

# WUOLAH



Rubenjo7

[www.wuolah.com/student/Rubenjo7](http://www.wuolah.com/student/Rubenjo7)



61999

## examenmodulo1.pdf

*Exámenes 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**

1.-

a)

El usuario lo creamos con : adduser nombreusuario

```
[root@localhost ~]# adduser examen
```

Para averiguar el directorio home lo hacemos de la siguiente manera:

1º nos logueamos como el usuario que hemos creado

2º ponemos la orden cd para ir a nuestro directorio principal

3º ponemos la orden pwd para ver la ruta en el sistema

```
[root@localhost examen]# su examen
[examen@localhost ~]$ cd
[examen@localhost ~]$ pwd
/home/examen
```

b)

Lo haríamos haciendo uso de la orden chage

```
root@raul-msi:/home/raul/Escritorio# chage examen
cambiando la información de la edad para examen
introduzca el nuevo valor, o presione INTRO para el predeterminado

Duración mínima de la contraseña [0]:
Duración máxima de la contraseña [99999]:
Último cambio de contraseña (AAAA-MM-DD) [2015-10-27]:
Aviso de caducidad de la contraseña [7]:
Contraseña inactiva [-1]:
Fecha de caducidad de la cuenta (AAAA-MM-DD) [-1]: 2015-11-27
```

y en el campo de fecha de caducidad introducimos la fecha

2.-

a)

primero lo creamos con mknod

```
raul@raul-msi:/dev$ sudo mknod /dev/loop0 b 7 0
```

creamos el archivo de 10MB

```
raul@raul-msi:/dev$ sudo dd if=/dev/zero of=/root/archivo_SA10 bs=1k count=10000
10000+0 registros leídos
10000+0 registros escritos
10240000 bytes (10 MB) copiados, 0,0522426 s, 196 MB/s
```

y asociamos ambos

```
raul@raul-msi:/dev$ sudo losetup /dev/loop0 /root/archivo_SA10
raul@raul-msi:/dev$
```

vemos lo que contiene con fdisk

```
raul@raul-msi:/dev$ sudo fdisk -l /dev/loop0

Disco /dev/loop0: 10 MB, 10240000 bytes
255 cabezas, 63 sectores/pista, 1 cilindros, 20000 sectores en total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x00000000
```

b)

Para crear el sistema de archivos ext3, primero debemos hacer un formateo lógico en ext3

```
raul@raul-msi:/dev$ sudo mkfs.ext3 -L 'LABEL_ext3' /dev/loop0
mke2fs 1.42.9 (4-Feb-2014)
Discarding device blocks: hecho
Etiqueta del sistema de ficheros=LABEL_ext3
OS type: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
2512 inodes, 10000 blocks
500 blocks (5.00%) reserved for the super user
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=10485760
2 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
1256 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193

Allocating group tables: hecho
Escribiendo las tablas de nodos-i: hecho
Creating journal (1024 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
o

raul@raul-msi:/dev$
```

Tamaño bloque lógico: 1024  
 Número de inodos: 2512  
 Número de bloques: 10000  
 Máximo número de montajes: 0

c)

Los pasos a realizar son:

- 1º hacemos un: `yum -y update`, para que nos actualice todos los paquetes.
- 2º hacemos un: `yum install "paquete1 paquetequedepependedepaquete1"`



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



3.-

a)

```
top - 12:16:19 up 42 min, 3 users, load average: 0,66, 0,41, 0,24
Tareas: 184 total, 5 ejecutar, 171 hibernar, 0 detener, 0 zombie
%Cpu(s): 51,1 usuario, 0,3 sist, 0,0 adecuado, 48,6 inact, 0,0 en espera, 0,
KiB Mem: 4048224 total, 1444384 used, 2603840 free, 113820 buffers
KiB Swap: 1046524 total, 0 used, 1046524 free. 576896 cached Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
4701	raul	20	0	12912	1324	932	R	100,0	0,0	0:14.34	burbuja

He lanzado un algoritmo de ordenación burbuja.

Con la orden 'top' podemos ver el estado de este proceso.

La prioridad del proceso la da el campo NI, en nuestro caso es 0, cuanto más negativo más prioridad.

El estado actual se puede ver ahí, por ejemplo está consumiendo un 100% de CPU, lleva ejecutandose 14,34s.

b)

```
raul@raul-msi: ~/Escritorio
raul@raul-msi:~/Escritorio$ ln pruebaexamen.txt hardlink
raul@raul-msi:~/Escritorio$ ls -l
total 1155960
-rwxrwxrwx 1 raul raul 74 oct 24 16:53 arranque.sh
-rwxrwxrwx 1 raul raul 74 oct 24 00:58 arranque.sh~
-rwxrwxr-x 1 raul raul 192 oct 20 16:43 cronreduce.sh~
-rw-rw-r-- 1 raul raul 167 oct 20 16:43 crontab-reducelista~
-rw-rw-r-- 1 raul raul 1073741824 oct 27 12:01 Fedora14-x86-root_fs
-rw-rw-r-- 2 raul raul 20 oct 27 12:20 hardlink
-rw-rw-r-- 1 raul raul 16 oct 26 11:51 hardlink~
-rwxrwxr-x 1 raul raul 83814586 sep 24 13:13 kernel32-3.0.4
-rw-rw-r-- 1 raul raul 26103794 oct 21 09:53 kernel32-3.0.rar
-rw-rw-r-- 1 raul raul 0 oct 20 16:08 prueba~
-rw-rw-r-- 2 raul raul 20 oct 27 12:20 pruebaexamen.txt
-rw-rw-r-- 1 raul raul 0 oct 26 11:47 prueba.txt~
-rw-rw-r-- 1 raul raul 0 oct 27 12:20 pruebaexamen.txt~
-rwxrwxr-x 1 raul raul 59 oct 20 13:18 script46.sh~
```

Como podemos observar en la 2ª columna, los archivos que tienen enlaces están marcados con 2 en vez de con 1.

En la imagen se puede observar que ambos archivos pesan lo mismo y tienen las mismas propiedades.

Si hubiésemos creado un enlace suave (acceso directo) nos saldría en el nombre del archivo una flechita --> al que accede.

c)

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

```
cd $1
```

```
du -h -B 2 > salida
```

```
tail -n 1 salida > salida2
```

```
rm salida
```

Con ese Script accederíamos al directorio y mostraríamos el número de bloques en la última línea del fichero salida, lo recortamos con tail y lo metemos en el fichero 2, ya solo tendríamos que hacer

un cat de fichero 2 para ver el resultado

d) Este apartado lo hacemos con un cron.

El cron sería este

#Nombre: LOG

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

\* 13 \* 11 \* /home/raul/Escritorio/examen.sh > LOG:`date +%Y-%j-%T`

Como al demonio no se le puede pasar argumentos he modificado el Script para que siempre lo haga del directorio actual (pwd)

Finalmente lo ejecutamos:

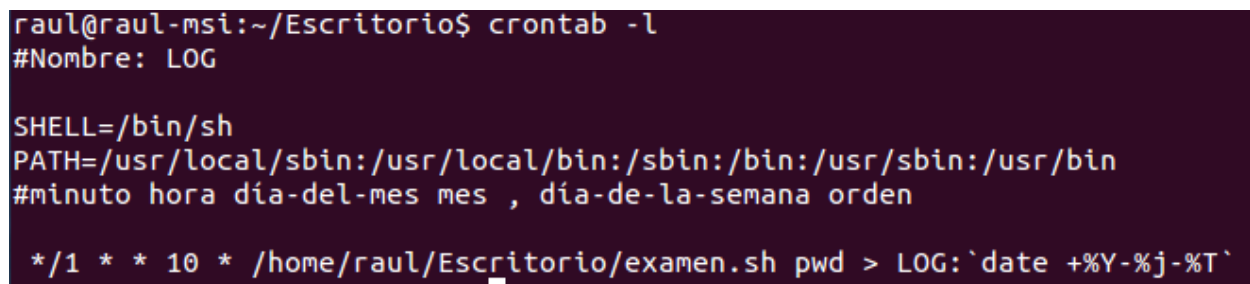


The screenshot shows a terminal window with the title 'raul@raul-msi: ~/Escritorio'. The user runs 'pwd' and the output is '/home/raul/Escritorio'. Then, the user runs 'crontab LOG' to add a new cron job. On the left side of the terminal, there are icons for 'examen.sh' and a file named 'LOG' with a 'cron' icon.

Para probar que funciona lo pruebo cada minuto, lo haríamos de la siguiente forma:

\*/1 \* \* 10 \* /home/raul/Escritorio/examen.sh > LOG:`date +%Y-%j-%T`

Aquí veríamos que está activado:



The screenshot shows a terminal window with the title 'raul@raul-msi: ~/Escritorio'. The user runs 'crontab -l' to list the current cron jobs. The output shows the configuration for the 'LOG' crontab, including the SHELL, PATH, and the cron job entry: '\* /1 \* \* 10 \* /home/raul/Escritorio/examen.sh pwd > LOG:`date +%Y-%j-%T`'.

La imagen esta es anterior a que modificara el script para que directamente buscara en pwd.

El Script me funciona correctamente pero, el cron no me lo acaba de hacer.