

Gli strumenti di sviluppo liberi

di

Rodolfo Giometti

giometti@linux.it

(f|ht)tp://ftp.enneenne.com/docs/talks/developing.p(s|df)

Editor

All'interno di un sistema UNIX possono esistere diversi tipi di editor ma quelli senz'altro più potenti sono `vi` ed `emacs`.

Possono risultare di non banale utilizzo agli inesperti ma, una volta che si è entrati nella logica di funzionamento, qualsiasi altro editor risulterà «non all'altezza».

Help

Gli strumenti di aiuto più comuni in un sistema Linux sono `man` ed `info`.

Il comando `man` è uno standard dei sistemi UNIX.

```
rg@zaigor:~$ man printf
```

Il comando `info` invece è una «estensione» che se utilizzata dà un notevole contributo alla documentazione.

```
rg@zaigor:~$ info libc
```

Tramite un linguaggio chiamato *texinfo* è possibile generare uno stesso documento in diversi formati. È possibile quindi fornire all'utente un manuale stile `man`, oppure in formato stampabile (PS o PDF) o anche in formato HTML per la pubblicazione in rete.

Programmi di manipolazione file

Poiché all'interno di un sistema UNIX tutto è visto come un file questi programmi di utilità rivestono un ruolo molto importante.

```
rg@zaigor:~$ cat /dev/hdb > /dev/hdc
```

```
rg@zaigor:~$ dd if=/dev/fd0 of=image bs=512 count=1
```

```
rg@zaigor:~$ tail -f /var/log/kernel.log
```

```
rg@zaigor:~$ od -v -tx1 -Ax < /dev/iocard
```

```
rg@zaigor:~$ cat /dev/framegrabber > mypic.jpeg
```

Questi comandi possono anche essere raggruppati per generare una operazione più complessa.

```
rg@zaigor:~$ dd if=/dev/framegrabber skip=1 bs=4K  
count=128 | gzip -9 | uuencode mypic.jpeg | mail -s "See my  
pic" mom@home.org
```

Programmazione

Esiste la possibilità di usare diversi linguaggi di programmazione all'interno di un sistema Linux. Ogni linguaggio ha speciali caratteristiche che si adattano più o meno bene ad un dato problema. Vi sono:

- Il compilatore C `gcc` (che si compone di `ccp`, `cc1` e `ld`).
- L'assemblatore `as`.
- L'interprete `bash` (da usare come shell - ne esistono altre).
- L'interprete TCL/TK.
- L'interprete PHP.
- ecc..

Supporto allo sviluppo

Un progetto complesso, di solito, è composto da più file sorgente, da una serie di file che definiscono le procedure di compilazione/installazione, una documentazione, ecc..

Per gestire la fase di compilazione/installazione possiamo utilizzare l'applicazione `make` che, dato un file di «istruzioni» riesce ad eseguire tutti i passi nella giusta sequenza per ottenere in programma (o i programmi) eseguibile finale evitando di ripetere operazioni non necessarie.

Per gestire invece un progetto nella sua globalità possiamo utilizzare l'applicazione `cvcs` che permette di gestire non solo il «versioning» di un progetto ma anche il lavoro di gruppo sullo stesso tramite una serie di accessi controllati ad uno stesso «repository».

Il debugging

La fase di debugging è una fase molto importante quando stiamo sviluppando un progetto ed il sistema Linux ci mette a disposizione una serie di programmi di utilità che ci permettono, sia di individuare possibili errori, sia di eseguire una serie di test sulla nostra applicazione.

Per capire «cosa» sta facendo una certa applicazione possiamo usare `gdb` o `strace`.

```
rg@zaigor:~$ echo test | strace -eopen,read,write cat
...
open("/lib/libc.so.6", O_RDONLY)          = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\204\221"... , 1024) = 1024
...
read(0, "test\n", 4096)                  = 5
write(1, "test\n", 5test
)                                          = 5
read(0, "", 4096)                        = 0
```

Il debugging - 2

Per capire cosa sta «passando» sulla rete possiamo usare `tcpdump` e `netstat`, mentre per testare le funzionalità di un server possiamo usare `telnet`.

```
rg@zaigor:~$ netstat -ln | grep '^tcp.*25'
tcp      0      0 0.0.0.0:25          0.0.0.0:*          LISTEN

rg@zaigor:~$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 zaigor.enneenne.com ESMTX Exim 3.36 #1 Wed, 12 Mar
2003 13:25:48 +0100

root@zaigor:~$ tcpdump host zaigor and arp
tcpdump: listening on eth0
13:30:47.465011 arp who-has goldrake.enneenne.com tell
zaigor.enneenne.com
13:30:47.465126 arp reply goldrake.enneenne.com is-at
0:e0:4c:69:30:de
```

Applicazioni generiche

Quando si deve realizzare un progetto alcune volte è preferibile fare una fase di studio per vedere cosa è già disponibile.

Se decidiamo di utilizzare software libero abbiamo a disposizione una quantità incredibile di programmi che risolvono già, nella maggior parte dei casi, il nostro problema o una sua parte (ad esempio la distribuzione Debian «*Woody*» ci mette a disposizione quasi 11.000 pacchetti software!).

Normalmente una applicazione di un sistema UNIX è composta da più applicazioni (semplici) che risolvono un particolare problema (e lo fanno bene).

Esistono applicazioni per le basi di dati (MySQL e PostgreSQL) per le applicazioni in rete (Apache, ftpd, Sendmail/Exim), ecc..

Applicazioni generiche - 2

Una applicazione molto potente per applicazioni distribuite sulla rete è il super server `inetd`; grazie a questo strumento è possibile remotizzare una qualsiasi applicazione che inizialmente non è stata progettata per funzionare in rete.

Quello che fa `inetd` è collegare lo *stdin*, *stdout* and *stderr* di un processo sul socket utilizzato per la connessione con un client. Così facendo il processo crederà di scrivere/leggere dati in locale ma, in realtà, questi ultimi viaggeranno sulla rete verso il client remoto.

```
#!/bin/bash
while true ; do
    read cmd
    $cmd
done
exit 0
```