
UID PID PPID C STIME TTY TIME CMD
root 522 1 0 04:42 ? 00:00:00 /usr/sbin/cron -f
```

### 4.2 Ejecución postergada de órdenes con at (I)

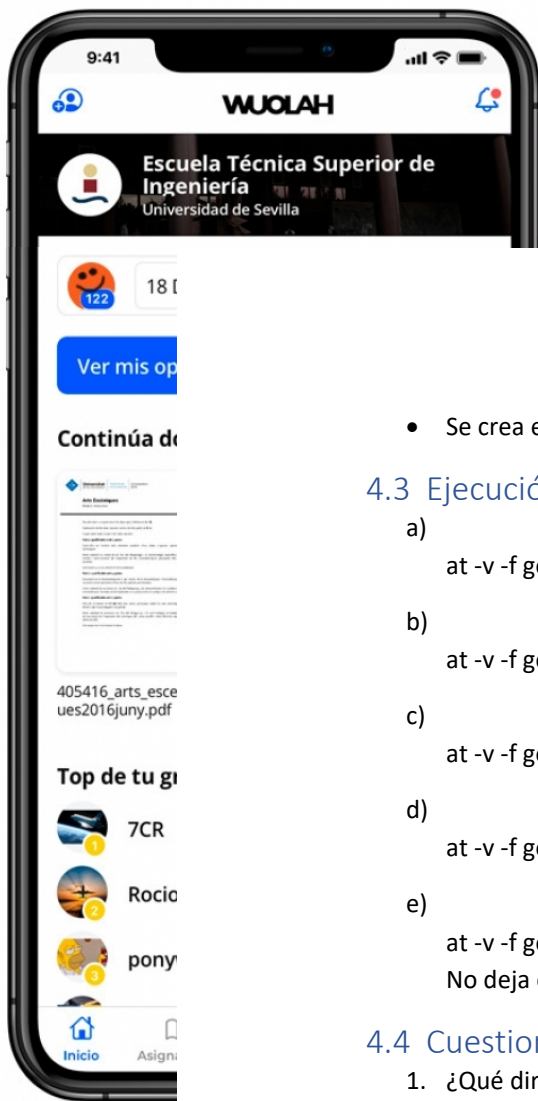
#!/bin/sh

ls >> listahome-`date +%Y-%j-%T-\$`

chmod +x genera\_apunte

at -f ./genera\_apunte.sh now + 1 minutes

job 2 at Fri Oct 26 00:39:00 2018



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



- Se crea en el directorio desde el que se lanza el demonio.

## 4.3 Ejecución postergada de órdenes con at (II)

- a) `at -v -f genera_apunte midnight`
- b) `at -v -f genera_apunte midnight + 1 minutes`
- c) `at -v -f genera_apunte 17:30 27 Oct [2018]`
- d) `at -v -f genera_apunte 25 Dec [2018]`
- e) `at -v -f genera_apunte 00:00 1 Jan 2018`  
No deja crearla porque no puede crear trabajos para el pasado

## 4.4 Cuestiones sobre at

1. ¿Qué directorio de trabajo tiene inicialmente? ¿hereda el que tenía el proceso que invocó a at o bien es el home, directorio inicial por omisión?  
- El directorio de trabajo se mantiene desde el momento de la invocación, así que hereda el que tenía el proceso que invocó a at.
2. ¿Qué máscara de creación de archivos umask tiene? ¿es la heredada del padre o la que se usa por omisión?  
- La máscara es 0022 y se hereda, se mantiene desde el momento de la invocación.
3. ¿Hereda las variables locales del proceso padre?  
- El entorno también se mantiene desde el momento de la invocación, excepto las variables **BASH\_VERSION**, **DISPLAY**, **EUID**, **GROUPS**, **SHELLOPTS**, **TERM**, **UID**, y **\_**.

## 4.5 Relación padre-hijo con órdenes ejecutadas mediante at

#!/bin/bash

```
nombreakarchivo=$(date +"%Y-%d-%T")
ps -ef > $nombreakarchivo
echo Mi pid = $$ >> $nombreakarchivo

at -f ./padre-hijo.sh now + 1 minutes

cat <nombre generado>
```

## 4.6 Script para orden at

```
#!/bin/sh
```

```
fecha=$(date +"%Y-%d-%T")
```

```
nombre="modificados_.$fecha"
```

```
find - -atime -1 >> $nombre
```

```
at -f ./script.sh now+1 minutos
```

## 4.7 Utilización de las colas de trabajo de at

```
At -q e -f prueba_procesos now + 2 minutes (asignar un trabajo a una cola)
```

```
at -q d -f prueba_procesos now + 2 minutes
```

```
at -q c -f prueba_procesos now + 2 minutes
```

```
atq -q c (visualizar los trabajos asignados a la cola c)
```

## 4.8 Relación padre-hijo con órdenes ejecutadas mediante crontab

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
#minuto hora día-mes mes día-semana orden
```

```
* * * * * /home/Laura/script.sh
```

Crontab <nombre\_fichero\_crontab>

Los asteriscos indican que se ejecuta cada minuto de cada hora de cada día del mes cada semana.

## 4.9 Ejecución de scripts con crontab (I)

```
#!/bin/sh
```

```
rm -v /tmp/varios/Core* >> /tmp/listacores
```

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
minuto hora dia-mes mes dia-semana orden
```

```
* * * * * /home/Laura/<script_anterior>.sh
```

crontab <nombre\_crontab>

**crontab -l** : muestra la lista actual de trabajos.

## 4.10 Ejecución de scripts con crontab (II)

```
#!/bin/bash
```

```
tail -n 10 /tmp/listacores > /tmp/temporal
```

```
rm /tmp/listacores
mv /tmp/temporal /tmp/listacores
```

```
#Nombre: cron_5.3
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/reducelista
```

```
crontab cron_5.3
```

#### 4.11 Ejecución de scripts con crontab (III)

```
#Nombre: script_5.4.sh
#!/bin/bash
fecha=`date +%Y-%j-%T`
echo $fecha - $PATH >> /home/superjes/SO/Practica_1/listabusqueda
```

```
#Nombre: cron_5.4
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
#minuto hora día-del-mes mes , día-de-la-semana orden
* * * * * /home/superjes/SO/Practica_1/script_5.4.sh
```

```
cat listabusqueda
```

```
crontab cron_5.4
```

```
cat listabusqueda
```

Como podemos ver, la variable **PATH** es diferente según si lo ejecutamos directamente o con **crontab**, dado que la hemos redefinido dentro del fichero en formato crontab.

#### 4.12 Variables de entorno en archivos crontab

```
#!/bin/bash
#Nombre: script_5.5.sh
echo "SHELL=/bin/sh" > crontab-reducelista
echo "PATH=`pwd`/:$HOME/SO:$PATH" >> crontab-reducelista
echo "* * * * * reducelista" >> crontab-reducelista
```

```
SHELL=/bin/sh
PATH=/home/superjes/SO/Practica_1:/home/superjes/SO:/usr/local/sbin:/usr/local/b
in:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
* * * * * reducelista
```

```
crontab crontab-reducelista
```

#### 4.13 Archivos crontab de diferentes usuarios

```
#!/bin/bash
#Nombre: script_5.6.sh
echo $LOGNAME > /tmp/fichero
```

```
echo $HOME >> /tmp/fichero
```

```
#Nombre: cron_5.6
```

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

```
* * * * * /home/superjes/SO/Practica_1/script_5.6.sh
```

Con otro usuario, lanzo con la orden **crontab** el fichero anterior:

```
$ crontab /home/superjes/SO/Practica_1/cron_5.6
```

```
$ cat /tmp/fichero
```

```
pato
```

```
/home/pato
```

**Los valores se toman del usuario que lanza el comando crontab.**

#### 4.14 Ejecución de scripts con crontab (IV)

```
#!/bin/bash
```

```
#Nombre: script_5.7.sh
```

```
find ~ -mtime 1 | cpio -pmduv /tmp/salvado$HOME
```

```
#Nombre: cron_5.7
```

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
0 0 * * * /home/superjes/SO/Practica_1/script_5.7.sh
```

```
crontab cron_5.7
```

#### 4.15 Gestión del servicio crond como usuario root

1.

- Como root añado (o creo si no existe) al archivo **/etc/cron.deny** una nueva línea con el nombre del usuario (en este caso el usuario **pato**) al que quiero prohibir el uso de **crontab**:

```
echo pato >> /etc/cron.deny
```

- Nos metemos como el usuario que acabamos de indicar y probamos a ejecutar crontab:

```
su pato
```

Contraseña:

```
pato@superjes-VirtualBox:/home/superjes/SO/Practica_1$ crontab cron_5.6
```

**You (pato) are not allowed to use this program (crontab)**

**See crontab(1) for more information**

- Como podemos ver, efectivamente no tenemos permisos para ejecutar crontab.

1.

```
service cron stop
```

```
cron stop/waiting
```

```
service cron start
```

```
cron start/running, process 21796
```