



Universidad
Inca Garcilaso de la Vega

Nuevos Tiempos. Nuevas Ideas

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

UNIDAD IV: GESTIÓN DE PERMISOS Y PROCESOS

CONTROL Y SUPERVISIÓN DE PROCESOS

Dr. Santiago Gonzales Sánchez
rgonzales@uigv.edu.pe

Objetivo de aprendizaje

Al final de esta sesión, el estudiante:

- Reconocer los mecanismos de control y supervisión de procesos que permita verificar el estado del sistema GNU/Linux.



Gestor de Arranque

- El gestor de arranque es el encargado de iniciar cualquier sistema operativo que hay sido previamente instalado.
- El gestor de arranque utilizado en GNU/Linux es GRUB.
- GRUB es un gestor de arranque que permite gestionar el inicio de nuestro equipo entre diferentes sistemas operativos.



Gestor de Arranque

- El método que utiliza GRUB para la carga de sistemas operativos GNU/Linux se denomina **carga directa**, ya que el propio gestor de arranque es el encargado de hacerlo directamente y no existe ningún intermediario.



Gestor de Arranque

- En caso tener la carga de otro sistema operativo, como por ejemplo Microsoft Windows el método de arranque se denomina **carga encadenada**, en el que MBR (Master Boot Record o Registro de Arranque principal) indicar el primer sector de la partición que contiene el sistema operativo.



GNU GRUB version 0.97 (636K lower / 1046400K upper memory)

CentOS (3.2.48)

CentOS (2.6.32-358.14.1.el6.x86_64)

CentOS (2.6.32-358.el6.x86_64)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.

BURG version 1.98+20100623-1+maverick

/-----\

- | Ubuntu GNU/Linux, with Linux 2.6.35-28-generic
- | Ubuntu GNU/Linux, with Linux 2.6.35-27-generic
- | Ubuntu GNU/Linux, with Linux 2.6.35-22-generic
- | Windows 7 (loader) (on /dev/sda1)

-----\

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands before
booting or 'c' for a command-line.

The highlighted entry will be executed automatically in 3s.

Gestor de Arranque

- El gestor de arranque GRUB nombra los discos duros como (hd0), (hd1),..., (hdN), siendo (hd0) el primer disco duro reconocido por la **BIOS** sin hacer distinciones entre discos de tipo SCSI o IDE.
- Para hacer referencia a las particiones de un disco tiene que hacerlo respecto al disco al que pertenecen, comenzando a numerar también por cero. Por ejemplo, (hd0,0) es la primera partición del primer disco duro, (hd0,1) es la segunda partición del primer disco duro, etc.



Gestor de Arranque

Por lo tanto...

- *La BIOS del sistema comprueba y lanza la primera etapa del gestor de arranque del MBR del disco duro primario.*



Gestor de Arranque

Una vez que la segunda etapa del gestor de arranque está en memoria, presenta al usuario una pantalla gráfica mostrando los diferentes sistemas operativos o kernel que puede arrancar. En esta pantalla el usuario puede usar las flechas direccionales para escoger el sistema operativo o kernel con el que desea arrancar y presionar la tecla [Intro]. Si no se presiona ninguna tecla, el gestor de arranque carga la entrada predeterminada después de un período de tiempo de espera (también configurable).



Niveles de Ejecución

Encontrado el kernel e 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.



Niveles de Ejecución

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.



Niveles de Ejecución

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.



Niveles 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.



```
[root@localhost ~]# cat /etc/inittab
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
# Individual runlevels are started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

Niveles de Ejecución

- 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:

```
# init 3
```



Niveles de Ejecución

- Cada nivel de ejecución, tiene asociado un directorio donde se especifican los servicios que se deben ejecutar o parar.

```
[root@localhost ~]# ls /etc | grep rc[0-6]
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
[root@localhost ~]# file /etc/rc3.d
/etc/rc3.d: symbolic link to `rc.d/rc3.d'
[root@localhost ~]# _
```



Niveles de Ejecución

- Por ejemplo, el directorio `/etc/rc0.d` corresponde al nivel 0, el directorio `/etc/rc1.d` al nivel 1, etc.



Niveles de Ejecución

¿cómo puedo ver los scripts que se ejecutan en un determinado nivel?



Niveles de Ejecución

```
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
```



Niveles de Ejecución

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 pare el servicio correspondiente.



cd /etc/init.d

```
[root@localhost init.d]# ls
acpid                dc_server            kerneloops            nfslock               saslauthd
anacron              dnsmasq              killall               nscd                  sendmail
atd                  firstboot            lm_sensors            ntpd                  setroubleshoot
auditd               functions            mdmonitor             ntpdate              smartd
avahi-daemon         fuse                 messagebus            pcsd                  smolt
bluetooth            gpm                  microcode_ctl         portreserve           snmpd
btseed              haldaemon            multipathd            psacct               snmptrapd
bttrack             halt                 named                 rdisc                 squid
capi                 httpd                netconsole            restorecond           sshd
cpuspeed             iptables             netfs                 rpcbind              udev-post
crond                iptables             netplugd              rpcgssd               vsftpd
cups                 irda                 network               rpcidmapd             winbind
cups-config-daemon  irqbalance          NetworkManager       rpcsvcgssd            wpa_supplicant
dc_client            isdn                 nfs                   rsyslog              ypbind

[root@localhost init.d]# █
```



Universidad
Inca Garcilaso de la Vega

Nuevos Tiempos. Nuevas Ideas

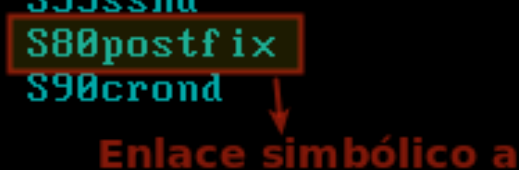
Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

Niveles de Ejecución

Dentro de estos directorios se encuentran una serie de enlaces simbólicos a los scripts que inician y detienen los servicios del sistema. Por ejemplo, en la figura podemos ver que `/etc/rc3.d/S80postfix` es un enlace simbólico que apunta a `/etc/init.d/postfix`, el cual es un script que levanta el servicio de correo del sistema. Dicho sea de paso, la `S` utilizada como prefijo en los nombres de los enlaces simbólicos indica que, en este nivel de de corrida en particular, se debe iniciar el servicio asociado.



```
[root@localhost ~]# cd /etc/rc3.d
[root@localhost rc3.d]# ls
K10saslauthd      S02lvm2-monitor  S11auditd          S25netfs          S99local
K50netconsole     S07iscsid        S12rsyslog         S26udev-post
K87multipathd     S08ip6tables    S13iscsi           S55sshd
K87restorecond   S08iptables     S15mdmonitor       S80postfix
K89rdisc          S10network      S25blk-availability S90crond
[root@localhost rc3.d]# file S80postfix
S80postfix: symbolic link to '../init.d/postfix'
[root@localhost rc3.d]# file ../init.d/postfix
../init.d/postfix: Bourne-Again shell script text executable
[root@localhost rc3.d]# _
```



A red arrow points from the text "Enlace simbólico a" to the file path `../init.d/postfix` in the terminal output. Another red arrow points from the box around `S80postfix` in the `ls` command output to the same file path.


```
[root@localhost rc6.d]# ls
```

```
K10saslauthd      K75blk-availability
```

```
K25sshd           K75netfs
```

```
K30postfix        K75udev-post
```

```
K50netconsole     K85mdmonitor
```

```
K60crond          K87multipathd
```

```
[root@localhost rc6.d]# cd
```

```
[root@localhost ~]# runlevel
```

```
N 3 ← Runlevel actual
```

```
[root@localhost ~]# _
```

```
K87restorecond
```

```
K88auditd
```

```
K88iscsi
```

```
K88rsyslog
```

```
K89iscsid
```

```
K89rdisc
```

```
K90network
```

```
K92ip6tables
```

```
K92iptables
```

```
K99lvm2-monitor
```

```
S00killall
```

```
S01reboot
```

Estos enlaces simbólicos comienzan con S porque son los que deben ejecutarse al comenzar el reinicio y el apagado del sistema

Niveles de Ejecución

¿Cómo hace el proceso Init para arrancar y parar los servicios?.

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.



Niveles de Ejecución

- 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
```



Niveles de Ejecución

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.



Herramientas gráficas

ntsysv es una utilidad que se puede ejecutar desde la línea de comandos o desde el programa setup, opción System services.



Proceso

Los procesos tienen un papel muy importante en las distribuciones GNU/Linux, ya que consumirán recursos hardware en entornos de producción, administrarlos y gestionarlos correctamente.

Los procesos son programas en ejecución por parte del usuario o del sistema. En el caso de que lo ejecute el sistema se denomina **servicios**.



Proceso

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.



Comando 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  
 10318 pts/0    00:00:00 ps  
[root@server ~]#
```

Figura 11.8. Ps



Comando ps

Opción para <code>ps</code>	Descripción
<code>-a</code>	Muestra todos los procesos con una terminal controladora, no sólo los procesos actuales del usuario.
<code>-r</code>	Muestra sólo los procesos en ejecución (vea la descripción de estados de los procesos más adelante en esta sección).
<code>-x</code>	Muestra los procesos que no tienen una terminal controladora.
<code>-u</code>	Muestra los propietarios de los procesos.
<code>-f</code>	Presenta las relaciones padre/hijo entre los procesos.
<code>-l</code>	Produce una lista en un formato largo.
<code>-w</code>	Muestra los parámetros de la línea de comandos de un proceso hasta la mitad de una línea.
<code>-ww</code>	Muestra todos los parámetros de la línea de comandos de un proceso sin importar su longitud.



Comando ps

- El conjunto más común de parámetros que se usa con el comando ps es auxww. Estos parámetros muestran todos los procesos (sin importar si tienen o no terminal controladora), los propietarios de cada proceso y todos los parámetros de las líneas de comandos. Examinemos la salida muestra de una llamada a **ps auxww** .



Comando ps

```
[yyang@serverA ~]$ ps auxww
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.2	2332	564	?	S	Mar07	0:12	init [3]
root	2	0.0	0.0	0	0	?	SN	Mar07	0:00	[ksoftirqd/0]
root	3	0.0	0.0	0	0	?	S<	Mar07	0:04	[events/0]
root	4	0.0	0.0	0	0	?	S<	Mar07	0:00	[khelper]
root	5	0.0	0.0	0	0	?	S<	Mar07	0:00	[kacpid]
root	1216	0.0	0.2	3004	504	?	S<s	Mar07	0:00	udevd
root	1732	0.0	0.0	0	0	?	S	Mar07	0:00	[kjournald]
root	1733	0.0	0.0	0	0	?	S	Mar07	0:00	[kjournald]
root	1734	0.0	0.0	0	0	?	S	Mar07	0:00	[kjournald]
root	2076	0.0	0.3	3284	584	?	Ss	Mar07	0:02	syslogd -m 0
root	2080	0.0	0.2	1792	468	?	Ss	Mar07	0:00	klogd -x
rpc	2108	0.0	0.3	2216	636	?	Ss	Mar07	0:00	portmap
rpcuser	2128	0.0	0.4	2496	844	?	Ss	Mar07	0:00	rpc.statd
root	2161	0.0	0.3	2972	588	?	Ss	Mar07	0:21	rpc.idmapd



Universidad
Inca Garcilaso de la Vega

Nuevos Tiempos. Nuevas Ideas

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

Reinicio 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*.



Comando top

- Es una versión interactiva del comando ps. En lugar de visualizar una versión estática de lo que está pasando, top refresca la pantalla con una lista de procesos cada dos o tres segundos.



File Edit View Search Terminal Help

```
top - 13:08:41 up 2:47, 2 users, load average: 1.56, 0.91, 0.43
Tasks: 136 total, 2 running, 134 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.3%us, 2.3%sy, 0.0%ni, 94.4%id, 0.0%wa, 0.4%hi, 0.8%si, 0.0%st
Mem: 3908512k total, 852084k used, 3056428k free, 45260k buffers
Swap: 4046844k total, 0k used, 4046844k free, 437588k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4	root	20	0	0	0	0	R	4.2	0.0	0:28.49	ksoftirqd/0
2003	root	20	0	183m	36m	11m	S	2.9	1.0	1:27.18	Xorg
234	root	20	0	0	0	0	S	2.0	0.0	1:06.52	scsi_eh_1
4802	centos	20	0	15032	1244	936	R	2.0	0.0	0:01.25	top
24	root	20	0	0	0	0	S	1.0	0.0	1:18.00	ata_sff/0
7	root	20	0	0	0	0	S	0.7	0.0	1:38.66	events/0
2340	centos	20	0	288m	12m	9628	S	0.7	0.3	0:04.93	gnome-terminal
36	root	39	19	0	0	0	S	0.3	0.0	0:03.08	khugepaged
1859	root	20	0	80896	3436	2532	S	0.3	0.1	0:01.69	master
2258	centos	20	0	251m	7312	5760	S	0.3	0.2	0:03.62	gnome-power-man
2274	root	20	0	45224	672	404	S	0.3	0.0	0:54.85	udisks-daemon
4663	postfix	20	0	80976	3396	2524	S	0.3	0.1	0:00.19	pickup
1	root	20	0	19352	1588	1272	S	0.0	0.0	0:21.79	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.11	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:04.88	watchdog/0

Comando free

Muestra la cantidad de memoria libre y usada que tiene el sistema. Por una parte muestra la memoria física y por otra la swap, también muestra la memoria caché y de buffer consumida por el Kernel.

```
[centos@localhost Downloads]$ free
```

	total	used	free	shared	buffers	cached
Mem:	3908512	851160	3057352	3752	45324	437588
-/+ buffers/cache:		368248	3540264			
Swap:	4046844	0	4046844			



Comando du

"du"(Uso de disco) es comando Linux / Unix podemos comprobar el volumen ocupado (uso de la información) en el disco duro de carpetas si archivos.

```
[root@localhost boot]# du -sh /boot  
37M      /boot
```



Comando du

```
[root@localhost boot]# du -sh /boot/*
107K    /boot/config-2.6.32-642.el6.x86_64
256K    /boot/efi
276K    /boot/grub
25M     /boot/initramfs-2.6.32-642.el6.x86_64.img
4.9M    /boot/initrd-2.6.32-642.el6.x86_64kdump.img
13K     /boot/lost+found
212K    /boot/symvers-2.6.32-642.el6.x86_64.gz
2.5M    /boot/System.map-2.6.32-642.el6.x86_64
4.1M    /boot/vmlinuz-2.6.32-642.el6.x86_64
```



Comando df

Provee información sobre la utilización del espacio en disco en los diferentes sistemas de archivos montados en el sistema. La sintaxis de df es la siguiente:

df [opciones] [sistema-de-archivo...]

- k: Mostrar los tamaños en bloques de 1024 bytes.
- m: Mostrar los tamaños en bloques de mega-bytes.



Comando df

```
[root@localhost boot]# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda3	47187896	2456416	42327796	6%	/
tmpfs	1954256	232	1954024	1%	/dev/shm
/dev/sda1	289293	39539	234394	15%	/boot

```
[root@localhost boot]# df -m
```

Filesystem	1M-blocks	Used	Available	Use%	Mounted on
/dev/sda3	46082	2399	41336	6%	/
tmpfs	1909	1	1909	1%	/dev/shm
/dev/sda1	283	39	229	15%	/boot



Resumen

- Se abordó los mecanismos de control y supervisión de procesos que permita verificar el estado del sistema GNU/Linux.





— Universidad —
Inca Garcilaso de la Vega
Nuevos Tiempos. Nuevas Ideas

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones