# Introduzione ai comandi di terminale Unix/Linux

*Giorgio Grisetti*
*Luca Iocchi*
*Daniele Nardi*
*Alberto Pretto*

*Dipartimento di Ingegneria Informatica, Automatica e Gestionale*
*Facoltà di Ingegneria dell'Informazione, Informatica, Statistica*
*Università di Roma "La Sapienza"*

*Edizione 2017/2018*


2018

# Indice

# 1. Comandi di terminale Unix/Linux

## 1.1. File system

Il *file system* è una struttura del sistema operativo Unix/Linux che consente la gestione dei file memorizzati su disco.

Il file system Unix/Linux è organizzato in *directory*. Ogni directory contiene *file* e altre directory formando quindi un albero. Directory e file sono specificati con dei nomi e sono *case-sensitive*. La directory d1 è quindi diversa dalla directory D1.

Ogni file o directory è associato ad un *percorso* o *path* che è denotato da una sequenza di directory separate dal simbolo /. La directory radice (o directory principale) del file system è la directory al livello più alto del file system. Essa è denotata dal path /. Un file test.c che si trova nella directory tmp che si trova nella directory radice viene indicato con il path /tmp/test.c. In generale, un file x che si trova nella directory d3 (inclusa nella directory d2 (inclusa nella directory d1 (inclusa nella radice del file system))) ha path /d1/d2/d3/x.

Un path può essere assoluto o relativo. Un path assoluto indica un file o directory nel file system a partire dalla directory principale del file system e viene indicato sempre con il simbolo / iniziale. Ad esempio, /d1/d2/d3 è un path assoluto (inizia con /) e denota la directory d3, inclusa in d2, inclusa in d1, inclusa nella directory radice del file system.

Un path relativo invece non inizia con il simbolo / e viene considerato in relazione alla directory corrente in cui esso viene utilizzato. Ad esempio, il path relativo d2/d3 usato dalla direcory /d1 equivale al path assoluto /d1/d2/d3. Si noti che il path relativo d2/d3 usato dalla directory /e1 equivale invece al path assoluto /e1/d2/d3.

Il simbolo . viene usato per indicare la directory corrente, mentre

il simbolo .. viene usato per indicare la directory precedente a quella corrente. Ad esempio, se ci troviamo nella directory /d1/d2/d3 il path . indica la directory stessa, il path .. indica la directory /d1/d2, mentre il path ../e3 indica la directory /d1/d2/e3.

### 1.1.1. Terminale

È possibile gestire (accedere, creare, modificare, eliminare) i file tramite comandi di terminale. Un terminale è un programma che consente all'utente di eseguire comandi tramite inserimento da tastiera e mostra a video l'output dell'esecuzione di tali programmi. Nel seguito saranno evidenziati i principali comandi di terminale per la gestione dei file.

Per avviare un terminale dall'ambiente grafico Linux, usare la corrispondente opzione dal menù. Ad esempio, nei sistemi Ubuntu selezionare dal menù di sistema il programma `Terminale` specificato. Ad esempio, nella macchina virtuale BIAR, il terminale installato si chiama `ROXTerm` e può essere eseguito dal menu `System Tools`, oppure selezionando l'icona in basso a sinistra, oppure tramite la combinazione di tasti ALT + c.

### 1.1.2. Navigazione nel file system

All'avvio di un terminale, viene presentato un prompt di comandi, che attende comandi dall'utente. Un esempio di struttura del prompt di comandi è il seguente:

```
utente@macchina:path$
```

Il path iniziale è la *home* dell'utente. La directory home ha tipicamente path /home/nomeutente e può essere indicata anche con il simbolo ∼.

Nota: nei Linux sistemi con tastiera italiana il simbolo ∼ si ottiene premendo i tasti ALT GR + ì.

Ad esempio, il seguente prompt indica che ci troviamo nella home (∼) dell'utente biar sulla macchina di nome pfp-VirtualBox.

```
biar@pfp-VirtualBox:~$
```

I comandi principali per la navigazione e gestione di un file system sono i seguenti

```
pwd            stampa la directory corrente
cd <dir>       si sposta nella directory specificata
mkdir <dir>    crea una nuova directory
rmdir <dir>    rimuove una directory vuota
```

Nota: <dir> può indicare un path assoluto o relativo.

Ad esempio, con cd / possiamo spostarci alla directory principale del file system.

**ATTENZIONE!!!** La cancellazine di file o directory traimte il comando rm non è reversibile. Usare con cautela.

## 1.2. Gestione dei file

La gestione dei file (creazione, modifica, eliminazione) all'interno di una directory avviene mediante l'uso di programmi o comandi di terminale.

I comandi di terminale più comuni per la gestione dei file sono i seguenti.

```
ls   stampa la lista dei file nella directory corrente
ls <dir>  stampa la lista dei file nella directory dir
touch <file>            crea un file vuoto
cp <file> <newfile>     copia un file in un nuovo file
mv <oldfile> <newfile>  rinomina un file
rm <file>               cancella un file
```

Nota: <file> e <dir> possono essere path assoluti o relativi.

La creazione di file può avvenire anche mediante l'uso di programmi esterni. Ad esempio, un file può essere creato da un editor di testo (ad esempio, `gedit`), mediante un comando da terminale che specifica il nome del file da creare.

Ad esempio, il seguente comando

```
gedit primo.c &
```

esegue il programma `gedit` (editor di testo) per creare il file primo.c nella directory corrente. Il file sarà presente nella directory al momento del salvataggio da parte dell'editor. Nota: il simbolo & indica che il programma `gedit` sarà eseguito in background rispetto al terminale,

quindi sarà possibile eseguire altri comandi di terminale mentre `gedit` è in esecuzione. In mancanza del simbolo &, il terminale attende il termine dell'esecuzione del programma invocato, prima di poter inserire altri comandi.

L'editor può essere usato anche per modificare file esistenti, quindi l'invocazione del programma con un nome di file esistente consente l'apertura e la modifica di tale file.

Nota: per l'editing di file C nei sistemi con tastiera italiana i caratteri { e } si ottengono con le combinazioni di tasti SHIFT + ALT + [ e SHIFT + ALT + ].

Si crei ad esempio un file denominato primo.c contenente il seguente programma C.

```
#include<stdio.h>

int main() {
  printf("Hello\n");
}
```

Per essere eseguito, un programma C deve essere prima compilato. I file eseguibili prodotti dalla compilazione di programmi sorgente vengono creati dal compilatore. Ad esempio, se nella directory corrente abbiamo scritto un programma C corretto memorizzato nel file primo.c, il seguente comando genera il file eseguibile primo nella stessa directory.

```
g++ -o primo primo.c
```

Si può verificare la presenza del file con il comando ls.

### 1.2.1. Esecuzione di file eseguibili

L'esecuzione di file eseguibili da terminale avviene semplicemente scrivendo il path del file eseguibile al prompt del terminale.

Il path completo del programma può essere omesso se il file eseguibile si trova nelle *directory di sistema*, directory particolari note al sistema operativo. Ad esempio, per l'esecuzione del programma g++ non è necessario specificare il path in quanto il programma g++ si trova in una directory di sistema.

I programmi generati dal compilatore si trovano tipicamente nelle directory utente che non sono comprese nelle directory di sistema. Per

l'esecuzione di questi programmi quindi è necessario specificare il path (tipicamente il path relativo).

Ad esempio, per eseguire il programma primo nella directory corrente si userà il comando

```
./primo
```

## 1.3. Organizzazione directory e file per esercitazioni

Per lo svolgimento delle esercitazioni di Tecniche di programmazione, si suggerisce una struttura di directory e file contenuti nella directory TDP nella home dell'utente. Per un corretto uso di macchine condivise (ad esempio in laboratorio), si suggerische di creare una cartella associata alla propria matricola. All'interno della directory TDP, creare quindi una directory denotata con la propria matricola e al suo interno una directory per ogni esercitazione. In ogni directory di esercitazione scrivere i file C corrispondenti all'esercitazione.

Ad esempio, lo studente con la matricola 12345678 nello svolgimento dell'esercitazione 1 potrà configurare la struttura delle directory con i seguenti comandi da terminale a partire dalla sua home directory (cioè all'avvio del terminale).

```
mkdir TDP   (se non esiste)
cd TDP
mkdir 12345678
cd 12345678
mkdir esercitazione_1
cd esercitazione_1
gedit esercizio_1.1.c &
g++ -o esercizio_1.1 esercizio_1.1.c
./esercizio_1.1
gedit esercizio_1.2.c &
...
```

Nota: `gedit` stampa messaggi di warning sul terminale in cui esso viene eseguito. Tali messaggi possono disturbare la visualizzazione degli output del compilatore e dei programmi in esecuzione. Per evitare la stampa di tali messaggi di errore si può usare il seguente comando

```
gedit file.c &> /dev/null &
```

oppure si può eseguire il comando `gedit` in un terminale e i comandi di compilazione ed esecuzione di programmi in un altro terminale.

## 1.4. Compressione e decompressione di file e directory

Durante lo svolgimento delle esercitazione e dell'esame sarà necessario comprimere e decomprimere file per ricevere i file di supporto per lo svolgimento delle esercitazioni/esami e per consegnare ai docenti il risultato delle esercitazioni/esami. Il comando Linux per effettuare le operazioni di compressione/decompressione dei file è `tar` e si usa con la seguente sintassi.

Per comprimere una directory in un file compresso:

```
tar czvf nomefilecompresso.tgz  directory
```

Nota: questo comando deve essere eseguito nella directory superiore a quella che si vuole comprimere e crea un file .tgz nella directory corrente.

Per decomprimere un file compresso in una directory:

```
tar xzvf nomefilecompresso.tgz
```

Nota: il file .tgz deve essere presente nella directory corrente, oppure bisogna specificare un path assoluto o relativo. Il comando `tar` crea una nuova cartella all'interno della directory corrente con il contenuto dei file presenti nel file compresso.

Ad esempio, se si vuole decomprimere un file esercitazione_1.tgz scaricato dal web e memorizzato nel path ~/Downloads nella directory di lavoro di uno studente in una macchina condivisa configurata come indicato nella sezione precedente si possono eseguire i seguenti comandi:

```
cd TDP
mkdir 12345678    (se non esiste)
cd 12345678
tar xzvf ~/Downloads/esercitazione_1.tgz
```

Dopo lo svolgimento dell'esercitazione si vuole creare un file compresso contenente tutti i file scritti nella directory dell'esercitazione. Dalla directory esercitazione_1 in cui sono stati svolti gli esercizi, eseguire i seguenti comandi:

```
cd ..
tar czvf esercitazione_1.tgz esercitazione_1
```

Il file esercitazione_1.tgz nella directory corrente conterrà ora tutti i file contenuti nella cartella esercitazione_1.