

Percorso 5:

- Moduli del kernel



Bartolomeo Montrucchio

`bartolomeo.montrucchio@polito.it`

Giovanni Squillero

`giovanni.squillero@polito.it`

Kernel

- Microkernel
- Kernel monolitico (Linux)
- Utilizzo dei moduli

Linux Kernel(1)

- Gestisce tutte le risorse hardware: CPU, memoria, I/O
- Fornisce un insieme di API (system call), circa 300
 - Permettono alle applicazioni utente e alle librerie di utilizzare l'hardware
- Gestisce la concorrenza e l'uso dell'hardware:
 - Scheduling dei processi
 - Condivisione dell'HW, ad esempio della scheda di rete
 - Lavora in interruzione (nelle ultime versioni il kernel può essere esso stesso interrotto)

Linux Kernel(2)

- Le system call sono stabili nel tempo
 - Solo nuove system call possono venire aggiunte
- Le informazioni di sistema e del kernel sono messe a disposizione tramite pseudo filesystem (anche detti filesystem virtuali)
- I più importanti sono:
 - /proc
 - /sys

/proc

- Il filesystem proc, montato su /proc, contiene informazioni su:
 - Processi
 - Memoria
 - Parametri di gestione
 - Vi è una directory per ciascuno dei processi in esecuzione
 - Ad es. In /proc/cpuinfo vi sono i dettagli sulla CPU

/sys

- Il filesystem sysfs, montato su /sys, contiene informazioni su:
 - device
 - Driver
- È di fatto una interfaccia tra spazio utente e spazio kernel
- Esempi sono [7]:
 - /sys/modules
 - /sys/kernel
 - /sys/fs
 - /sys/firmware
 - /sys/dev

Esercizio

- Utilizzando le pagine di manuale, provare ad esplorare le directory `/proc` e `/sys`, con particolare riferimento ai moduli del kernel già installati
- Provare a rifare un esempio di questo genere:

```
$ sleep 100 &
```

```
[1] 4357
```

```
$ ls /proc/4357
```

```
...
```

```
$ cat /proc/4357/cmdline
```

```
sleep100
```

/dev

- Contiene i device (dispositivi virtuali) utilizzati da Linux per virtualizzare i dispositivi fisici
- Il gestore dei device è Udev
- Udev crea di volta in volta solamente i device (in /dev) corrispondenti ai device fisici realmente collegati
- Si notino in particolare
 - /dev/sda per il primo disco
 - i tty per le console (ALT-F1 etc...) e i pts per i terminali all'interno dell'interfaccia grafica (usare who per vederli)

modprobe

- Per vedere le opzioni settate per il modulo in memoria:
 - `systool -v -m module_name` (installare `sysfsutils` se necessario)
- Modprobe può vedere la configurazione e le dipendenze di un modulo
 - `modprobe -c | grep module_name`
 - `modprobe --show-depends module_name`
- `modprobe nome_modulo` carica il modulo (più efficacemente di `insmod`)

lsmod

- Permette di verificare quali e quanti moduli siano caricati
- Dato un modulo si può utilizzare:
 - `modinfo nome_modulo`Per avere informazioni su quel modulo
- Ad es.:
 - `modinfo bluetooth`

Caricamento dei moduli

- I moduli sono caricati in modo automatico da udev
- Per caricarne un altro esterno o per bloccarne uno va inserito in `/etc/modules`
- Se è necessario passargli dei parametri, vanno messi in `/etc/modprobe.d/nomemodulo.conf` come opzione:
 - `options thinkpad_acpi fan_control=1`
- Si può anche passare l'opzione tramite il bootloader
- Ad es. in Grub bisogna premere e alla visione del menù e poi aggiungere il parametro e premere b per il boot [8]

lspci

- Permette di individuare i dispositivi PCI collegati
- lspci permette per esempio (sulla macchina virtuale su portatile Apple) di vedere:
 - la scheda grafica Innotek
 - il controller USB Apple

lsusb

- Permette di vedere i dispositivi collegati all'USB
- Altri comandi utili per avere una descrizione dell'hardware disponibile:
 - lshw
 - cat /proc/cpuinfo
 - dmidecode, per il BIOS
 - iwconfig (per la scheda di rete wireless)
- Se i risultati sono molti si possono reindirizzare su file

Kernel module – un esempio

- Per eliminare un modulo si può usare `rmmmod`
- Oppure mettere in blacklist il modulo che si desidera bloccare
- La lista nera è in `/etc/modprobe.d/blacklist.conf`
- Si possono bloccare solo i moduli non presenti nel kernel, dunque solo quelli caricati separatamente

Esercizio

- Provare a replicare l'esempio della slide precedente
- In particolare provare con il modulo e1000
- Cosa succede? Provare anche con rmmmod
- Provare poi con il modulo lp e seguire quanto riportato in [9]

Fase di boot

- La sequenza di boot è divisa in varie fasi:
 1. Hardware boot
 2. Loader del sistema operativo
 3. Startup del kernel
 4. Init e inittab
 5. Script di boot
- Fare riferimento a man boot

Fase di boot(1)

- Hardware boot
 - Self test (BIOS) e lettura parametri da CMOS (NVRAM); UEFI sarà il nuovo standard al posto del classico BIOS
 - Via hardware viene fatto accesso al device di boot, caricando il loader del sistema operativo e trasferendo il controllo ad esso
 - Esiste anche il boot via rete, tramite vari protocolli, tra cui DHCP, TFTP, PXE, Etherboot

Fase di boot(2)

- Loader del sistema operativo
 - Nei personal computer, esso è nell'MBR (primo settore del dispositivo di boot)
 - La dimensione limitata dell'MBR (512 byte) rende quasi impossibile installare un vero e proprio OS loader in esso
 - Viene dunque chiamato un loader secondario situato su di una partizione del disco
 - Normalmente si usa LILO oppure GRUB
 - Possono entrambi funzionare da loader secondario (richiamati da un MBR installato dal DOS) oppure da loader in due parti, completo di MBR+seconda parte dalla partizione di root
 - Il compito del loader è di trovare il kernel nel disco, caricarlo e lanciarlo, eventualmente passandogli dei parametri
 - Vi è solitamente anche un minimo di interfaccia

Fase di boot(3)

- Startup del kernel
 - Quando il kernel è caricato, esso inizializza i dispositivi (tramite i relativi driver), fa partire lo swapper (kswapd) e monta il filesystem di root (/)
 - I parametri passati al kernel possono modificare tale comportamento (ad esempio modificare il root filesystem di default)
 - Il kernel crea il primo processo, con PID 1; esso esegue /sbin/init, passandogli eventuali altri parametri

Runlevel

- Indica il runlevel attuale
- Legge da `/var/run/utmp`

Esercizio

- Provare ad individuare i vari runlevel disponibili
- Provare a verificare la presenza del processo init
- Provare ad individuare il runlevel di Ubuntu
- Provare a cambiare il runlevel a 1
 - Cosa succede? Provare a fare ssh su 127.0.0.1
- E provando init 2?

Fase di boot(4)

- Init e inittab
 - Init provvede a leggere /etc/inittab per ulteriori istruzioni
 - Su Ubuntu inittab non è presente, le sue funzioni vengono svolte da altri script, i quali definiscono i vari run-level, ciascuno associato ad un insieme di servizi (di solito S è per singolo utente, il livello 2 è quello in cui la maggior parte dei servizi di rete partono, 3 è multiuser, 5 è il livello multiuser con interfaccia grafica)
 - Il livello può essere cambiato con init
 - Tramite gli script i servizi vengono di fatto lanciati o fermati

Fase di boot(5)

- Script di boot
 - Per ciascun servizio tra quelli gestiti vi è uno script di startup posizionato in `/etc/init.d`
 - Tali script accettano come parametri `start`, `stop` e in taluni casi `restart`
 - Lanciati senza parametri gli script mostrano i parametri utilizzabili
 - OpenBSD ed altri sistemi (anche Slackware Linux) utilizzano uno schema di script di boot diverso

/etc/rc[0-6S].d

- Si tratta di directory nelle quali sono presenti dei link (solitamente simbolici) agli script reali che sono posti in /etc/init.d
- I vari file (link agli script) vengono richiamati in ordine:
 - i link con nome che inizia con S vengono richiamati con il parametro start
 - quelli con la K servono allo stop del servizio
- Dopo S o K vi è un numero e la descrizione del servizio; gli script vengono richiamati in ordine di numero, dato il runlevel di interesse

Boot in Ubuntu [si veda 10]

- Le directory ed i file di configurazione importanti sono:
 - /etc/init pur non essendo dei veri script, i file presenti nella directory contengono quanto serve per rimpiazzare i precedenti script sysvinit
 - /etc/init.d per retrocompatibilità la directory contiene script compatibili, che pure chiamano service myservice start
 - /etc/init/rc-sysinit.conf controlla l'esecuzione degli script tradizionali aggiunti a mano o con update-rc.d ai livelli /etc/rc*
 - /etc/default contiene file di configurazione sia per i tradizionali sia per i nuovi script basati su upstart (service myservice start)

Partenza di un servizio

Tradizionale:

```
/etc/init.d/myservice start
```

Modalità Upstart

```
service myservice start
```

Stop di un servizio

Tradizionale:

```
/etc/init.d/myservice stop
```

Modalità Upstart

```
service myservice stop
```

Elenco dei servizi disponibili

Tradizionale:

```
ls /etc/init.d
```

Modalità Upstart

```
service --status-all
```

- Si noti che la modalità Upstart è in grado di mostrare sia i servizi tradizionali sia quelli di tipo upstart

Aggiungere un servizio

Tradizionale:

update-rc.d apache2 defaults

Modalità Upstart

Non vi è il concetto di runlevel.

È necessario aggiungere un upstart config in
/etc/init e editare un file in /etc/default

Cancellare un servizio

Tradizionale:

```
rm /etc/rc*/*myscript
```

Modalità Upstart

Se non c'è nulla in /etc/default, editare config in
/etc/init

Esercizio

- Individuare, tramite ricerca sui manuali e/o su Internet i servizi per:
 - la rete
 - ssh
- Provare ad abilitarli e disabilitarli
- Provare ad inserire in rc.local un semplice script capace di stampare un messaggio
- Provare a creare un servizio di un tipo qualsiasi

Bibliografia

- <https://wiki.ubuntu.com/>
- <http://wiki.ubuntu-it.org>
- <http://help.ubuntu-it.org/>
- <http://free-electrons.com/docs/>
- https://wiki.archlinux.org/index.php/kernel_modules
- [https://help.ubuntu.com/community/Loadable Modules](https://help.ubuntu.com/community/Loadable_Modules)
- [7] <http://www.linux.org/threads/sysfs-and-configfs.4956/>
- [8] [https://wiki.archlinux.org/index.php/Kernel_parameters#When starting the kernel](https://wiki.archlinux.org/index.php/Kernel_parameters#When_starting_the_kernel)
- [9] <http://askubuntu.com/questions/134266/how-do-i-prevent-the-lp-module-from-loading-on-boot>
- [10] <https://help.ubuntu.com/community/UbuntuBootupHowto>

These slides are licensed under a **Creative Commons**

**Attribution
Non Commercial
Share Alike
4.0 International**

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Versione in Italiano:

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>

