

Sistemi Operativi

AAF - Secondo anno - 3CFU

A.A. 2024/2025

Corso di Laurea in Matematica

Bash: primi comandi

Annalisa Massini

Dipartimento di Informatica
Sapienza Università di Roma

Argomenti trattati

- 1 Comandi per la gestione di file e directory
 - Il comando `ls`
 - I comandi `mkdir` e `tree`
 - Comandi per la creazione di file
 - Installazione di pacchetti
 - Il comando `cat`

Directory e file

Il comando ls

Comando ls

- Il comando `ls [-la] [-R] [nomedir]` mostra il contenuto della directory `nomedir` (o della `cwd`, se `nomedir` non è dato)
 - `ls -l` mostra informazioni relative al contenuto della directory
 - `ls -a` mostra i file considerati *nascosti* (*hidden*), cioè i file con nomi che cominciano con il punto
 - `ls -R` mostra tutto il sottoalbero con radice in `nomedir`
 - `ls -s` mostra la dimensione dei file

Il comando `ls`

- Il comando `ls` ha molte opzioni e può fornire dettagli su file e directory:

```
ls [-a] [-c] [-u] [-R] [-l] [-i] [-n] [-S] [-h]  
[-1] [nomedir1 ... nomedirn] [nomefile1 ...  
nomefilen]
```

- Possiamo specificare *indifferentemente directory e files*:
visualizzeremo informazioni sulle directory o i files indicati

Il comando `ls`

- In particolare, usando l'opzione `-l` si visualizza un insieme di informazioni sui proprietari del file, sui permessi, sulle dimensioni e su data e ora

Esempio

```
drwxr-xr-x  3 massini massini  4096 Feb  22  2023 report
drwxr-xr-x  4 massini massini  4096 May  4  12:58 script
drwxr-xr-x 31 massini massini 12288 Dec 12  2022 verilog
-rw-r--r--  1 massini massini   197 Jul 16  11:30 nomi.txt
```

```
drwxr-xr-x  3 massini massini  4096 Feb  22  2023 report
drwxr-xr-x  4 massini massini  4096 May   4 12:58 script
drwxr-xr-x 31 massini massini 12288 Dec  12  2022 verilog
-rw-r--r--  1 massini massini   197 Jul  16 11:30 nomi.txt
```

Le informazioni visualizzate sono (da destra a sinistra):

- **Nome** Il nome del file o della directory
- **Data/Ora modifica** L'ultima volta che il file è stato modificato. Se l'ultima modifica è avvenuta più di sei mesi addietro, vengono mostrati la data e l'anno. Altrimenti, è mostrata l'ora del giorno.
- **Dimensione** *Per i file* quanto effettivamente occupato da quel file su disco, in bytes. *Per le directory* dimensione del file speciale corrispondente, che consiste in una lista di informazioni, senza riferimento alla dimensione dei file contenuti. Di solito le directory hanno una dimensione minima di 4kB, anche se sono vuote.

```
drwxr-xr-x  3 massini massini  4096 Feb  22  2023 report
drwxr-xr-x  4 massini massini  4096 May  4  12:58 script
drwxr-xr-x 31 massini massini 12288 Dec 12  2022 verilog
-rw-r--r--  1 massini massini   197 Jul 16  11:30 nomi.txt
```

Le informazioni visualizzate sono (da destra a sinistra):

- **Gruppo** Il nome del gruppo cui appartiene il proprietario.
- **Proprietario** Il nome dell'utente a cui appartiene il file
- **Numero di hard link** Un hard link è una copia di un file esistente (i due file condividono lo stesso i-node, che vedremo)
- **Permessi** Si distinguono tre tipi di accesso: lettura, scrittura, esecuzione. Nella rappresentazione dei permessi di accesso:
 - Il primo carattere è il tipo di file: "-" indica un file regolare, "l" indica un link, "d" indica una directory
 - Si distinguono i diritti di lettura (r), di scrittura (w) e di esecuzione (x)
 - La prima tripla mostra i diritti di accesso del proprietario del file, la seconda i diritti del gruppo, l'ultima i diritti per chiunque altro

Il comando `ls`

- L'opzione `-l` mostra anche una riga iniziale *total* per la directory di cui mostra il contenuto: si tratta della dimensione come numero di *blocchi* su disco (normalmente, 1 blocco ha dimensione di un 1kB)
 - si può vedere la dimensione dell'output in Mega (10^6) o in MegaByte (2^{20}) scrivendo:
`ls --block-size=M` o `ls --block-size=MB`
oppure in kilo ($1000 = 10^3$) o in kiloByte ($1024 = 2^{10}$) scrivendo:
`ls --block-size=k` o `ls --block-size=kB`
 - N.B. le dimensioni sono tutte approssimate per eccesso
- Per vedere i timestamp, si usa `-l` in combinazione con:
 - `-c` (per *ctime* changed time)
 - `-u` (per *atime* access time)
 - senza niente (per *mtime* modified time)

Il comando `ls`

- Notare che queste ed altre informazioni non sono sul `man`: vanno cercate nelle informazioni estese, reperibili con il comando `info ls`
- **Esercizio** verificare le dimensioni con `ls --block-size=M`, `ls --block-size=MB`, `ls --block-size=k` e `ls --block-size=kB` posizionandosi in diverse directory
- **Esercizio** creare un file, modificarlo, poi leggerlo e verificare che i valori per `atime` ed `mtime` cambino di conseguenza

Directory e file

I comandi `mkdir` e `tree`

Comando `mkdir`

- Il comando `mkdir [-p] nomedir` crea la directory `nomedir` (vuota)
 - Se si usa l'opzione `-p` si può creare una sequenza di directory annidate, cioè specificando il path `nomedir` composto da più directory si creano tutte le directory nel path (se non esistono)
 - **Esempio:** supponendo che `dir11` esista già, il comando `mkdir -p dir11/dir13/dir15`, creerà la directory `dir13` dentro `dir11`, e poi `dir15` dentro `dir13`
 - **Esercizio** provare il comando `mkdir` con e senza l'opzione `-p`
- **Esercizio**
 - creare l'intero albero di directory dato sopra (ovvero, `/home/utente1/dir1/dir3/dir7/`), posizionarsi dentro `dir1` e poi in `dir7` sia usando che non usando la directory parent ..
 - controllare il risultato usare il comando `ls`;
 - controllare come cambia il path riportato nel prompt

Comando `tree`

- Si può visualizzare un intero albero di directory con il comando `tree [-a] [-L maxdepth] [-d] [-x] [nomedir]`
 - potrebbe non essere installato e per installarlo si può scrivere:
`sudo apt-get install tree`
 - usare l'opzione `-L` per limitare la profondità dell'albero mostrato
- In generale: se si dà un comando sbagliato, e l'output sembra non finire mai, provare a premere `CTRL+c`

Directory e file

Comandi per la creazione di file

Comando touch

- Il comando `touch nomefile` crea il file `nomefile` (vuoto)
- Per aprire e scrivere in un file, si può usare un editor di testi
- Provare i comandi `geany nomefile &` o `gedit nomefile &` (se i corrispondenti editor sono installati)
- **Osservazione** il carattere `&` serve per l'esecuzione in background e permette di mantenere la finestra del terminale attiva (altrimenti il processo è bloccato)
- Si possono anche usare editor che si interfacciano direttamente con il terminale (se sono installati):
 - `nano nomefile` oppure `pico nomefile` (**senza** usare il carattere `&`)
 - oppure `vi nomefile` che è un editor un po' più complicato:
 - per uscire da `vi`, digitate i caratteri `:q` seguiti da invio

Il comando touch

- Qualche dettaglio in più su `touch [-a] [-m] [-t timestamp] {file}`
 - Il suo *vero* uso è quello di cambiare i timestamps negli attributi dei file dati
 - Li cambia normalmente tutti e tre, a meno che non si sia data l'opzione `-a` (solo atime) o `-m` (solo mtime)
 - Come *effetto collaterale*, se un file dato non esiste lo crea
 - Può essere usato anche su una directory (ma solo se esiste)
 - Con `-t timestamp` setta i timestamp del file al timestamp dato, anziché al timestamp del tempo attuale

Creare altri file (non di testo)

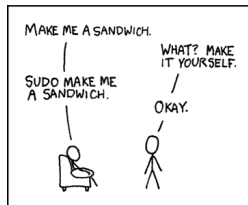
- Per creare file non di testo, occorre usare opportuni altri programmi (a seconda di cosa serve)
 - ad esempio, i vari applicativi di LibreOffice possono essere usati per creare file contenenti documenti formattati (DOC, DOCX, ODT, etc), fogli di calcolo (XLS, XLSX, ODS, etc)
 - basta usare il comando `libreoffice nomefile`
 - se `libreoffice` non è installato, va installato ma servono i diritti da amministratore (cioè occorre usare il comando `sudo apt-get install libreoffice`)
 - attenzione: `nomefile` deve già esistere; altrimenti si può eseguire `libreoffice` senza argomenti e poi usare l'interfaccia grafica per creare un nuovo documento del tipo desiderato

Comando xdg-open

- Il comando `xdg-open nomefile|url` apre un file o una url con l'applicazione selezionata come *preferita*
 - se l'argomento è un file sarà scelta l'applicazione preferita per il tipo di file
 - se l'argomento è una *url* sarà scelto il browser preferito
- Le opzioni sono:
 - `--help` mostra la sinossi del comando
 - `--manual` mostra la pagina del manuale relativa al comando
 - `--version` mostra le informazioni sulla versione di `xdg-utils`
- `xdg-open` fa parte del pacchetto `xdg-utils` che è un insieme di *tools* (o *utilities*) che permettono di integrare varie applicazioni con la distribuzione Linux utilizzata

Amministratore, utente e pacchetti

- Per installare nuovi pacchetti occorre avere i *privilegi da amministratore di sistema*
- Nel caso di Ubuntu, l'utente creato a tempo di installazione è un **sudoer**, cioè un *utente con privilegi di amministratore di sistema*, e appartiene al gruppo predefinito sudo
- *L'amministratore di sistema* può eseguire comandi da *superutente* semplicemente antepoendo al comando da eseguire la parola **sudo**, derivato da **super user do**
 - **sudo comando** è un comando particolare, che prende come argomento un altro comando, che può avere svariati argomenti



Directory e file

Il comando `cat`

Comando `cat`

- Il comando `cat [nomefile]`: scrive a schermo il contenuto di `nomefile`
 - funziona bene solo se il file è di testo (e se usa la codifica riconosciuta da `cat`), altrimenti scrive caratteri incomprensibili
 - si può usare `cat` per leggere più di un file alla volta scrivendo `cat file1 file2... fileN`:
 - in questo caso il comando `cat` stamperà prima il contenuto del primo file, poi quello del secondo, e così via
 - l'output sarà quindi la concatenazione del contenuto dei file specificati
 - senza argomenti, resta in attesa: se si scrive qualcosa e poi si preme invio, ripete quanto scritto, finché non si preme CTRL+d, che è il carattere EOF (*end-of-file*)