

Para configurar el menú de arranque hay que acceder a la carpeta `/etc/grub.d`, donde se encuentran los ficheros de configuración. Por ejemplo, para añadir en el menú de arranque una nueva opción para poder iniciar un sistema operativo Windows hay que añadir al final del fichero `/etc/grub.d/40_custom` las siguientes líneas:

```
menuentry 'Windows XP' {
    insmod ntfs
    set root=(h0,1)
    chainloader +1
}
```

Finalmente, para aplicar los cambios en el sistema hay que ejecutar:

```
#update-grub2
```

## ACTIVIDADES 11.1



- Configure GRUB para que muestre el menú de arranque durante 10 segundos.

### 11.1.2 PROCESO DE ARRANQUE Y PARADA DEL SISTEMA

#### 11.1.2.1 Niveles de ejecución

Una vez que se ha encontrado el *kernel* y se ha iniciado. El sistema operativo comienza a cargarse, se inicia el hardware, los discos están preparados, se asignan direcciones IP, se inician servicios, y se realizan otras muchas tareas. Para ello, Linux ejecuta el programa *init*, cuya función es iniciar el sistema operativo y sus servicios. Las tareas que realiza el proceso *init* son:

- Comprueba los sistemas de ficheros.
- Monta los sistemas de ficheros permanentes.
- Activa la zona de memoria *swap* o de intercambio.
- Activa los demonios o servicios del sistema (p.e.: *atd*, *syslog*).
- Activa la red.
- Inicia los demonios o servicios de red del sistema (p.e.: *sendmail*, *httpd*).
- Limpia los sistemas de ficheros temporales.
- Finalmente, habilita el *login* a los usuarios del sistema.

El proceso *init* es el estándar para iniciar y apagar equipos Linux y Unix llamado *SysV*. *SysV* es un modo de definir qué estado debe tener el equipo en un momento determinado. Para ello se emplea un concepto denominado modo de ejecución (o *runlevels*).

*SysV* utiliza siete modos de ejecución que van del 0 al 6, y cada distribución utiliza los modos de ejecución para diferentes fines aunque hay varios niveles que son comunes. Los niveles que son comunes son: el 0 se utiliza para apagar el equipo; el 1 es el modo monousuario; y el 6 se utiliza para reiniciar el equipo.

Los demás modos de ejecución (del 2 al 5) difieren ligeramente en Ubuntu y Fedora. En Ubuntu los modos del 2 al 5 se ejecutan en modo multiusuario mientras que en los sistemas basados en RedHat, como es el caso de Fedora, cambian ligeramente. En la Tabla 11.2 se muestran los modos de ejecución para las distribuciones Ubuntu y Fedora.

**Tabla 11.2.** Modos de ejecución

Modo	Ubuntu	Fedora
0	Apaga el equipo.	Apaga el equipo.
1	Modo monousuario.	Modo monousuario (mantenimiento).
2	Modo multiusuario.	Sin asignar.
3	Modo multiusuario.	Multiusuario en modo comandos.
4	Modo multiusuario.	Sin asignar.
5	Modo multiusuario.	Multiusuario con entorno gráfico.
6	Reinicia el equipo.	Reinicia el equipo.

A continuación se van a ver las tareas más frecuentes sobre el nivel de ejecución del sistema:

- Si lo desea, puede **cambiar el nivel de ejecución del sistema por defecto** de la siguiente forma:

#### ● UBUNTU

Modifique el fichero `/etc/init/rc-sysinit.conf` de la siguiente forma:

```
env DEFAULT_RUNLEVEL=2
```

#### ● FEDORA

Modifique el fichero `/etc/inittab` de la siguiente forma:

```
id:3:initdefault:
```

donde el 3 es el modo de ejecución del sistema.

- Para **ver el nivel de ejecución** que tiene actualmente el sistema debe ejecutar:

```
# runlevel
```

- Para **cambiar manualmente el nivel de ejecución del sistema** hay que ejecutar:

```
# telinit 3
```

```
o
```

```
# init 3
```

Cada nivel de ejecución, tiene asociado un directorio donde se especifican los servicios que se deben ejecutar o parar. Por ejemplo, el directorio `/etc/rc0.d` corresponde al nivel 0, el directorio `/etc/rc1.d` al nivel 1, etc.

Ahora bien, ¿cómo puedo ver los *scripts* que se ejecutan en un determinado nivel? Existen varias formas de ver los servicios asociados a un determinado nivel. Por ejemplo, si muestra el contenido del directorio:

```
cd /etc/rc3.d
ls -l
```

obtiene una salida como la siguiente:

```
lrwxrwxrwx 1 root root 17 3:11 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 16 3:11 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 14 3:32 S40cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14 3:11 S50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 13 3:11 S60nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 15 3:11 S70nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 18 3:11 S90lpd -> ../init.d/lpd.init
lrwxrwxrwx 1 root root 11 3:11 S99local -> ../rc.local
```

Como se puede observar, el directorio contiene enlaces simbólicos a *scripts* del directorio */etc/init.d*. Cada enlace tiene una letra (S o K) y un número al principio. El número establece el orden en el que se van a ejecutar los servicios mientras que la letra “S” significa que se inicia y la “K” que se para el servicio correspondiente.

¿Cómo hace el proceso *Init* para arrancar y parar los servicios? Sencillo. Cada uno de los *scripts* se escribe para aceptar un argumento que suele ser *start*, *stop*, *status*, *restart* o *reload*. Si lo desea puede ejecutar los *scripts* manualmente.

Por ejemplo, si quiere ver las opciones de un determinado servicio puede ejecutarlo directamente:

```
# /etc/init.d/httpd
Uso: ./httpd {start|stop|restart|condrestart|reload|status|fullstatus|graceful|help|configtest}
```

Por lo tanto, si quiere parar el servidor de páginas web manualmente ejecute:

```
# /etc/init.d/httpd stop
```

También puede administrar los servicios utilizando el comando *service* de la siguiente forma:

```
# service httpd stop
```

Una vez realizados todos los pasos que establece el nivel de ejecución, se procesa el fichero */etc/rc.local*. Este fichero es un “cajón desastre” donde se pueden escribir todos los comandos que el sistema ejecuta al iniciarse.

## ACTIVIDADES 11.2



➤ Determine el nivel de ejecución en el que se ejecuta el sistema.

### 11.1.3 SERVICIOS DEL SISTEMA

Los servicios son aplicaciones que se ejecutan, en segundo plano, independientemente del usuario y ofrecen una determinada funcionalidad. Normalmente se asocia el término “servicio” solo a servicios de red (p.e.: servidor web, servidor FTP) pero existen servicios que ofrecen todo tipo de funcionalidades (gestionan las conexiones de red, monitorizan el sistema, comprueban las actualizaciones y seguridad del sistema, permiten utilizar el hardware del equipo, etc.).

El administrador de servicios permite establecer los servicios que se van a ejecutar al iniciar el sistema, y permite parar, ejecutar o reanudar los servicios que se ejecutan actualmente en el sistema.

A continuación se van a ver varias formas de administrar los servicios en el sistema.

11.1.3.1 Herramientas gráficas

Ubuntu

En Ubuntu es posible administrar los servicios de modo gráfico utilizando *Boot Up Manager* en x-Windows o *sys-rc-conf* en el terminal del sistema.

Para ejecutar *Boot Up Manager* primero hay que instalarla ejecutando:

```
# apt-get install bup
```

A continuación, desde el entorno gráfico ejecute *Boot-up Manager*, que se encuentra en *System / Administration* (véase la Figura 11.3).



Figura 11.3. Boot-up Manager

Si desea administrar los servicios del sistema en modo terminal entonces debe instalar la herramienta *sysv-rc-config*:

```
# apt-get install sysv-rc-config
```

Una vez instalada ejecute en el terminal:

```
# sysv-rc-config
```

y en la pantalla que se muestra en la Figura 11.4 habilite o deshabilite los servicios que estime oportunos:

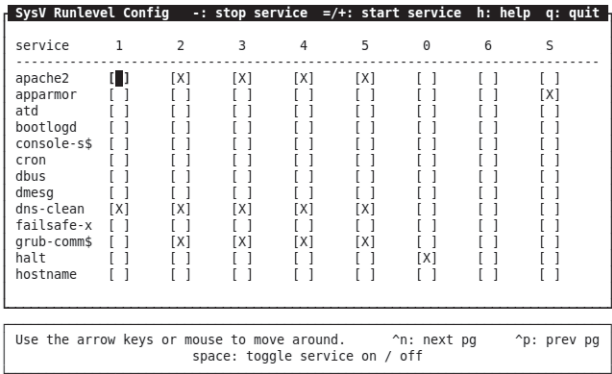


Figura 11.4. sysv-rc-conf

Fedora

En los sistemas Fedora es posible administrar los servicios del sistema a través de la herramienta de x-Windows *Configuración del servicio*. Si desea configurar los servicios de forma gráfica por terminal entonces debe ejecutar la herramienta *ntsysv*.

Para ejecutar el administrador de servicios (véase la Figura 11.5) ejecute *system-config-services* desde un terminal o desde el entorno gráfico ejecute *servicios* que se encuentra en *Administración* → *Servicios*.

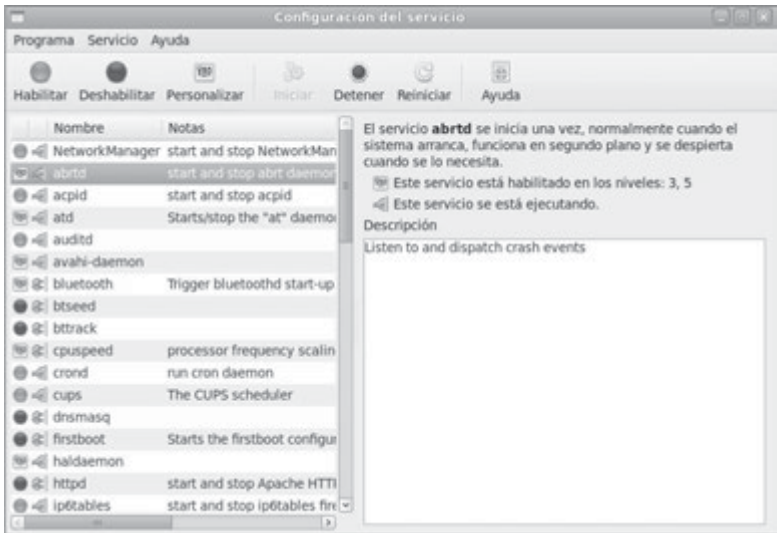


Figura 11.5. Administrador de servicios (x-Windows)

*ntsysv* es una utilidad que se puede ejecutar desde la línea de comandos o desde el programa *setup*, opción *System services*. Para utilizarlo, tal y como muestra la Figura 11.6, debe ir marcando los servicios que quiere que se ejecuten de forma automática.



Figura 11.6. Administrador de servicios (ntsysv)

### 11.1.3.2 chkconfig

*chkconfig* permite administrar los servicios que se van a iniciar automáticamente cuando arranca el sistema.

#### ● UBUNTU

Para utilizar *chkconfig* en Ubuntu antes es necesario instalarlo ejecutando:

```
# apt-get install chkconfig
```

Para ver el estado de los diferentes servicios ejecute:

```
# chkconfig --list
```

Donde en cada fila muestra un determinado servicio y en cada columna se indica si el servicio se inicia automáticamente en ese modo de ejecución (del modo 0 al 6).

Por ejemplo, si quiere que el servidor web se ejecute automáticamente ejecute:

```
# chkconfig httpd on
```

Si desea activarlo en los niveles 235 ejecute:

```
# chkconfig --levels 235 httpd on
```

Y si desea deshabilitarlo el ejecute:

```
# chkconfig httpd off
```

named	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
netconsole	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
netfs	0:desactivado	1:desactivado	2:desactivado	3:activo	4:activo	5:activo	6:desactivado
network	0:desactivado	1:desactivado	2:desactivado	3:activo	4:desactivado	5:desactivado	6:desactivado
nfs	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
nfslock	0:desactivado	1:desactivado	2:desactivado	3:activo	4:activo	5:activo	6:desactivado
nsd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
ntpd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
ntpdate	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
openvpn	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
pcscd	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
portreserve	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
psacct	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
rdisc	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
restorecond	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
rpcbind	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
rpcgssd	0:desactivado	1:desactivado	2:desactivado	3:activo	4:activo	5:activo	6:desactivado
rpcidmapd	0:desactivado	1:desactivado	2:desactivado	3:activo	4:activo	5:activo	6:desactivado
rpcsvcgssd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
rsyslog	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
ssslauthd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
sendmail	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
smartd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
smolt	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
snmpd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
snmptrapd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
sshd	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
sssd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
udev-post	0:desactivado	1:activo	2:activo	3:activo	4:activo	5:activo	6:desactivado
vnscs-server	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
webmin	0:desactivado	1:desactivado	2:activo	3:activo	4:desactivado	5:activo	6:desactivado
wpa_supplicant	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
xpbind	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado

Figura 11.7. *chkconfig --list*

## ACTIVIDADES 11.3



- Examine los servicios que se ejecutan en el sistema.
- Inicie el servidor de páginas web *Apache*. Accede a un navegador web y comprueba que el servicio funciona correctamente.
- Ejecute el comando *chkconfig* para ver los servicios que se encuentran activos en el sistema.

### 11.1.4 PROCESOS

En los sistemas GNU/Linux se ejecutan una gran cantidad de servicios que permiten realizar una determinada actividad en el sistema. Cada servicio o demonio consiste en uno o más procesos que se ejecutan en el equipo.

Además de los procesos vinculados a servicios, en el sistema se encuentran los procesos que ejecuta un usuario. Por ejemplo, un editor de textos, un navegador Web, etc. A continuación se van a ver las herramientas que permiten gestionar los procesos del sistema.

#### 11.1.4.1 ps

El comando *ps* permite ver los procesos que se están ejecutando en el sistema. Tal y como se muestra en la Figura 11.8, para cada proceso se muestra su identificador (PID), terminal donde se ejecuta (TTY), tiempo de uso de CPU (TIME) y el comando que ejecuta (CMD).

```

root@server:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@server ~]# ps
  PID TTY          TIME CMD
 9551 pts/0    00:00:00 bash
18318 pts/0    00:00:00 ps
[root@server ~]#

```

Figura 11.8. Ps

Si desea ver todos los procesos que se ejecutan en el sistema utiliza la opción *-A*:

```
$ ps -A
```

Si desea eliminar un proceso que se está ejecutando en el sistema puede utilizar el comando *kill* de la siguiente forma:

```
$ kill -9 <ID del proceso>
```

#### 11.1.4.2 top

Es una aplicación que, en tiempo real, informa sobre la actividad del sistema. Proporciona información sobre la carga del sistema operativo, grado de utilización de la CPU, memoria y *swap*, y los procesos que se encuentran en ejecución (véase la Figura 11.9).

```

top - 00:42:06 up 25 min, 1 user, load average: 0.07, 0.02, 0.00
tasks: 73 total, 1 running, 72 sleeping, 0 stopped, 0 zombie
psu(s): 0.30s, 0.30s, 0.00s, 99.30s, 0.00s, 0.00s, 0.00s, 0.00s
Mem: 508132k total, 83828k used, 424312k free, 11864k buffers
Swap: 795212k total, 0k used, 795212k free, 39828k cached

```

PID	USER	PPID	NI	UPT	STK	SIZE	VSZ	SS	RES	%CPU	%MEM	TIME	COMMAND
1224	root	20	0	2492	1960	844	8	0.2	0.2	0:00.03	0.00	0:00.03	top
1	root	20	0	2860	1624	1220	3	0.0	0.3	0:01.24	0.00	0:01.24	init
2	root	20	0	0	0	0	0.3	0.0	0.0	0:00.00	0.00	0:00.00	ktlthread
3	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.01	0.00	0:00.01	konftirqd-0
4	root	RT	0	0	0	0.3	0.0	0.0	0.0	0:00.00	0.00	0:00.00	migration-0
5	root	RT	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	watchdog-0
6	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	events-0
7	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	cpuset
8	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	khelper
9	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	netns
10	root	20	0	0	0	0.3	0.0	0.0	0.0	0:00.00	0.00	0:00.00	syncmgr
11	root	20	0	0	0	0.3	0.0	0.0	0.0	0:00.00	0.00	0:00.00	ps
12	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	sync_supers
13	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	hdi-default
14	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	hntcgrrtd-0
15	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	khlschd-0
16	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	kacpid
17	root	20	0	0	0	0.5	0.0	0.0	0.0	0:00.00	0.00	0:00.00	kacpi_notify

Figura 11.9. Top

## ACTIVIDADES 11.4



- Ejecute un terminal el comando *nano texto*.
- Ejecute *ps -A*, localiza el proceso que corresponde al editor de textos y finaliza su ejecución.
- Busque e Internet información sobre: proceso, servicio, interrupción y excepción en un proceso.

### 11.1.5 PROGRAMACIÓN DE TAREAS

La programación de tareas permite programar la ejecución de un determinado programa en un momento determinado. Por ejemplo, se puede programar una copia de seguridad, enviar un fichero, comprobar la seguridad del sistema, enviar un informe, etc.

Antes de programar las tareas hay que comprobar que el servicio *crond* se encuentra en ejecución mediante el comando:

```
# service crond status
```

Para modificar el fichero de configuración de *crond*, ejecute el comando:

```
# crontab -e
```

y aparece un fichero con el siguiente formato:

```
PATH=/bin
0 0 * * * /root/comprobar_seguridad.sh
0 0 1 * * /root/copia_seguridad.sh
```

La sintaxis de las tareas programadas es:

```
# .----- minuto (0 - 59)
# | .----- hora (0 - 23)
# | | .----- día del mes (1 - 31)
# | | | .----- mes (1 - 12) o jan,feb,mar,apr ...
# | | | | .----- día de la semana (0 - 6) (Sunday=0 o 7) OR
# | | | | | sun,mon,tue,wed,thu,fri,sat
# * * * * * Comando a ejecutar
```

En el ejemplo anterior se ejecuta el *script* *comprobar\_seguridad.sh* todos los días a las 0:00 h y se ejecuta *copia\_seguridad.sh* el primer día de cada mes.



Otra forma de poder programar tareas es guardar el *script* que quiere ejecutar en las siguientes carpetas de configuración de *cron*:

/etc/cron.hourly	# Ejecuta el script cada hora
/etc/cron.daily	# Ejecuta el script diariamente
/etc/cron.weekly	# Ejecuta el script semanalmente
/etc/cron.monthly	# Ejecuta el script mensualmente



Para asegurar el sistema solo el usuario *root* puede modificar los *scripts* que ejecuta *crontab*.

Una ventaja muy interesante que permite *crontab* es que cada vez que se ejecuta la tarea manda un email con el resultado de la ejecución de dicha tarea (véase la Figura 11.10).



Figura 11.10. mail de crontab

## ACTIVIDADES 11.5



- Examine los *scripts* que se ejecutan en el sistema.
- Programe *crontab* para que cada día a las 0:05 h se eliminen todos los ficheros que se encuentran en el directorio */tmp*.

### 11.1.6 REINICIO Y PARADA DEL SISTEMA

El proceso de parada y reinicio del sistema lo establecen los nivel de ejecución 0 y 6 respectivamente. De esta forman en los respectivos directorios */etc/rc0.d* y */etc/rc6.d* puede ver los pasos que realiza el sistema para apagar o reiniciar el equipo. Por lo tanto, si desea reiniciar el equipo puede llamar al proceso *init* de la siguiente forma:

```
# init 0
```

y si quiere reiniciarlo ejecute:

```
# init 6
```

Además, puede utilizar comandos específicos para apagar el equipo como *halt* o *shutdown*, o se puede reiniciar el equipo ejecutando *reboot*.