

Laboratorio di Sistemi Operativi

Amministrazione

Renzo Davoli
Alberto Montresor

Copyright © 2004-2005 Renzo Davoli, Alberto Montresor

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found at: <http://www.gnu.org/licenses/fdl.html#TOC1>

Introduzione

▪ **Obiettivo**

- Breve introduzione ad alcuni tool e meccanismi disponibili per l'amministrazione

▪ **Sommario**

- Esecuzione periodica dei processi: cron
- Autorizzazione più fine: sudo
- File systems: mount & C.
- File di log
- Init

Cron

- **Schedulazione automatica di processi periodici**
- **Il daemon cron**
 - avviato al boot della macchina
 - legge il contenuto di uno o più file di configurazione
 - elenchi di comandi, script, etc.
 - orari in cui queste comandi vengono eseguiti
 - questi file vengono detti crontab (da cron table)
 - ogni utente ha un suo crontab
 - chiamato con il nome dell'utente
 - posizionato in /var/spool/cron/tabs
 - altri file/directory che vengono utilizzati
 - /etc/crontab, /etc/cron.d, /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly

Formato dei file crontab

- **Un insieme di linee del tipo**
 - *minuto ora giorno mese giorno_settimana comando*
- **Esempi di valori**
 - *: qualunque valore
 - un solo intero: valore preciso
 - due interi separati da un linea: range
 - possono includere "steps" separati da /
 - lista di interi separati da virgole: più valori
- **Esempi**
0,30 * 13 * 5 (cd /home/joe/project ; make)
0 9-15/2 * * 1-5 write joe % Lavora!
10 17 * * 1-5 write joe % Sei ancora qui?

Come interagire con cron

- **Modificando il crontab personale o di un utente**

```
crontab [ -u user ] file
```

```
crontab [ -u user ] { -l | -r | -e }
```

corrispondono a: update file, list, remove, edit

- **Abilitando/disabilitando un particolare insieme di utenti**

- /var/spool/cron/allow
 - se esiste, il suo contenuto determina chi può eseguire crontab
- /var/spool/cron/deny
 - se non esiste, il suo contenuto determina chi non può crontab
- PS: locazione precisa dipende dal sistema

Esempi di cron

- **Comandi che può essere utile eseguire periodicamente:**

- distribuzione remota di file di configurazione e software (rdist)
- rotazione dei file di log
- backup

- **find / -xdev -name core -atime +7 -exec rm -f {} ';'**

SUDO

▪ Problema:

- il meccanismo di root su UNIX dà "poteri" assoluti, ma....
- a volte è necessario concedere a singoli utenti alcuni dei poteri di root ma non tutti (es. backup)

▪ Soluzione: sudo

- Permette di specificare chi può eseguire un certo comando in base a:
 - host in cui viene eseguito
 - utente che richiede l'esecuzione
 - tipo di comando specificato

▪ Sintassi: `sudo comando`

SUDO

▪ Come funziona?

- richiede come argomento una linea di comando da eseguire in qualità di root (o di un altro utente)
- consulta il file di configurazione `/etc/sudoers`
- se il comando è permesso
 - chiede la password *dell'utente*
 - esegue il comando

▪ Note:

- una volta inserita la password, successivi comandi sudo entro 5 minuti (configurabili) non la chiederanno
- sudo memorizza nei file di log tutto quello che viene fatto
-

Esempio di /etc/sudoers

```
# User alias specification
User_Alias    FULLTIMERS = millert, mikef, dowdy
User_Alias    PARTTIMERS = bostley, jwfox, crawl
User_Alias    WEBMASTERS = will, wendy, wim

# Host alias specification
Host_Alias    SPARC = bigtime, eclipse, moet, anchor
Host_Alias    CUNETS = 128.138.0.0/255.255.0.0
Host_Alias    CSNETS = 128.138.243.0, 128.138.204.0/24

# Cmnd alias specification
Cmnd_Alias    DUMPS = /usr/sbin/dump, /usr/sbin/restore
Cmnd_Alias    KILL = /usr/bin/kill
Cmnd_Alias    SHUTDOWN = /usr/sbin/shutdown
Cmnd_Alias    HALT = /usr/sbin/halt, /usr/sbin/fasthalt

FULLTIMERS    ALL = NOPASSWD: ALL
PARTTIMERS    ALL = ALL
operator      ALL = DUMPS, KILL, SHUTDOWN, HALT, /usr/oper/bin/
%users        ALL= NOPASSWD: /usr/local/bin/startmodem
```

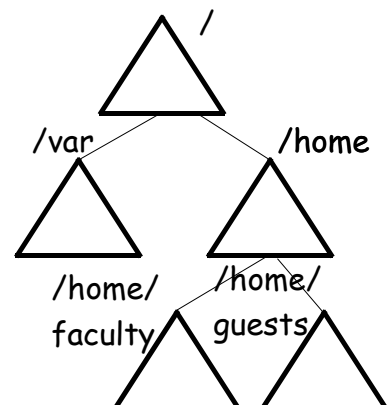
© 2004 Renzo Davoli, Alberto Montresor

9

Filesystem

■ Si utilizza il termine file system

- per indicare un insieme di file e directory appartenenti ad un'unica partizione (*)
- per indicare la composizione in unico albero di file di più file system



■ Per "montare" un file system in un altro:

- `mount radice-filesystem directory-destinazione`

■ Esempio:

- `mount /dev/sda1 /usbkey`

Una directory esistente
Il contenuto diventa
"invisibile"

© 2004 Renzo Davoli, Alberto Montresor

(*) E non solo: NFS, /proc, etc

10

Filesystem

- Per "smontare" un filesystem:

- `umount directory-destinazione`

- Cosa succede se un file nel file system e' in uso?

- `umount: /home: device is busy`

- Chi utilizza un certo file file system? fuser o lsof

```
fuser -mv /home
```

	USER	PID	ACCESS	COMMAND
/home/	montreso	1641	f.c..	kde
	montreso	1681	f.c..	kdeinit
	montreso	1684	f.c..	kdeinit
	montreso	1687	f.c..	kdeinit

Filesystem

- **fstab (estratto da dotto.cs.unibo.it)**

```
# <file system><mount point><type><options><dump><pass>
/dev/hda1 / ext2 errors=remount-ro 0 1
/dev/hda8 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/hda5 /tmp ext2 rw 0 2
/dev/hda6 /var ext2 rw 0 2
/dev/hda7 /usr ext2 rw 0 2
d2:/home/mail /var/spool/mail nfs bg,nodev,nosuid,hard,intr 0 0
d2:/home/faculty /home/faculty nfs bg,nodev,nosuid,hard,intr 0 0
d1:/home/students /home/students nfs bg,nodev,nosuid,hard,intr 0
0
d1:/home/guests /home/guests nfs bg,nodev,nosuid,hard,intr 0 0
d2pub:/space2/public /public nfs bg,nodev,nosuid,hard,intr 0 0
golem:/var/www/people /home/web nfs bg,nodev,nosuid,hard,intr 0
0
```

Filesystem

▪ **fstab options**

- `sync / async` I/O fatto in modo (a)sincrono
- `auto / noauto` montato automaticamente (o no)
- `dev / nodev` chiunque può montare fs (o no)
- `exec / noexec` permette/previene esecuzione di binari
- `suid / nosuid` permette/previene utilizzo di suid-sgid
- `ro / rw` read-only, read-write
- `user / nouser` chiunque può montare fs
- `defaults` rw,suid,dev,exec,auto,nouser,async.

▪ **Più altre ancora, dipendenti dal file system**

- Esempi: `man nfs`

Filesystem

▪ **Automount**

- Meccanismo che monta in modo automatico supporti rimovibili, share remote, etc.

▪ **Due tipi di automount**

- Automount
 - Implementato in user-space
 - Fornisce un file system virtuale basato su NFS
 - Non più usato (per via del passaggio tramite NFS)
- Autofs + Automount
 - Autofs implementato a livello kernel (da 2.2 in poi)
 - Quando rileva una richiesta in un file system, invoca automount

Filesystem

▪ Automount - Esempio

- `/etc/auto.master`
contiene l'indicazione del mount point "radice" per tutti i file system auto-mounted, l'indicazione di un file di mappa, e quale file di configurazione utilizzare

```
#Mount point      Map              Options
/auto             /etc/auto.misc  --timeout=60
```

- `/etc/auto.misc`
contiene il mapping fra file system da montare e device

```
cd                -fstype=iso9660,ro    :/dev/cdrom
floppy            -fstype=vfat          :/dev/fd0
server            -fstype=auto
server.local:/export
```

- Per fare ripartire autofs:
`/etc/init.d/autofs restart`

Filesystem /proc

▪ Cosa contiene il file system proc?

- Una directory `<pid>` per ogni processo
 - `cmdline` `cwd` `environ` `exe` `fd` `mapped_base` `maps` `mem` `mounts` `root` `stat` `statm` `status`
- Varie informazioni relative al sistema
 - `acpi`, `asound`, `buddyinfo`, `bus`, `cmdline`, `cpufreq`, `cpuinfo`, `devices`, `dma`, `driver`, `execdomains`, `fb`, `filesystems`, `fs`, `ide`, `interrupts`, `iomem`, `ioports`, `irq`, `kcore`, `kmsg`, `ksyms`, `loadavg`, `locks`, `lvm`, `mdstat`, `meminfo`, `misc`, `modules`, `mounts`, `mtrr`, `net`, `partitions`, `pci`, `scsi`, `self`, `slabinfo`, `splash`, `stat`, `swaps`, `sys`, `sysrq-trigger`, `sysvipc`, `tty`, `uptime`, `version`, `video`, `processor`

▪ Alcuni esempi...

Filesystem /proc

- **Tramite il filesystem /proc è possibile scoprire**
 - processi il cui eseguibile è stato cancellato
 - ottenere l'eseguibile in esecuzione
- **Come**
 - `file /proc/[0-9]*/exe | grep '(deleted)'`
`/proc/xxx/exe: broken symbolic link to /yyy/www`
 - `cp /proc/xxx/exe orig-copy`
funziona ancora!

File system /proc

- **Come ottenere informazioni in modo più "umano":**
 - `procinfo`
 - informazioni su cpu, memoria e altro
 - prende informazioni da processor, uptime, modules, meminfo, driver, devices
 - `lsdev`
 - mostra indirizzi relativi all'hardware installato
 - prende informazioni da interrupts, ioports e dma'
 - `socklist`
 - mostra le attuali connessioni in corso
 - prende informazioni da tcp, udp, net/raw, [0-9]*/fd/*
 - `lspci`
 - ritorna informazioni sul bus pci, prese da pci

Una parentesi: gestione della priorità

- **Indica quanto è importante un processo**
 - più basso il valore, più alta la priorità
 - valori in Linux: da -20 a +19
 - i processi ereditano i valori di priorità del genitore
- **nice command**
 - lancia un processo con diversa priorità
- **renice priority -p pid**
 - cambia la priorità di un processo in esecuzione

File di log

- **Syslog**
 - Meccanismo centralizzato per gestire i log di tutti i processi
 - semplifica la vita ai programmatori
 - sposta la gestione verso gli amministratori di sistema
- **Caratteristiche principali**
 - Gestione eventi in base alla loro severità e provenienza
 - Possibilità di inviare gli eventi a vari tipi di destinazioni
- **Tre componenti**
 - **syslogd** demone
 - **openlog** libreria di programmazione
 - **logger** comando per shell-scripting

File di log

▪ Syslogd

- Riceve log entry da `openlog` e `logger`
- In base ai file di configurazione, smista le log entry nelle opportune destinazioni

▪ Gestione segnali

- segnale `HUP`: riavvio del processo e riletture file config
- segnale `TERM`: terminazione
- Process id di `syslogd`:
 - Contenuto in `/var/run/syslogd.pid`
 - `kill -HUP '/bin/cat /var/run/syslogd.pid'`

File di log

▪ Configurazione `syslogd`: `/etc/syslog.conf`

- Insieme di linee della forma: `selector action`
- Selettori: `facility.level`

▪ Esempio:

```
kern.warn;*.err;authpriv.*      | /dev/xconsole
*.emerg                          *
mail.*                            -/var/log/mail
mail.=info                       -/var/log/mail.info
mail.warn                        no sync → -/var/log/mail.warn
mail.err                          /var/log/mail.err
```

File di log

▪ Facility

- kernel
- user
- mail
- daemon (demoni)
- auth (autorizzazione)
- lpr
- news
- cron
- mark (marking periodico)
- local0-7
- syslog
- authpriv
- ftp
- * (tutti tranne mark)

▪ Livelli di gravità

- emerg
- alert
- crit
- err
- warning
- notice
- info
- debug

▪ Destinazioni

- *filename*
- *@hostname, @ip-addr*
- *utentel, utente2*
- * (tutti i terminali)

File di log

▪ Politiche di logging

- non loggare nulla
- azzerare i file di log a intervalli periodici
- ruotare i file di log, mantenendo i dati per un periodo prefissato
- comprimere e archiviare i log su file

▪ `logrotate config-file`

- E' in grado di gestire la rotazione, compressione, rimozione, e spedizione di log file
- Periodicamente: giornalmente, settimanalmente, mensilmente o quando cresce troppo
- Da combinare con `cron`

File di log

▪ Esempio di config file per rotateconfig

```
compress
/var/log/messages {
    rotate 5
    weekly
    postrotate
        /sbin/killall -HUP syslogd
    endscript
}
"/var/log/httpd/access.log" /var/log/httpd/error.log {
    rotate 5
    mail www@my.org
    size=100k
    sharedscripts
    postrotate
        /sbin/killall -HUP httpd
    endscript
}
```

Init

- **INIT: il meccanismo di gestione di servizi**
 - Responsabile per l'esecuzione (start/stop) dei processi che forniscono servizi per conto del sistema operativo
 - Esempi di servizi: sshd, lpd, etc.
- **Bootstrap del sistema operativo**
 - Caricamento di un bootloader in posizione fissa
 - Caricamento del s.o. da parte del bootloader stesso
 - Caricamento del kernel
 - Caricamento dei driver per le varie periferiche
 - Il file system di root viene montato
 - Viene fatto partire il primo processo user-level: init (`/sbin/init`)

Init

- **Per maggiori informazioni sul processo di boot**
 - The Linux System Administrator's Guide
<http://tldp.org/LDP/sag/html/index.html>
 - Capitolo 8
Boots And Shutdowns
 - Capitolo 9
Init
 - Il manuale in linea (man) per le seguenti voci:
 - init, inittab, runlevel
 - Ulteriori comandi a cui dare un'occhiata
 - shutdown, getty

Init

- **Alcune caratteristiche di init**
 - Init non può essere ucciso, nemmeno con sigkill
 - L'init di Linux segue il comportamento di UNIX System V
 - Init "adotta" i processi orfani...

Init

- **Runlevel**
 - Un numero che Linux/UNIX SV usa per distinguere differenti configurazioni in cui la macchina può fare boot
- **Runlevel esistenti**
 - 0 Halt the system
 - 1 Single user mode (utilizzato per amministrazione)
 - 2 Multi-user mode with networking disabled
 - 3 Multi-user mode with networking enabled
 - 5 Multi-user mode with networking enabled, X-Windows
 - 6 Reboot
- **0,1,6 sono standard, 2-5 sono dipendenti dalla distribuzione (ma normalmente molto simili)**

Init

- **Per conoscere il runlevel precedente e attuale**
 - `% /sbin/runlevel`
 - `N 5`
 - dove N è quello precedente (nessuno) e 5 è quello attuale
- **Per cambiare il runlevel attuale:**
 - `% telinit <runlevel>`

File /etc/inittab

```
# The default runlevel is defined here
id:5:initdefault:
# First script to be executed
si::bootwait:/etc/init.d/boot
# /etc/init.d/rc takes care of runlevel handling
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6
# getty-programs for the normal runlevels
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
```

File /etc/inittab

- **Ogni riga è composta in questo modo:**

id:runlevels:action:process

- *id*
 - E' un identificatore di linea univoco
 - Per le linee getty, specifica il numero di terminale a cui si riferisce
- *runlevels*
 - E' l'insieme di runlevel in cui la linea deve essere considerata
 - Una semplice lista di cifre
- *action*
 - Specifica come deve essere eseguito il processo
- *process*
 - Il processo da eseguire

File /etc/inittab

- **Alcune possibili azioni:**
 - *respawn*
 - Il processo viene fatto ripartire tutte le volte che termina (es. getty)
 - *wait*
 - il processo parte una volta sola all'attivazione del runlevel specificato
 - init attende la sua terminazione
 - *once*
 - il processo parte una volta sola all'attivazione del runlevel specificato
- *boot*
 - il processo è eseguito al boot del sistema; runlevel è ignorato
- *bootwait*
 - il processo è eseguito al boot del sistema; runlevel è ignorato
 - init attende la sua terminazione
- *initdefault*
 - specifica il runlevel di default; se assente, il valore viene richiesto al boot

rc

- **rc: lo script che viene eseguito subito dopo gli script di boot ed è responsabile per l'attivazione dei servizi**
- **Prende in input il nuovo runlevel**
 - disattiva i servizi del runlevel precedente
 - attiva i servizi del runlevel attuale

```
# /etc/init.d/rc takes care of runlevel handling
```

```
10:0:wait:/etc/init.d/rc 0
```

```
11:1:wait:/etc/init.d/rc 1
```

```
12:2:wait:/etc/init.d/rc 2
```

```
13:3:wait:/etc/init.d/rc 3
```

```
15:5:wait:/etc/init.d/rc 5
```

```
16:6:wait:/etc/init.d/rc 6
```

rc

- **Dove si trovano i servizi?**

- `ls /etc/init.d`
`cron network lpd nfs sshd`
- Sono script che permettono l'attivazione / disattivazione dei servizi

- **Come fa rc a sapere quali servizi attivare in un particolare runlevel?**

- Esistono delle sottodirectory, una per runlevel, che contengono dei link agli script contenuti in `/etc/init.d`
- rc si limita ad eseguire tutti gli script contenuti nella sottodirectory del runlevel corrispondente

rc

`/etc/init.d`

```
rc
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
cron
network
lpd
nfs
sshd
```

`/etc/init.d/rc.5`

```
K10cron
K20network
K30lpd
K40nfs
K41sshd
S10sshd
S11nfs
S20lpd
S30network
S40cron
```

..... link
—— directory

rc

- **Formato dei nomi dei link**
 - Formato: [SK][0-9]{2}[a-zA-Z]+
- **Significato**
 - S per far partire (start) il servizio
 - K per stoppare il servizio
 - xx (numero a due cifre): per indicare l'ordine in cui attivare/disattivare il servizio
 - Parte rimanente: nome dello script