

Sistemi Operativi 2

Marco Casu



Contents

1	Introduzione	3
1.1	Breve Panoramica su Unix	3
1.2	La Shell	3

1 Introduzione

1.1 Breve Panoramica su Unix

Moltissimi sistemi operativi moderni, come *MacOs*, *Linux*, e molti altri, sono basati su **Unix**, un sistema operativo il cui sviluppo cominciò nel lontano 1965. Inizialmente implementato totalmente in assembly, e limitato esclusivamente ad un tipo di architettura, si decise di costruire dei linguaggi di programmazione di più alto livello per garantire la portabilità, le versioni di Unix nei linguaggi *B* e *C* permisero di portare Unix su diverse CPU.

Venne distribuito con codice sorgente a centri di ricerca ed università, si diffuse rapidamente e nacquero diverse versioni. Le caratteristiche principali di un OS basato su Unix sono le seguenti:

- Supporta più utenti e la multiprogrammazione
- Il file system ha un'organizzazione gerarchica
- Ha un kernel rappresentante il cuore del sistema
- Si interagisce col kernel tramite le chiamate di sistema
- Ha una *shell* (si vedrà in seguito)
- È modulare e fornisce ambienti di programmazione

È composto da una serie di programmi limitati che eseguono molto bene un compito specifico, presentano un output minimale, sono detti "silenziosi", e lavori più complessi possono essere svolti componendo ed articolando diversi programmi semplici. I programmi solitamente manipolano file di testo (interpretabili dall'uomo secondo la codifica ASCII) e non file binari. Qualsiasi risorsa può essere vista come file o come processo.

1.2 La Shell

La **Shell** non è altro che un programma che esegue dei comandi, che possono essere scritti sul terminale, esistono vari tipi di shell, come quella denominata *bash*. La *bash* scrive il cosiddetto *prompt*, ossia una dicitura che indica che il terminale è in attesa di ricevere comandi, molto spesso è costituito dal nome dell'utente, il nome del computer, ed il percorso della directory nella quale è aperto il terminale.

```
nomeutente@nomemacchina:~$
```

in attesa di ricevere comandi

Ogni **comando** seguirà il seguente template : Prima il nome del comando, poi le varie opzioni, e poi gli argomenti del comando, ci deve essere almeno un argomento obbligatorio, e zero o più argomenti opzionali, gli argomenti vanno separati da un carattere se indicato. *Esempio* :

```
casufrost@debian:~$ ps -p $$ -ocmd -h
```

Uno dei comandi fondamentali è **man**, e sta per *manuale*, è un comando fondamentale che fornisce le informazioni più autorevoli possibile riguardo un comando, basta chiamarlo, dando come argomento appunto, il nome di un comando, e fornirà una lista dettagliata di tutte le opzioni ad esso correlato.

casufrost@debian:~\$ man ps produce il seguente output :

```
casufrost@debian: ~
PS(1) User Commands PS(1)

NAME
    ps - report a snapshot of the current processes.

SYNOPSIS
    ps [options]

DESCRIPTION
    ps displays information about a selection of the active processes. If
    you want a repetitive update of the selection and the displayed
    information, use top instead.

    This version of ps accepts several kinds of options:

    1  UNIX options, which may be grouped and must be preceded by a dash.
    2  BSD options, which may be grouped and must not be used with a dash.
    3  GNU long options, which are preceded by two dashes.

Manual page ps(1) line 1 (press h for help or q to quit)
```

Consultando il manuale è possibile capire come utilizzare i comandi fondamentali :

- **ps** - Mostra le informazioni dei processi attualmente in esecuzione.
- **ls** - Mostra una lista delle risorse contenute in una directory.
- **cp** - Copia file e directory.

Analizziamo il comando di esempio scritto in precedenza, ossia:

casufrost@debian:~\$ ps -p \$\$ -ocmd -h tale comando mostra i processi attualmente in esecuzione, l'opzione serve a selezionare il processo in base al suo PID, e prede come parametro \$\$, ossia il codice identificativo del processo bash.

Il comando mostra varie informazioni, come il PID del processo ed il nome, L'opzione **-o** serve a selezionare solo uno dei parametri del processo, in questo caso prende come argomento **cmd** e seleziona quindi solo quel campo, ossia il nome. Si noti come in questo caso non è necessario lasciare uno spazio fra l'opzione e l'argomento. L'ultima opzione, ossia **-h**, serve a rimuovere il nome dei campi visualizzati, ci si aspetta quindi che tale comando in output restituirà esclusivamente il nome del processo bash :

```
casufrost@debian:~$ ps -p $$ -ocmd -h
bash
```

Un altro esempio :

```
casufrost@debian:~$ ps -p $$
  PID TTY          TIME CMD
 3134 pts/0    00:00:00 bash
```

Durante la configurazione di un sistema Unix è necessario specificare almeno un utente. Differenti utenti hanno differenti privilegi, l'utente **root**, o **superutente**, è predefinito in ogni

sistema e possiede tutti i privilegi, è detto *amministratore di sistema*.

Tale utente però, non può effettivamente eseguire un login ed entrare nel sistema, se necessario eseguire un'operazione privilegiata, gli altri utenti possono acquisire temporaneamente i diritti di amministratore tramite il comando **sudo**, acronimo di *super user do*.

Gli utenti appartengono a dei *gruppi* che definiscono diversi privilegi, coloro che possono richiedere temporaneamente i diritti di amministratore appartengono al gruppo dei *sudoers*, è possibile mostrare a quali gruppi appartiene un utente tramite il comando **groups**.

sudo è un comando che prende come input un altro comando da eseguire in modalità root, esiste anche un altro comando chiamato **su**, e sta per *substitute user*, e serve per cambiare utente, e diventare possibilmente amministratore, risulta comunque meno sicuro del comando **sudo**, in quanto quest'ultimo permette solo di eseguire un'operazione in maniera privilegiata, senza rimanere nella condizione di root.