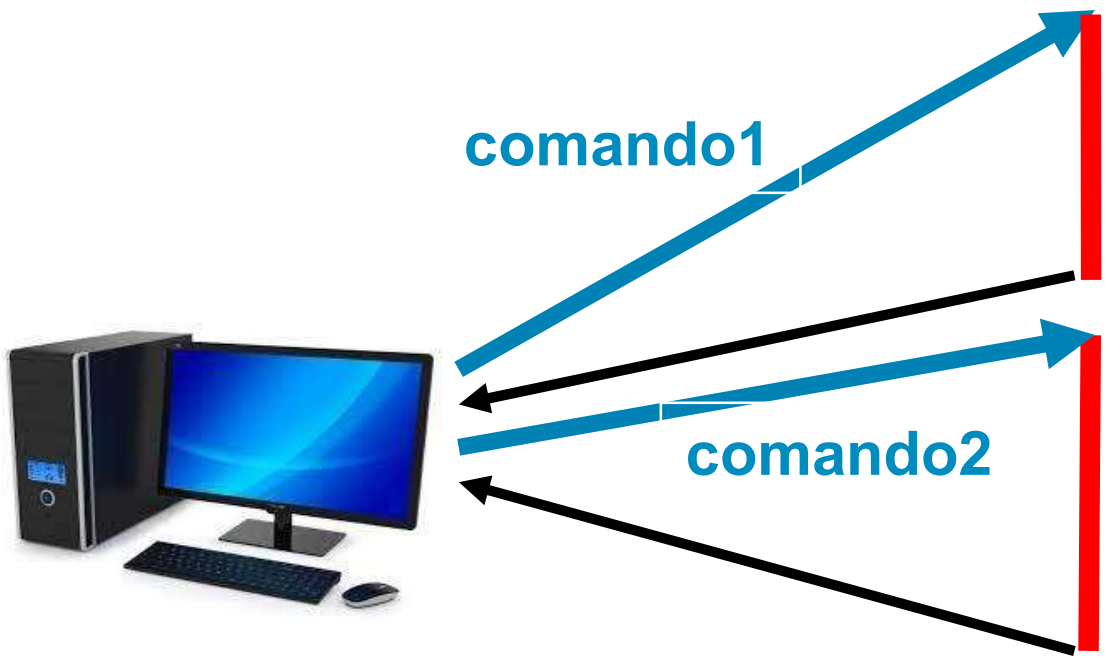
Processi

Comandi di shell per la gestione di processi

Stefano Quer
Dipartimento di Automatica e Informatica
Politecnico di Torino

Esecuzione in foreground

- I comandi di shell "standard"
  - Permettono di eseguire processi in modo **sequenziale**
  - Tali processi sono eseguiti in **foreground**



- `comando1`

```
Output comando1
```

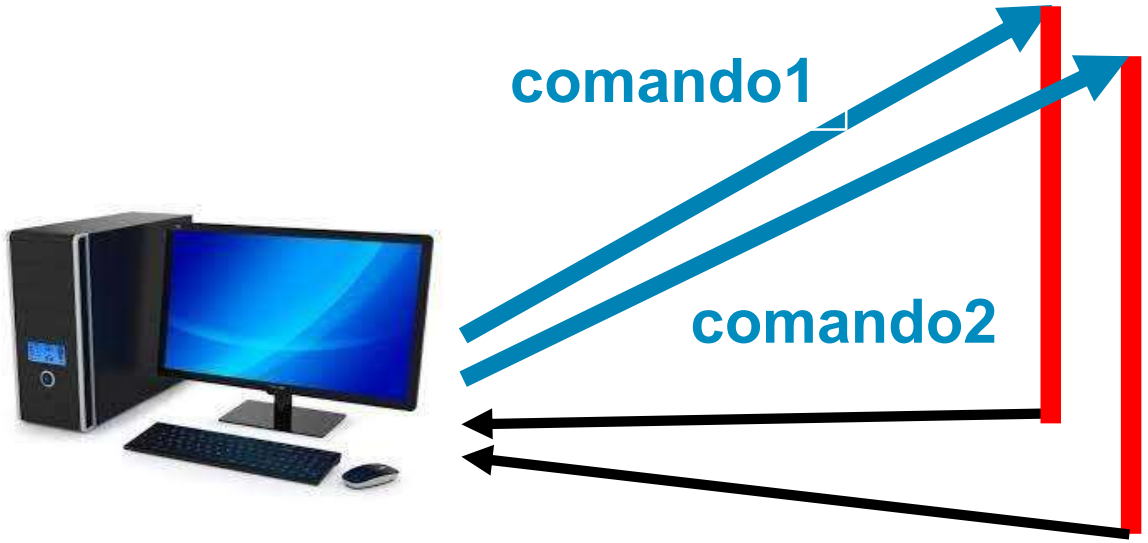
- `comando2`

```
Output comando2
```

comando1; comando2; …
Esecuzione sequenziale

Esecuzione in background

- L'utilizzo del carattere & permette di eseguire processi in **background**
  - Il processo viene eseguito in maniere indipendente dalla shell
  - Lascia il terminale libero per altri lavori
  - È possibile eseguire processi in parallelo



- `comando1 &`
- `comando2 &`

```
Output comando1
Output comando2
```

Comando ps

- Esistono due comandi principali per visualizzare lo stato dei processi
  - Il comando **ps** (**process status of active process**)
    - Elenca i processi attivi e i relativi dettagli
    - Senza opzioni (default) stampa (in formato compatto) lo stato dei processi con stesso user ID dell'utente da cui si effettua il comando

La shell è il padre di tutti i comandi di shell

Comando ps

```
ps [opzioni]
```

Opzioni			
Formato		Significato	Effetto
Compatto	Esteso		
-a			Elenca i processi di tutti gli utenti del sistema
-u			Visualizza informazioni più dettagliate (resident size, virtual size, etc.)
-u user			Visualizza i processi dell'utente <user>
-x			Aggiunge all'elenco i processi che non hanno un terminale di controllo (e.g. daemon)

Comando ps

Opzioni			
Formato		**Significato**	**Effetto**
Compatto	**Esteso**		
-o, -A			Elenca tutti i processi running nel sistema
-f			Visualizza le informazioni in format steso
r (non –r)			Visualizza solo i processi “running”

Comando ps

- ps

```
  PID TTY          TIME CMD
 1954 pts/0    00:00:00 bash
 1966 pts/0    00:00:00 ps...
```

Per una descrizione dei vari campi man ps

- ps -aux

```
USER   PID %CPU %MEM    VSZ   RSS TTY   STAT START   TIME COMMAND
root     1  0.2  0.0 168388 11688 ?     Ss   09:22   0:00 /sbin/init splash
root     2  0.0  0.0      0     0 ?     S    09:22   0:00 [kthreadd]
root     3  0.0  0.0      0     0 ?     I<   09:22   0:00 [rcu_gp]
root     4  0.0  0.0      0     0 ?     I<   09:22   0:00 [rcu_par_gp]
root     5  0.0  0.0      0     0 ?     I<   09:22   0:00 [netns]
...
quer  2012  0.4  0.0 755788 35116 ?     SNsl 09:28   0:00 bash
quer  2013  0.0  0.0   2496   508 pts/0    S 09:43   0:00 ./a 2 10
quer  2014  0.0  0.0      0     0 pts/0    Z 09:43   0:00 [a] <defunct>
quer  2015  0.0  0.0  11696  3596 pts/0   R+ 09:28   0:00 ps -aux
```

zombie

Comando top

- Il comando **top**
  - Visualizza informazioni sui processi in esecuzione, aggiornate in run-time

```
top

top - 10:26:58 up 57 min,  3 users,  load average: 0.00, 0.01, 0.05
Tasks: 152 total, 2 running, 150 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.0 us, 0.6 sy, 0.4 ni, 93.5 id,  1.4 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem:  8177092 total,  1382976 used,  6794116 free,   174096 buffers
KiB Swap: 10482684 total,        0 used, 10482684 free.   544664 cached Mem

 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
1821 quer      20   0 1297200 198644  39328 S 65.6  2.4   1:59.62 compiz
1302 root      20   0  326708 101316  17712 S 13.1  1.2   0:23.63 Xorg
   1 root      20   0   33648   3028   1492 S  0.0  0.0   0:00.78 init
   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
...
```

Comando kill

```
kill [-sig] pid
```

- Per inviare un segnale a un processo dalla linea di comando di una shell è possibile utilizzare comando **kill**
  - Invia il segnale sig al processo di PID pid
- Opzioni e parametri
  - L'opzione **sig** indica il codice del segnale
  - Il parametro **pid** è il process identifier (PID) del processo a cui inviare il segnale

Comando kill

- Ogni segnale **sig** può essere indicato mediante un nome o il numero corrispondente
  - La lista dei segnali disponibili può essere ottenuta mediante l'opzione "-l"
    - SIGKILL = KILL = 9
    - SIGUSR1 = USR1 = 10
    - SIGUSR2 = USR2 = 12
    - SIGALRM = ALRM = 14
    - etc.
  - Il segnale di default del comando kill è **SIGTERM** (o **TERM**) commando di terminazione standard

Comando kill

- Esempi

```
kill -l


kill -9 10234
kill -SIGKILL 10234
kill -KILL 10234
```

Fornisce l'elenco dei segnali riconosciuti dal sistema

Tre comandi equivalenti per terminare (incondizionatamente) il processo di process identifier (PID) 10234

Comando killall

```
killall [-sig] name
```

- Il commando di shell **killall** termina tutti i processi di dato nome
  - Può essere utile per terminare tutti i processi generate dallo stesso programma senza specificarne esplicitamente tutti i PID
- Esempio

```
killall -9 myProgram
```