LINUX BÁSICO

Administración de Servidores - Manuel Jesús Cobo Martín
Universidad de Cádiz

COMANDOS BÁSICOS

COMANDOS BÁSICOS

man

COMANDOS BÁSICOS

- cd: cambiar el directorio

```
cd [options][file(s)]
```

- ls: muestra el contenido de un directorio

```
ls [options][file(s)]
```

- cp: copiar ficheros / directorios

```
cp [options] sourceFile targetFile
```

- mv: mover ficheros / directorios

```
mv [options] sourceFile targetFile
```

- rm: eliminar ficheros / directorios

```
rm [options][file(s)]
```

COMANDOS BÁSICOS

- ln: crea un enlace del archivo fuente al destino con un nombre diferente

```
ln [options] sourceFile targetFile
```

```
ln -s sourceFile targetFile
```

- mkdir
- rmdir
- tar: empaquetar archivos

```
tar -cvf paquete.tar /dir/
```

```
tar -xvf paquete.tar
```

```
tar -cvfz paquete.tar.gz /dir
```

```
tar -xvfz paquete.tar.gz
```

COMANDOS BÁSICOS

- ln: crea un enlace del archivo fuente al destino con un nombre diferente

```
ln [options] sourceFile targetFile
```

```
ln -s sourceFile targetFile
```

- mkdir
- rmdir
- tar: empaquetar archivos

```
tar -cvf paquete.tar /dir/
```

```
tar -xvf paquete.tar
```

```
tar -cvfz paquete.tar.gz /dir
```

```
tar -xvfz paquete.tar
```

COMANDOS BÁSICOS

- du: estimar el espacio del disco duro utilizado
- pwd: mostrar el directorio local
- uname: mostrar información del sistema
- whoami: mostrar el usuario actual
- who: mostrar quien está conectado
- history: mostrar el historial de ordenes introducidas
- touch: actualiza la fecha de modificación y acceso de un fichero
- cal: mostrar el calendario
- date: mostrar la fecha actual
- cat: mostrar el contenido de un archivo
- find: buscar archivos en la jerarquía de directorios

SHELL BÁSICO

SHEL BÁSICO

Redirecciones y tuberías

SHEL BÁSICO: REDIRECCIONES Y TUBERÍAS

- Cada proceso tiene al menos tres canales de comunicación disponibles:
  - Entrada estándar: STDIN
  - Salida estándar: STDOUT
  - Salida estándar de error: STDERR
- Los procesos no tiene porque conocer el medio final en el que se recibe o muestran los valores
  - Se pueden enviar a un archivo, red, o al canal de otro programa
- UNIX ha unificado las E/S, en el que cada canal es llamado con un entero
  - STDIN: 0
  - STDOUT: 1
  - STDERR: 2

SHEL BÁSICO: REDIRECCIONES Y TUBERÍAS

- Los comandos <, > y >> permitan enviar las entradas o salidas de un comando de o a un archivo
  - < conecta la entrada de un archivo con un proceso
  - > envía la salida de un comando a un archivo (reemplaza)
  - >> envía la salida de un comando a un archivo (concatena)
  - &> envía el error y la salida estándar al mismo archivo
  - 2>

SHEL BÁSICO

Variables de entorno

SHEL BÁSICO: VARIABLES DE ENTORNO

- Definición:

```
varName=valor
```

- Uso:

```
$varName
```

- Importante, no poner espacio alrededor del =
- Normas de estilo
  - Mayusculas para variables globales
  - Minúsculas para variables locales

SHEL BÁSICO

Quoting

SHEL BÁSICO: QUOTING

- Las cadenas de caracteres entrecomilladas entre comillas simples o dobles tienen un tratamiento similar
  - Las comillas dobles permiten expansión
  - Las comillas simples invertidas permiten ejecutar el código en su interior y reemplazar el resultado en su lugar

```
echo “Mi correo es $miCorreo”
```

```
echo “Mi correo es ${miCorreo}”
```

```
echo ‘Mi correo es $micorreo’
```

```
echo “Hoy es `date +%a`”
```

COMANDOS DE FILTRADO

COMANDOS DE FILTRADO

- cut: separar una línea en campos
  - -d cambia el delimitador
- sort: ordena las líneas
- uniq: muestra las líneas únicas
  - Suele necesitar que las líneas estén ordenadas
- wc: cuenta líneas, caracteres, y palabras
- head: muestra el comienzo de un fichero
- tail: muestra el final de un fichero
- grep: búsqueda de texto

CONTROL DE ACCESO

CONTROL DE ACCESO

- Reglas generales
  - Los objetos tienen dueños
    - Los dueños tienen acceso amplio a los objetos
  - Los objetos creados por nosotros nos pertenecen
    - El usuario especial root puede acceder a cualquier objeto
    - Sólo el usuario root puede realizar ciertas tareas

CONTROL DE ACCESO

Sistema de ficheros

CONTROL DE ACCESO: SISTEMA DE FICHEROS

- Sistema de ficheros
  - Cada fichero tiene un dueño y un fichero
  - Se puede establecer permisos de acceso a nivel de dueño o ficheros
    - Incluso se puede ser tan restrictivo que el propio usuario no pueda acceder
  - El sistema guarda el usuario y grupo como un número
    - UID
    - GID
  - La traducción entre número e identificación se guardan en:
    - /etc/passwd
    - /etc/group

CONTROL DE ACCESO

Procesos

CONTROL DE ACCESO: PROCESOS

- El dueño de un proceso puede
  - enviar señales al proceso
  - degradarlo

CONTROL DE ACCESO

Root

CONTROL DE ACCESO: ROOT

- El root (superusario) es la cuenta con mayor privilegios de un sistema UNIX
  - UID 0
  - Podemos cambiar el nombre de la cuenta y su UID
    - Mala idea…
- Los sistemas UNIX permiten al root realizar cualquier operación válida sobre cualquier archivo o proceso
- A veces es necesario ejecutar un proceso con un mayor nivel de acceso
  - setuid
  - setgid

CONTROL DE ACCESO: ROOT

- El bit setuid permite a un usuario ejecutar un determinado proceso como si fuera un usuario con mayor nivel, pero con ciertas restricciones
- Ejemplo
  - Un usuario quiere cambiar su contraseña
    - El comando passwd accede al fichero protegido /etc/shadow
    - El comando comprueba quien lo ejecuta, y modifica su comportamiento
      - El usuario sólo podrá cambiar su contraseña
      - El root podrá cambiar todas las contraseñas

CONTROL DE ACCESO

Sistema de acceso actuales

CONTROL DE ACCESO: ACTUALIDAD

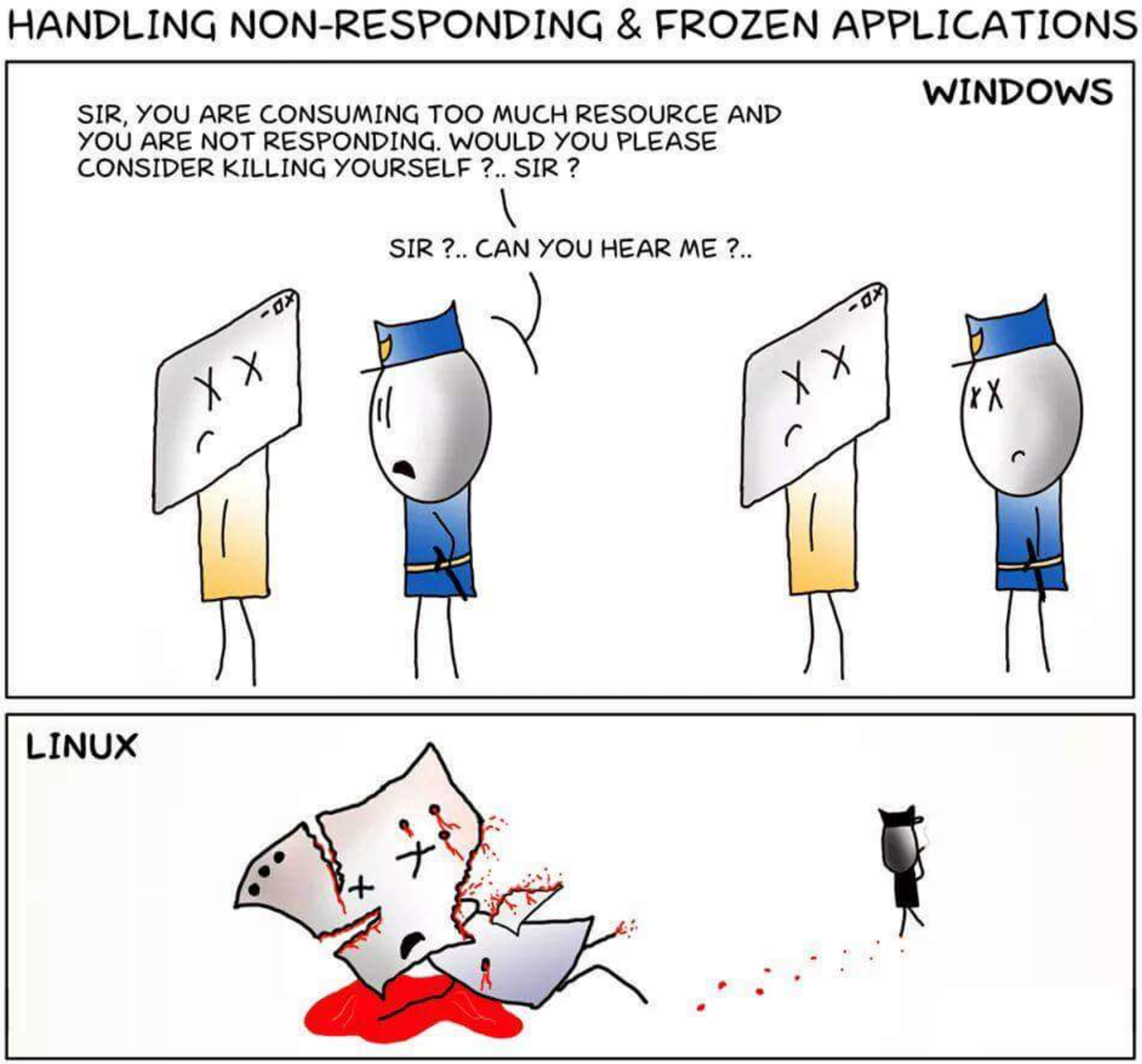
- El sistema tradicional de control de acceso ha perdurado en el tiempo por
  - su simplicidad,
  - previsibilidad y
  - capacidad de cumplir los criterios de control de acceso de la mayoría de los casos
- Pero, tiene algunos inconvenientes
  - El root es un punto único de fallo. Puede comprometer la integridad de todo el sistema
  - La única forma de subdividir los privilegios del root es mediante setuid
  - El modelo de seguridad no es lo suficiente seguro para la Internet
  - Es complicado de auditar

CONTROL DE ACCESO: ACTUALIDAD

- El comando sudo intenta ser simple y seguro
  - Limited su
- El comando sudo toma la línea que y la ejecuta con privilegios de root
  - El usuario tiene que estar en la lista sudoers
    - /etc/sudoers

CONTROL DE PROCESOS

CONTROL DE PROCESOS



CONTROL DE PROCESOS

- Un proceso es una abstracción para representar un programa en ejecución
  - Objeto a través del que el programa hace uso de la memoria, espacio y recursos E/S del sistema
- La mayor parte del trabajo realizado se tiene que realizar a través de procesos en lugar de por el Kernel
-

CONTROL DE PROCESOS

Componentes

CONTROL DE PROCESOS: COMPONENTES

- En general, un proceso consiste en un espacio de nombres y un conjunto de estructuras de datos dentro del kernel
- Componentes principales:
  - El espacio de direcciones del proceso
    - Sistema de memoria virtual
  - El estado del proceso
  - La prioridad de ejecución
  - Recursos utilizados
  - Información sobre los archivos y puertos de red que el proceso ha abierto
  - La máscara de señal del proceso
  - El dueño del proceso

CONTROL DE PROCESOS: COMPONENTES

- PID: process ID number
  - El kernel asigna un ID único a cada proceso
  - Se asignan en el orden en el que los procesos se crean
- PPID: parent ID
  - Ni Linux ni Unix pueden tiene una llamada del sistema para iniciar un nuevo proceso que ejecute un programa particular
    - Se clonan de un padre
  - Todos los procesos en Linux/Unix tienen un proceso padre
    - Excepto el proceso 0

CONTROL DE PROCESOS: COMPONENTES

- UID
  - Número de identificación del usuario que inició el proceso
  - Sólo el creador del proceso y el superusuario pueden manipular el proceso
- EUID
  - Número de identificación del usuario “efectivo”
    - setuid
- GID
  - Número de identificación del grupo al que pertenece el proceso
- EGID
  - Número de identificación del grupo “efectivo”
  - setgid

CONTROL DE PROCESOS

Ciclo de vida

CONTROL DE PROCESOS: CICLO DE VIDA

- Para crear un nuevo proceso, un proceso se copia asimismo mediante la llamada del sistema fork
  - Fork crea una copia idéntica del padre, pero con diferente PID
- fork tiene la única propiedad de devolver dos valores
  - Desde el punto de vista del hijo, devuelve 0
  - El padre recibe el PID del hijo
  - Los procesos tienen que evaluar el valor devuelto para saber su rol
- Tras el fork, el proceso hijo hace uso de exec

CONTROL DE PROCESOS: CICLO DE VIDA

- Cuando el proceso termina, llama a la rutina llamada _exit para notificar al kernel que está listo para morir
  - Devuelve un código de salida: entero
    - 0 indica que todo ha terminado bien
- Antes de que el proceso pueda desaparecer de forma completa, el kernel necesita que el padre conozca la muerte del hijo
  - El padre realiza una llamada a wait
- El padre recibe una copia del código de salida del hijo
- El esquema funciona bien si el padre sobrevive al hijo y realiza de forma correcta la llamada a wait
- Si el padre muere antes, el kernel lo detecta y lo resigna al proceso init

CONTROL DE PROCESOS

Señales

CONTROL DE PROCESOS: SEÑALES

- Las señales son peticiones de interrupción a nivel de procesos
- Existen más de 30 clases de señales predefinidas
- Formas de uso:
  - Pueden enviarse entre los procesos a modo de comunicación
  - Pueden enviarse desde el terminal: <Control-C> y <Control-Z>
  - Pueden enviarse via kill
  - Pueden enviarse desde el kernel si ocurre una violación
  - Pueden enviarse desde el kernel para notificar al proceso de algo que le interese:
    - Muerte de un hijo
    - Disponibilidad de datos en la entrada

CONTROL DE PROCESOS: SEÑALES

- Cuando se recibe la señal existe dos posibles alternativas:
  - Si el proceso es capaz de gestionarla, la gestiona él mismo
    - Capturar la señal
  - Si el proceso no es capaz de gestionarla, el kernel actúa de forma predeterminada
- Los procesos pueden ignorar o bloquearse tras una señal
  - Las señales kill y stop no se pueden ignorar o capturar

CONTROL DE PROCESOS: SEÑALES

- Señales más conocidas

Name	Descripción	Acción por defecto	¿Capturar?	¿Bloquear?	¿Volcado?
HUP	Hangup	Terminate	Si	Si	No
INT	Interrupt	Terminate	Si	Si	No
QUIT	Quit	Terminate	Si	Si	Si
KILL	Kill	Terminate	No	No	No
BUS	Bus error	Terminate	Si	Si	Si
SEGV	Segmentation fault	Terminate	Si	Si	Si
TERM	Software termination	Terminate	Si	Si	No

CONTROL DE PROCESOS: SEÑALES

- ¿KILL, INT, TERM, HUP Y QUIT?
  - KILL: termina el proceso a nivel de kernel
  - INT: enviado desde la terminal cuando se introduce <Control-C>
  - TERM: petición para que el proceso termine completamente
    - Se espera que el proceso limpie su estado y termine
  - HUP: dos interpretaciones
    - Petición de reset
    - Limpieza del proceso asociado a un terminal
      - Enviado de forma automática cuando el terminal se cierra
  - QUIT: similar a TERM pero produce un core dump si no se captura

CONTROL DE PROCESOS

Kill

CONTROL DE PROCESOS: KILL

- Kill es el comando más utilizado para terminar un proceso
- Por defecto envía la señal TERM
- Se puede invocar por los dueños de los procesos o por el usuario root
- Sintaxis

```
kill [-signal] pid
```

- Matar un proceso de forma garantizada

```
kill -9 pid
```

CONTROL DE PROCESOS

Estados

CONTROL DE PROCESOS: ESTADOS

- Los procesos pueden tener cuatro estados diferentes:
  - Runnable: se puede ejecutar
  - Sleeping: esperando a algún recurso
  - Zombie: intentando morir
  - Stopped: suspendido, no se puede ejecutar

CONTROL DE PROCESOS

Monitorización de procesos

CONTROL DE PROCESOS: MONITORIZACIÓN

- ps
  - ps aux
  - ps lax
- top
- htop

CONTROL DE PROCESOS: MONITORIZACIÓN

~/Desktop/VagrantMachines/machine1 — ubuntu@ubuntu-xenial: ~ — ssh ◂ vagrant ssh — 109×36

```
ubuntu@ubuntu-xenial:~$ ps aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1 31.2  0.5   6716  5160 ?        Ss   22:30   0:20 /sbin/init
root          2  0.0  0.0      0     0 ?        S    22:30   0:00 [kthreadd]
root          3  0.0  0.0      0     0 ?        S    22:30   0:00 [ksoftirqd/0]
root          4  0.0  0.0      0     0 ?        S    22:30   0:00 [kworker/0:0]
root          5  0.0  0.0      0     0 ?        S<   22:30   0:00 [kworker/0:0H]
root          6  3.8  0.0      0     0 ?        S    22:30   0:02 [kworker/u4:0]
root          7  0.0  0.0      0     0 ?        S    22:30   0:00 [rcu_sched]
root          8  0.0  0.0      0     0 ?        S    22:30   0:00 [rcu_bh]
root          9  0.0  0.0      0     0 ?        S    22:30   0:00 [migration/0]
root         10  0.0  0.0      0     0 ?        S    22:30   0:00 [watchdog/0]
root         11  0.0  0.0      0     0 ?        S    22:30   0:00 [watchdog/1]
root         12  0.0  0.0      0     0 ?        S    22:30   0:00 [migration/1]
root         13  0.0  0.0      0     0 ?        S    22:30   0:00 [ksoftirqd/1]
root         14  0.0  0.0      0     0 ?        S    22:30   0:00 [kworker/1:0]
root         15  0.0  0.0      0     0 ?        S<   22:30   0:00 [kworker/1:0H]
root         16  0.0  0.0      0     0 ?        S    22:30   0:00 [kdevtmpfs]
root         17  0.0  0.0      0     0 ?        S<   22:30   0:00 [netns]
root         18  0.0  0.0      0     0 ?        S<   22:30   0:00 [perf]
root         19  0.0  0.0      0     0 ?        S    22:30   0:00 [khungtaskd]
root         20  0.0  0.0      0     0 ?        S<   22:30   0:00 [writeback]
root         21  0.0  0.0      0     0 ?        SN   22:30   0:00 [ksmd]
root         22  0.0  0.0      0     0 ?        SN   22:30   0:00 [khugepaged]
root         23  0.0  0.0      0     0 ?        S<   22:30   0:00 [crypto]
root         24  0.0  0.0      0     0 ?        S<   22:30   0:00 [kintegrityd]
root         25  0.0  0.0      0     0 ?        S<   22:30   0:00 [bioset]
root         26  0.0  0.0      0     0 ?        S<   22:30   0:00 [kblockd]
root         27  0.0  0.0      0     0 ?        S<   22:30   0:00 [ata_sff]
root         28  0.0  0.0      0     0 ?        S<   22:30   0:00 [md]
root         29  0.0  0.0      0     0 ?        S<   22:30   0:00 [devfreq_wq]
root         30  1.7  0.0      0     0 ?        S    22:30   0:01 [kworker/u4:1]
root         31  0.0  0.0      0     0 ?        S    22:30   0:00 [kworker/1:1]
root         33  0.0  0.0      0     0 ?        S    22:30   0:00 [kswapd0]
root         34  0.0  0.0      0     0 ?        S<   22:30   0:00 [vmstat]
root         35  0.0  0.0      0     0 ?        S    22:30   0:00 [fsnotify_mark]
```

CONTROL DE PROCESOS: MONITORIZACIÓN

~/Desktop/VagrantMachines/machine1 — ubuntu@ubuntu-xenial: ~ — ssh ◂ vagrant ssh — 109×36

```
top - 22:32:49 up 2 min,  1 user,  load average: 0.08, 0.11, 0.05
Tasks: 110 total,   1 running, 109 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  1023836 total,   859660 free,    38224 used,   125952 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   839192 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0    6716   5160   3828 S   0.0  0.5   0:20.66 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.01 ksoftirqd/0
    4 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S   0.0  0.0   0:02.54 kworker/u4:0
    7 root      20   0       0      0      0 S   0.0  0.0   0:00.05 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/0
   10 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/1
   12 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/1
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.01 ksoftirqd/1
   14 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0
   15 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
   16 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kdevtmpfs
   17 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 netns
   18 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 perf
   19 root      20   0       0      0      0 S   0.0  0.0   0:00.00 khungtaskd
   20 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 writeback
   21 root      25   5       0      0      0 S   0.0  0.0   0:00.00 ksmd
   22 root      39  19       0      0      0 S   0.0  0.0   0:00.00 khugepaged
   23 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 crypto
   24 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kintegrityd
   25 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 bioset
   26 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kblockd
   27 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 ata_sff
   28 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 md
   29 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 devfreq_wq
```

CONTROL DE PROCESOS: MONITORIZACIÓN

~/Desktop/VagrantMachines/machine1 — ubuntu@ubuntu-xenial: ~ — ssh ◂ vagrant ssh — 109×36

```
  1  [                                          0.0%]   Tasks: 29, 21 thr; 1 running
  2  [                                          0.0%]   Load average: 0.04 0.10 0.04
  Mem[|||||||||                          48.4M/1000M]   Uptime: 00:03:21
  Swp[                                        0K/0K]

   PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  1472 ubuntu     20   0  5384  3396  2848 R  0.0  0.3  0:00.02 htop
  1163 root       20   0 30832  2428  2096 S  0.0  0.2  0:00.05 /usr/sbin/VBoxService
  1452 ubuntu     20   0 10960  4592  3724 S  0.0  0.4  0:00.04 sshd: ubuntu@pts/0
     1 root       20   0  6716  5160  3828 S  0.0  0.5  0:20.66 /sbin/init
   409 root       20   0  5744  2492  2224 S  0.0  0.2  0:00.16 /lib/systemd/systemd-journald
   437 root       20   0 13280  1368  1216 S  0.0  0.1  0:00.00 /sbin/lvmetad -f
   466 root       20   0 12024  3648  2852 S  0.0  0.4  0:00.08 /lib/systemd/systemd-udevd
   839 root       20   0  6012   608     0 S  0.0  0.1  0:00.00 /sbin/dhclient -1 -v -pf /run/dhclient.enp0s3.
  1010 root       20   0  2984   112    44 S  0.0  0.0  0:00.00 /sbin/iscsid
  1011 root       10 -10  3444  2860  2008 S  0.0  0.3  0:00.03 /sbin/iscsid
  1023 syslog     20   0 31652  3044  2608 S  0.0  0.3  0:00.00 /usr/sbin/rsyslogd -n
  1024 syslog     20   0 31652  3044  2608 S  0.0  0.3  0:00.00 /usr/sbin/rsyslogd -n
  1025 syslog     20   0 31652  3044  2608 S  0.0  0.3  0:00.00 /usr/sbin/rsyslogd -n
  1018 syslog     20   0 31652  3044  2608 S  0.0  0.3  0:00.02 /usr/sbin/rsyslogd -n
  1031 root       20   0  827M 18884  9160 S  0.0  1.8  0:00.00 /usr/lib/snapd/snapd
  1032 root       20   0  827M 18884  9160 S  0.0  1.8  0:00.00 /usr/lib/snapd/snapd
  1033 root       20   0  827M 18884  9160 S  0.0  1.8  0:00.00 /usr/lib/snapd/snapd
  1040 root       20   0  827M 18884  9160 S  0.0  1.8  0:00.00 /usr/lib/snapd/snapd
  1041 root       20   0  827M 18884  9160 S  0.0  1.8  0:00.00 /usr/lib/snapd/snapd
  1020 root       20   0  827M 18884  9160 S  0.0  1.8  0:00.05 /usr/lib/snapd/snapd
  1035 root       20   0  4148  2996  2704 S  0.0  0.3  0:00.01 /lib/systemd/systemd-logind
  1038 root       20   0  2244  1100  1032 S  0.0  0.1  0:00.00 /usr/sbin/acpid
  1052 daemon     20   0  3480  2096  1932 S  0.0  0.2  0:00.00 /usr/sbin/atd -f
  1054 messagebu  20   0  6052  3664  3292 S  0.0  0.4  0:00.05 /usr/bin/dbus-daemon --system --address=system
  1059 root       20   0 38276  5924  5416 S  0.0  0.6  0:00.00 /usr/lib/accountsservice/accounts-daemon
  1061 root       20   0 38276  5924  5416 S  0.0  0.6  0:00.00 /usr/lib/accountsservice/accounts-daemon
  1057 root       20   0 38276  5924  5416 S  0.0  0.6  0:00.02 /usr/lib/accountsservice/accounts-daemon
  1064 root       20   0  5160  2744  2528 S  0.0  0.3  0:00.00 /usr/sbin/cron -f
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

/PROC

/PROC

- Los comandos ps y top obtienen la información del directorio /proc
- /proc es un pseudo-sistema de ficheros en el que el kernel muestra información interesante sobre el estado del sistema
  - No sólo muestra información de procesos
- Permite leer el estado y escribir en ciertos archivos
- Se subdivide en subdirectorios por PID
  - /proc/1 contiene información de init

/PROC

Archivo	Contenido
cmd	Comandos o programas en ejecución
cmdline	Linea completa del comando del proceso
cwd	Enlace simbólico del directorio actual del proceso
environ	Variables de entorno del proceso
exe	Enlace simbólico del archivo en ejecución
fd	Subdirectorio conteniendo enlaces para cada archivo abierto
maps	Información del mapeado de la memoria
root	Enlace simbólico a los procesos del directorio root
stat	Información general de los procesos
stam	Uso de la memoria

/PROC

cat /proc/cpuinfo

cat /proc/meminfo

cat /proc/loadvg

cat /proc/partitions

cat /proc/version

INSTALACIÓN DE PAQUETES

ADMINISTRACIÓN DE SOFTWARE

- En Linux existen dos formatos de paquetes:
  - .rpm
  - .deb
- Existen programas que ponen una capa superior para facilitar la instalación del paquete y sus dependencias
  - yum
  - apt

ADMINISTRACIÓN DE SOFTWARE

- Todo programa de ordenador se sustenta en otros programas y librerías
  - La base en la que se cimientan es el kernel
  - Los paquetes se pueden reemplazar por otros
    - Puede causar que el programa deje de funcionar
- Las herramientas de administración de paquetes de Linux están pensadas para minimizar problemas

ADMINISTRACIÓN DE SOFTWARE

- El sistema ayuda a evitar problemas de varias formas:
  - Paquetes: colecciones de archivos que se instalan en el ordenador
  - Base de datos de archivos instalados
  - Dependencias: requisitos mutuos de programas
  - Sumas de control: verificación
  - Actualización y desinstalación de paquetes
  - Creación de paquetes binarios

ADMINISTRACIÓN DE SOFTWARE - RPM

- Desarrollado por Red Hat
  - Publicado bajo GPL
  - Usado por Red Hat, Fedora, CentOS, Mandrake, Mandriva, Suse, Open Suse
    - Las distribuciones se han ido separando con el tiempo
      - Variabilidad considerable
- RPM es una herramienta independiente de plataformas
- Los paquetes se llaman de la siguiente forma:

```
nombre_paquete-a.b.c-x.arq.rpm
```

ADMINISTRACIÓN DE SOFTWARE - RPM

- nombre_paquete: Nombre del paquete
- a.b.c: número de versión del programa
- x: número de compilación
- arq: arquitectura
  - i386: x86
  - ppc: PowerPC
  - x86_64: plataformas x86 de 64 bits

ADMINISTRACIÓN DE SOFTWARE – RPM

- ¿Puede cualquier rpm instalarse en cualquier distribución de Linux compatible?
  - No
- Problemas de incompatibilidad
  - Diferentes versiones de RPM
  - Dependencias para un Linux particular
  - Dependencias a nombres de paquetes diferentes
  - Las distribuciones pueden incorporar ligeramente distintos
    - Puede ser necesaria la instalación de paquetes adicionales
  - Script y/o archivos de configuración específicos

ADMINISTRACIÓN DE SOFTWARE

- El comando rpm instala, verifica y consultar el estado de los paquetes en formato .rpm
  - rpm -i: instalar
  - rpm -U: actualizar
  - rpm -e: borrar
  - rpm -q: consultar
- rpm no instala las dependencias
  - Se puede consultar las dependencias mediante:

```
rpm -q —-whatrequeries paquete
```

ADMINISTRACIÓN DE SOFTWARE – DEBIAN

- Similares a RPM en objetivos, pero diferentes características
- Usado en distribuciones basadas en Debian: Ubuntu, Mint, Xandros
- El formato de paquete Debian es neutral al sistema operativo y al tipo de CPU
- Los paquetes se suelen llamar de forma similar a los RPM
  - A veces se omite la arquitectura cuando es x86
  - all significa independiente de la arquitectura

ADMINISTRACIÓN DE SOFTWARE – DEBIAN

- El comando dpkg instala los paquetes .deb
  - dpkg –-install
  - dpkg –-remove
  - dpkg –l

DMINISTRACIÓN DE SOFTWARE – DEBIAN

- Los sistemas basados en Debian suelen usar unas utilidades de nivel superior para gestionar la instalación y eliminación de paquetes
  - apt-get
  - dselect
- Se suelen utilizar cuando se tienen que instalar varios paquetes
- Es preferible usar dpkg cuando se instala un paquete o cuando este se ha descargado previamente

`apt-get [opciones][comando] [nombre-paquetes]`

- Comandos:
  - update: obtiene información actualizada de los paquetes
  - upgrade: actualiza todos los paquetes instalados
  - dselect-Select-upgrade: realiza cualquier cambio que se haya quedado pendiente tras dselect
  - dist-upgrade: similar a update pero con resolución de conflictos inteligente
  - install: instala un paquete
  - remove: elimina un paquete
  - source: obtiene el paquete fuente más reciente
  - check: revisa la consistencia de la base de datos de paquetes y las instalaciones erróneas de paquetes
  - clean: realiza tareas de mantenimiento para hacer una limpieza a fondo de la información de los archivos obtenidos de la base de datos
  - autoclean: borra la información de los paquetes que ya no se pueden descargar

SISTEMA DE ARCHIVOS

SISTEMA DE ARCHIVOS

- En las primeras distribuciones de Linux no seguían los mismos patrones de ficheros
  - Confusión entre distribuciones
  - Cismas dentro de la comunidad
- FSSTDN (File System Standard)
  - La primera versión se publicó en 1994
  - Hacia 1995 se hicieron evidentes las limitaciones de FSSTDN
    - Se creo FHS (File System Hierarchy Standard)

SISTEMA DE ARCHIVOS

- FHS
  - Amplia sustancialmente a FSSTDN
  - Se emplea para definir la estructura de archivos de SO tipo Unix
  - Diferencia entre archivos que se pueden compartir y no se pueden compartir
    - Se pueden compartir: archivos de usuario, archivos de programas.
      - NFS
    - No se pueden compartir
      - Información específica del sistema (archivos de configuración)

SISTEMA DE ARCHIVOS

- Diferencia entre archivos estáticos y dinámicos
  - Estáticos: no cambian a no ser que el administrador haga un cambio sustancial
    - Programas ejecutables
  - Dinámicos / variables: los usuarios pueden cambiarlos
    - Directorios de usuarios, pilas de correo, etc.
- FHS intenta categorizar cada directorio de acuerdo a estos aspectos

	Compartir	No compartir
Estáticos	/usr /opt	/etc /boot
Dinámicos	/home /var/mail	/var/run /var/lock

SISTEMA DE ARCHIVOS

- Directorios importantes
  - /: raíz
    - /etc y /sbin deberían estar dentro de la misma partición principal
  - /boot: archivos estáticos y no compartibles relacionados con el arranque del sistema
    - GRUB y/o LILO
    - Es recomendable guardarlo en una partición diferente
  - /etc: archivos de configuración del sistema estáticos y que no se pueden compartir
    - Archivos de inicio y de configuración de nivel superior
    - Controlan programas y servicios que ofrecen el sistema

SISTEMA DE ARCHIVOS

- /bin: contiene los principales archivos ejecutables (ls, cp, mount)
  - Archivos estáticos que no se pueden compartir
  - Accesibles para todos los usuarios
  - Comandos básicos para usuarios normales
- /sbin: similar a /bin pero con programas que normalmente solo utiliza el administrador
  - Estático y compartible en teoría (no tiene sentido)
- /lib: librerías compartidas de programas
  - /lib/modules: módulos del kernel, controladores, etc.
  - Estático y compatible

SISTEMA DE ARCHIVOS

- /usr: aloja el grueso de los programas de ordenador de un sistema Linux
  - Compartible y estático
  - Se puede montar en una partición diferente en modo lectura
- /usr/local
  - Archivos que instala localmente un administrador
  - Se usa durante las instalaciones
- /usr/share/man: páginas del man
- /usr/X11R6: archivos relacionados con el sistema de ventanas X
  - En las distribuciones modernas se emplea /usr/bin

SISTEMA DE ARCHIVOS

- /opt: similar a /usr/local pero para paquetes prefabricados no incluidos en el SO
  - Procesadores de texto, juegos, etc.
- /home: directorio con los datos de los usuarios
- /root: home del usuario root
- /var: archivos de configuración
  - /var/adm: logs y configuración administrativos
  - /var/log: logs del sistema y programas
  - /var/spool: colas de impresión, correos, tec.
  - /var/tmp: espacio temporal mantenido en los reinicios
- /tmp: para archivos temporales
- /mnt: dispositivos extrapoles (puntos de montaje)

SISTEMA DE ARCHIVOS

- /media: similar a /mount pero para dispositivos de medios específicos
  - CD-ROM
  - DVD
  - USB
- /dev: interfaces hardware
- /proc: sistema de archivos virtual para la comunicación e información de procesos

SISTEMA DE ARCHIVOS

GESTIÓN DE USUARIOS

GESTIÓN DE USUARIOS

- Fichero /etc/passwd
  - Almacena la lista de usuarios reconocidos por el sistema
  - Puede extenderse o reemplazase por un servicio de directorio
  - El sistema los consulta en el login de los usuarios para saber su UID y su home
  - Cada línea representa un usuario

GESTIÓN DE USUARIOS

- Formato del fichero passwd
  - Login name
  - Encrypted password
  - UID
  - Default GID
  - “GECOS” information: full name, office, extension, home phone, etc.
  - Home directory
  - Login shell

GESTIÓN DE USUARIOS

- Almacenar la clave en passwd es un agujero de seguridad

~/Desktop/VagrantMachines/machine1 — ubuntu@ubuntu-xenial: / — ssh ◂ vagrant ssh — 109×36

```
ubuntu@ubuntu-xenial:/$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
ubuntu@ubuntu-xenial:/$
```

GESTIÓN DE USUARIOS

- Fichero /etc/shadow

~/Desktop/VagrantMachines/machine1 — ubuntu@ubuntu-xenial: / — ssh ‹ vagrant ssh — 109×36

```
ubuntu@ubuntu-xenial:/$ sudo cat /etc/shadow
root:*:17381:0:99999:7:::
daemon:*:17381:0:99999:7:::
bin:*:17381:0:99999:7:::
sys:*:17381:0:99999:7:::
sync:*:17381:0:99999:7:::
games:*:17381:0:99999:7:::
man:*:17381:0:99999:7:::
lp:*:17381:0:99999:7:::
mail:*:17381:0:99999:7:::
news:*:17381:0:99999:7:::
uucp:*:17381:0:99999:7:::
proxy:*:17381:0:99999:7:::
www-data:*:17381:0:99999:7:::
backup:*:17381:0:99999:7:::
list:*:17381:0:99999:7:::
irc:*:17381:0:99999:7:::
gnats:*:17381:0:99999:7:::
nobody:*:17381:0:99999:7:::
systemd-timesync:*:17381:0:99999:7:::
systemd-network:*:17381:0:99999:7:::
systemd-resolve:*:17381:0:99999:7:::
systemd-bus-proxy:*:17381:0:99999:7:::
syslog:*:17381:0:99999:7:::
_apt:*:17381:0:99999:7:::
lxd:*:17381:0:99999:7:::
messagebus:*:17381:0:99999:7:::
uuidd:*:17381:0:99999:7:::
dnsmasq:*:17381:0:99999:7:::
sshd:*:17381:0:99999:7:::
pollinate:*:17381:0:99999:7:::
ubuntu:$6$mU6Ik2iO$7bJcPa/jHHyqK9.aLnEjsBFB92WK.8bKajSvdQ7C0vYUE.rvObBXIcDb0Zwe1QB.b5HrvVj9eYlJJRZonM6cP1:175
97:0:99999:7:::
ubuntu@ubuntu-xenial:/$
```

GESTIÓN DE USUARIOS

- La gestión de usuarios se realiza mediante el comando useradd

PROCESOS PERIÓDICOS

PROCESOS PERIÓDICOS

- La automatización es la clave de la administración de servidores
- Casi cualquier tare puede ser codificada en un script bash, Perl o Python
- Ejemplo:
  - Verificar cada media hora que los router y switch están bien

PROCESOS PERIÓDICOS

- El demonio cron es el modo estándar de ejecutar comandos con una planificación determinada
- Comienza en el arranque del sistema y dura hasta que el sistema se apague
- cron lee un fichero de configuración que contiene la lista de comandos a ejecutar cuando se llame al demonio
- La línea de comandos se ejecutaran por sh
  - Cualquier cosa que se pueda ejecutar en un terminal, se podrá ejecutar en un cron

PROCESOS PERIÓDICOS

- El fichero de configuración de cron se llama crontab
  - /var/spool/cron
- Hay tantos crontabs como usuarios
  - Se llaman como el usuario
- cron suele trabajar en modo silencioso, pero también puede enviar información al log
  - /var/cron/log

PROCESOS PERIÓDICOS

- Formato del archivo crontab

```
minute hour dom month weekday command
```

Campo	Rango
Minuto	0 a 59
Hora	0 a 23
Día del mes	1 a 31
Mes	1 a 12
Día de la semana	0 a 6

PROCESOS PERIÓDICOS

- Cada campo relacionado con el tiempo, puede contener:
  - Un * indica cualquier cosa
  - Un entero, que indica el valor exacto
  - Dos enteros separados por -, indican un rango
  - Lista de valores separados por ,

```
45 10 * * 1-5
```

- Nota: los campos se tratan como OR y no como AND
  - 30 * 13 * 5 ==> cada media hora de los viernes y cada media hora del 13

```
crontab -e
```

```
crontab -l
```

```
crontab -r
```

```
crontab -u
```

BASH SCRIPTING