

Dentro la Shell

Lezione 3

Lorenzo Losa

11 aprile 2012

Richiami sui processi

Ricordiamo che...

I programmi in esecuzione sono divisi in *processi*, ognuno dei quali

- ▶ ha un *PID* (Process IDentifier),
- ▶ un utente *proprietario*,
- ▶ e un processo *padre* (tranne *init*).
- ▶ Può avere dei *figli*
- ▶ ed essere formato da più *thread*.

Per vedere i processi in esecuzione: *ps*, *pstree*, *top* e *htop*.

Jobs

Linux è multitasking! I processi possono essere eseguiti sullo sfondo (in background):

```
$ okular &  
[1] 5155
```

Okular è avviato con PID 5155 come *job* 1; per vedere i job attivi:

```
$ jobs  
[1]+  In esecuzione          okular &
```

Un job può essere ucciso:

```
$ kill %1  
$  
[1]+  Terminato             okular
```

fg e bg

Se il processo è già avviato? si può sospendere con `^Z` (`ctrl+z`):

```
$ okular
```

```
^Z
```

```
[1]+  Fermato
```

```
okular
```

Però così è *fermato*, non *in esecuzione*. Ma si può dire di eseguirlo sullo sfondo:

```
$ bg %1
```

```
[1]+  okular &
```

Oppure in primo piano (in foreground):

```
$ fg %1
```

```
okular
```

Far sopravvivere i processi

Se volessi scaricare un file con `wget` ma sloggandomi...

```
$ wget http://dominio.it/dvd.iso &> /dev/null &  
[1] 5711  
$ exit
```

Non funziona! Quando esco il download si ferma. Soluzione:

```
$ nohup wget http://dominio.it/dvd.iso &  
[1] 5733  
$ exit
```

E se me ne dimentico?

```
$ wget http://dominio.it/dvd.iso &> /dev/null  
^Z  
[1]+  Fermato          wget http://dominio.it/dvd...  
$ bg %1  
[1]+  wget http://dominio.it/dvd.iso &>/dev/null &  
$ disown %1  
$ exit
```

screen

La soluzione a molti problemi: `screen`.

`screen` permette di:

- ▶ aprire più sessioni di terminale in un'unica sessione di login;
- ▶ chiudere e riaprire il terminale senza interrompere l'esecuzione dei processi.

Comandi:

`C-a "` lista delle sessioni aperte

`C-a c` crea una nuova sessione (**c**reate)

`C-a n` passa alla sessione successiva (**n**ext)

`C-a p` passa alla sessione precedente (**p**revious)

`C-a 0` passa alla sessione 0

...

`C-a 9` passa alla sessione 9

`C-a d` "stacca" screen, mantenendolo in esecuzione
(**d**etach)

I pacchetti

Ricordiamo che...

Spesso sui sistemi Linux il software è distribuito in *pacchetti* che:

- ▶ possono contenere: programmi, ma anche librerie, documentazione, traduzioni, ecc.;
- ▶ possono *dipendere* da altri pacchetti;
- ▶ contengono sia file che le istruzioni su come trattarli.

Vari formati: `.deb` (Debian), `.rpm` (Red Hat), e altri.

I pacchetti Debian

Su Debian e derivate (come le *ubuntu) i pacchetti possono essere utilizzati su due livelli:

- ▶ a basso livello: gestirli a mano con `dpkg`;
- ▶ ad alto livello: gestirli in automatico con `apt`.

...oppure con le tante altre interfacce (anche grafiche).

dpkg – installare e disinstallare

- Installare un pacchetto:

```
dpkg -i pacchetto.deb  
dpkg --install pacchetto.deb
```

- Rimuovere un pacchetto (file di configurazione esclusi):

```
dpkg -r nomepacchetto  
dpkg --remove nomepacchetto
```

- Rimuovere un pacchetto (file di configurazione inclusi):

```
dpkg -P nomepacchetto  
dpkg --purge nomepacchetto
```

dpkg – cercare e riconfigurare

- ▶ Elencare i file di un pacchetto:

```
dpkg -L nomepacchetto  
dpkg --getfiles nomepacchetto
```

- ▶ Cercare file nei pacchetti:

```
dpkg -S ricerca  
dpkg --get-architecture ricerca
```

- ▶ Riconfigurare un pacchetto:

```
dpkg-reconfigure nomepacchetto
```

APT: che i pacchetti se li scarichi da solo!

APT usa la struttura di `dpkg` per:

- ▶ ottenere da solo i pacchetti richiesti;
- ▶ risolvere da solo le dipendenze, ottendendo i pacchetti necessari;
- ▶ tenere il sistema aggiornato.

APT deve essere istruito su dove cercare i pacchetti; nel caso più semplice è scritto in `/etc/apt/sources.list` come:

```
deb http://ftp.debian.org/debian/ stable main
deb-src http://ftp.debian.org/debian/ stable main
```

apt-get – installare e disinstallare

- ▶ Installare un pacchetto:

```
apt-get install nomepacchetto
```

- ▶ Rimuovere un pacchetto (configurazioni escluse):

```
apt-get remove nomepacchetto
```

- ▶ Rimuovere un pacchetto (configurazioni comprese):

```
apt-get remove --purge nomepacchetto
```

- ▶ Rimuovere i pacchetti “che non servono più” (installati come dipendenze di pacchetti che sono stati rimossi):

```
apt-get autoremove
```

apt-get – aggiornare

- ▶ Aggiornare le informazioni sui pacchetti (da fare sempre prima di un aggiornamento e dopo aver modificato `sources.list`):

```
apt-get update
```

- ▶ Aggiornare tutti i pacchetti (a meno che non abbiano nuove dipendenze):

```
apt-get upgrade
```

- ▶ Aggiornare davvero tutti i pacchetti:

```
apt-get dist-upgrade
```

- ▶ Cancellare i pacchetti scaricati:

```
apt-get clean
```

apt-get – Super Mucca

- ▶ APT ha i poteri della Super Mucca:

```
$ apt-get moo
```

```
  ( __ )
```

```
  ( oo )
```

```
 /-----\
```

```
/ |      ||
```

```
* /\---/\
```

```
  ~ ~    ~ ~
```

```
...."Have you mooed today?"...
```

Informazioni sui pacchetti

Online: `packages.debian.org` o `packages.ubuntu.com`.

Da shell:

- ▶ informazioni su un pacchetto:

```
apt-cache show nomepacchetto
```

- ▶ ricerca:

```
apt-cache search ricerca
```

- ▶ pacchetti installati:

```
dpkg -l [ricerca]
```

Esercitazione sui pacchetti

- 1 scaricate il pacchetto *debian-faq-it* da <http://packages.ubuntu.com/oneiric/all/debian-faq-it/download> e installatelo;
- 2 sapete dire a cosa serve questo pacchetto? (si intuisce dal nome, ma facciamo finta che non sia così)
- 3 come fate a visualizzarne i contenuti?
- 4 probabilmente *debian-faq-it* non vi serve a nulla: se volete potete disinstallarlo;
- 5 installate il pacchetto *openssh-server* (è sempre utile!).

Connettersi da remoto

Ci si può connettere ad un sistema anche da remoto:

- ▶ con un'interfaccia testuale: `ssh`
- ▶ con un'interfaccia grafica: direttamente via X, con VNC, ...

SSH

Con SSH (Secure SHell) ci si può connettere ad una macchina remota:

```
ssh [UTENTE@]MACCHINA
```

Esempio:

```
$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 6f:69:a7:cf:cb:99:79:ee:4e:8d:b6:2d:d2:e0:cc:90.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
utente@localhost's password:
Welcome to Ubuntu 11.10 (GNU/Linux 3.0.0-16-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Thu Mar 15 18:53:33 2012
$
```

SCP

SCP, appoggiandosi a SSH, permette di copiare file fra macchine diverse:

```
scp [-r] origine destinazione
```

Dove origine e destinazione sono nella forma:

```
[[utente@]macchina:]/directory/file
```

Example (copia con scp)

```
echo "Prova" > /tmp/prova  
scp /tmp/prova localhost:/tmp/copia  
cat /tmp/copia
```

Grafica

SSH è ottimo per un accesso testuale: e per le interfacce grafiche?

- ▶ `ssh -X`: così permette di lanciare programmi grafici sul server, con output sul client:

```
$ ssh -X localhost  
utente@localhost's password:  
$ okular
```

- ▶ VNC.

Fine.

Domande?