

Bash

- **var=val** → assegnazione
- **unset var** → eliminazione
- **\$var** → accesso
- **declare [opzioni] var**
 - **declare -p var** → mostra attributi var
 - **declare -i var** → imposta var con attributo intero
 - **declare +i var** → rimuove l'attributo intero da var
 - **declare --a="1"** → output che segnala l'assenza di attributi (--) della variabile a
- **read var** → lettura valore da input, viene salvato in var
- **echo arg** → stampa "arg" su terminale
 - **echo -e arg** → echo interpreta le sequenze di escape negli argomenti dopo aver interpretato i caratteri speciali
- **** → Escaping: il carattere successivo a \ non viene interpretato come speciale
- **` `** → Quoting forte: si disabilita ogni forma di interpretazione dei caratteri speciali
- **" "** → Quoting debole: mantiene l'interpretazione dei costrutti iniziati dai caratteri speciali \$ ` \
- **printf [formato] [arg]** → stampa output formattato, facendo uso delle stringhe di formato tipiche del linguaggio C
- **touch [nome_file]** → crea file vuoto
- **Manipolazioni di variabili:**
 - **\${var}** → manipolazione nulla, usata per evitare ambiguità
 - **\${var:-val}** → se var è definita la stampo, altrimenti stampo val
 - **\${var:=val}** → se var è definita la stampo, altrimenti la definisco e la pongo uguale a val
 - **\${#var}** → ritorna lunghezza di var
 - **\${var:start}** → estrae da var la sottostringa che inizia all'indice start
 - **\${var:start:len}** → estrae da var la sottostringa che inizia all'indice start ed è lunga len caratteri
 - **\${var#pattern}** → rimuove il match più piccolo di pattern all'inizio di var. var non è definita o modificata in nessun modo
 - **\${var##pattern}** → rimuove il match più grande di pattern all'inizio di var
 - **\${var%pattern}** → rimuove il match più piccolo di pattern alla fine di var
 - **\${var%%pattern}** → rimuove il match più grande di pattern alla fine di var
- **env** → stampa l'intero ambiente
- **test ESPRESSIONE** → testa la veridicità o meno di una certa espressione. Il risultato viene memorizzato in \$?
- **var=\$((1+2))** → assegna a var il risultato di un'espressione aritmetica
- **let var=1+2** → equivalente a **var=\$((1+2))**
- **Confronti aritmetici:**
 - **ARG1 -eq ARG2** VERO se ARG1=ARG2
 - **ARG1 -ne ARG2** VERO se ARG1≠ARG2
 - **ARG1 -lt ARG2** VERO se ARG1<ARG2
 - **ARG1 -gt ARG2** VERO se ARG1>ARG2
 - **ARG1 -le ARG2** VERO se ARG1≤ARG2
 - **ARG1 -ge ARG2** VERO se ARG1≥ARG2

- **Confronti tra stringhe:**

- **-z** *STR* VERO se *STR* ha lunghezza zero
- **-n** *STR* VERO se *STR* è non nulla
- *STR* VERO se *STR* non ha lunghezza zero
- ***STR1* == *STR2*** VERO se *STR1* e *STR2* sono uguali
- ***STR1* != *STR2*** VERO se *STR1* è diversa da *STR2*
- ***STR1* \< *STR2*** VERO se *STR1*<*STR2*
- ***STR1* \> *STR2*** VERO se *STR1*>*STR2*

- **Confronto logico:**

- *EXPR* ritorna il valore logico di *EXPR*
- **!*EXPR*** VERO se *EXPR* ha valore FALSO
- ***EXPR1* -a *EXPR2*** VERO se *EXPR1* e *EXPR2* lo sono
- ***EXPR1* -o *EXPR2*** VERO se *EXPR1* o *EXPR2* lo sono

- **Controllo del flusso:**

- Costrutto if

```
if COND_TEST1; then
    STATEMENTS1
elif COND_TEST2; then
    STATEMENTS2
else
    STATEMENTS3
fi
```

- Costrutto case:

```
case $var in
    VAL1)
        STATEMENTS1
    ;;
    VAL2)
        STATEMENTS2
    ;;
    VAL3)
        STATEMENTS3
    ;;
esac
```

- Costrutto while:

```
while COND_TEST1; ~
do
    STATEMENTS1
done
```

- Costrutto until

```
until COND_TEST1; ~
do
    STATEMENTS1
done
```

- Costrutto for:

```
for var in LIST;
do
    STATEMENTS1
done
```

- **Brace expansions**
 - $S_1\{S_2, S_3\}S_4 \rightarrow S_1S_2S_4 \quad S_1S_3S_4$
 - $\{0..N\} \rightarrow 0 \ 1 \ 2 \ \dots \ N$
 - $\{0..N..M\} \rightarrow 0 \ 0+M \ 0+2M \ \dots \ N$
- **Break** → statement che esce immediatamente dal costrutto di controllo del flusso corrente
- **Continue** → statement che esegue immediatamente la prossima iterazione del costrutto di controllo del flusso corrente
- **bash nome_script.sh** → esegue uno script (# per i commenti)
 - **-v** → Abilita la stampa di ogni statement prima della sua esecuzione consentendo di capire se e quando uno statement è eseguito
 - **-h** → Abilita l'esplicitazione di ogni statement prima della sua esecuzione consentendo di capire se variabili e parametri sono espansi correttamente. Queste stampe sono precedute dal carattere +, le altre sono gli output
- Definizione funzione:


```
function FUNC_NAME() {
    STATEMENTS
}
```
- **Parametri posizionali:**
 - **\$0** → nome dello script (continua a contenere il nome dello script anche all'interno delle funzioni)
 - **\$1, \$2, \$3 ...** → primo, secondo, terzo ... argomento (accessibili anche all'interno delle funzioni)
 - **\$?** → contiene il valore di ritorno (**n**) dopo l'invocazione di una funzione che ha uno statement **return n**
- **exit [intero]** → ritorna lo stato d'uscita di uno script
- **alias NOME_ALIAS='COMANDO'** → definizione alias
- **unalias NOME_ALIAS** → cancellazione alias
- **history** → se lanciato senza argomenti elenca tutti i comandi presenti nella storia
 - **history N** → Elenca gli ultimi N comandi nel buffer
 - **-c** → cancella il buffer, ma non il file di log
 - **-w** → Salva il buffer della storia sovrascrivendo forzando la sincronizzazione col file di log prima del termine di una sessione
 - **-a** → Appende il buffer della storia forzando la sincronizzazione col file di log prima del termine di una sessione
 - **-r** → Appende i comandi presenti nel file di log nel buffer
- **fc** → gestisce gli ultimi HISTSIZE comandi del buffer. Se lanciato senza argomenti viene avviato l'editor di default dove è possibile editare l'ultimo comando che verrà poi eseguito.
 - **-e** → cambia l'editor utilizzato da FC per quel comando
 - **fc N** → esegue l'N-mo comando presente nel buffer di storia dei comandi
 - **fc S** → viene ritornato l'ultimo comando che contiene S
 - Intervalli di comandi:
 - **fc N1 N2**
 - **fc S1 S2**
 - **fc N1 S2**
 - **fc S1 N2**
 - **fc -l [S]** → elenca i comandi nella cronologia (di default 16). Se viene passato S è possibile individuare il comando a partire dal quale iniziare l'elenco.
 - **fc -s [pattern=replace] [S|N]** → Individua il comando N-esimo (oppure che matcha con S) e sostituisce al suo interno pattern con replace.

- **Operatori di espansione nella storia utilizzabili nei comandi o da soli:**
 - **!N** ⇒ espande nell'N-mo comando presente nella storia
 - **!-N** ⇒ si espande nell'N-ultimo comando presente nel buffer della storia dei comandi
 - **!!** ⇒ si espande nell'ultimo comando immesso
 - **!S** ⇒ si espande nell'ultimo comando immesso che contiene la stringa S
 - **^S1^S2^** ⇒ sostituisce S1 con S2 nell'ultimo comando
 - **^S1^** ⇒ cancella la stringa S1 nell'ultimo comando
 - **E:s/S1/S2** Con il carattere : si può specificare con cosa si vuole sostituire
 - **E** è un operatore di espansione della storia dei comandi;
 - **S1** è la stringa da sostituire;
 - **S2** è la stringa sostituita.
- **Scorciatoie da tastiera**
 - **Navigazione nella storia**

Up, <CTRL>-p: comando precedente.

Down, <CTRL>-n: comando successivo.

<ALT>-<: inizio della storia.

<ALT>->: fine della storia.

<ALT>- .: ultimo argomento.

<CTRL>-N<ALT>- .: N-mo argomento (N=0, 1, ..., 9).

<CTRL>-r: ricerca incrementale all'indietro.

<CTRL>-a: inizio riga.

<CTRL>-e: fine riga.

<Right>, <CTRL>-f: un carattere avanti.

<Left>, <CTRL>-b: un carattere indietro.

<ALT>-f: una parola avanti.

<ALT>-b: una parola indietro.
 - **Navigazione singolo comando**

<CTRL>-a: inizio riga.

<CTRL>-e: fine riga.

<Right>, <CTRL>-f: un carattere avanti.

<Left>, <CTRL>-b: un carattere indietro.

<ALT>-f: una parola avanti.

<ALT>-b: una parola indietro.

, <CTRL>-d: cancella il carattere sotto il cursore.

<BACKSPACE>: cancella il carattere precedente.

<ALT>-d: cancella fino a fine parola.

<ALT>-<BACKSPACE>, <CTRL>-w: cancella fino a inizio parola.

<CTRL>-k: cancella fino a fine riga.

<CTRL>-u: cancella fino a inizio riga.

<CTRL>-_: undo.
- **type C₁ [C₂ ... C_n] →** ritorna la tipologia dei comandi in ingresso
 - **-a** vengono stampate tutte le tipologie del comando

Documentazione

- **help [COMANDO]** → fornisce la documentazione di un comando interno di BASH.
Nella sua forma più semplice, è eseguito senza opzioni e senza argomenti e fornisce la sinossi (sintassi) di tutti i comandi interni forniti da bash
 - **help "pattern"**: si ottiene la documentazione di tutti i comandi che matchano con il pattern.
E' possibile mischiare più patterne insieme a stringhe semplici:
help "a*" "b*" cd
- **man VOCE** → permette di "sfogliare" le pagine di manuale, in particolare la voce esatta passata come argomento. L'output di **man**, poiché voluminoso, viene dato in pasto ad un paginatore, solitamente il comando esterno **less**.
 - **man man** per scoprire i capitoli del manuale
 - **-s nr_sezione** è possibile specificare, direttamente nel comando man, la sezione di manuale in cui cercare.
Sono accettate anche le opzioni **-S nr_sezione** o direttamente **[nr_sezione]**
 - **nr_sezione1,nr_sezione2** è possibile specificare più sezioni
 - **nr_sezione1:nr_sezione2** è possibile specificare più sezioni
 - **-a VOCE** → consente di stampare tutte le pagine di manuale individuate
 - **--regex 'regex'** → permette di usare una espressione regolare per la selezione di pagine arbitrarie
man -a -S 2 --regex '.*' : Comando per stampare tutte le pagine di manuale di una data sezione
- **less** → paginatore usato per output molto lunghi
Comandi per navigare
 - **<Up>, <Down>, <Left>, <Right>** : spostamento.
 - **/:** immissione di una espressione regolare per la ricerca
 - **n:** ricerca la prossima occorrenza
 - **N:** ricerca l'occorrenza precedente
 - **Ng:** salta alla riga N (1g salta alla riga 1)
 - **G:** salta alla fine del file.
 - **h:** visualizza l'help in linea di less
 - **q:** esce
- **apropos KEY₁ [KEY₂... KEY_N]** → cerca i nomi esatti delle voci di manuale a partire da chiavi di ricerca testuali, inserendo più chiavi viene fatta la ricerca in OR
 - **-a** → ricerca in AND delle chiavi inserite
 - **-s S₁, S₂, ..., S_N** → S₁, S₂, ..., S_N sono una lista di sezioni specifiche nelle quali viene fatta la ricerca
 - **-r 'regex'** → è possibile fare uso di una regex per cercare un insieme di voci

File

- **stat[opzioni] nome_file** → visualizza i metadati di un qualsiasi file
- **ls [opzioni] FD₁ ... FD_N** → visualizza il contenuto di uno più file o directory
 - **-a** → Visualizzazione file e directory nascosti
 - **-l** → Visualizzazione dei metadati principali
 - **-h** → Visualizzazione in formato human readable
 - **-t** → Visualizzazione dei file in ordine di tempo di accesso decrescente (per primi i file che sono stati acceduti per ultimi)
 - **-S** → Visualizzazione dei file in ordine di dimensione di accesso decrescente
 - **-r** → Visualizzazione dei file in ordine inverso a quello selezionato
 - **-R** → Visualizzazione ricorsiva dei file e delle sottodirectory (analogo a tree)
- **lesspipe** → comando usato nei sistemi UNIX che, vedendo una determinata estensione di un file, fa le operazioni necessarie per poi mandarlo a less.
- **file FD₁ FD₂ ... FD_N** → stampa la tipologia di uno o più file o directory.
- **touch [opzioni] F₁ F₂ ... F_N** → nato per modificare i timestamp associati ad un file ma usato anche per la creazione di file.
 - **-a** → altera il timestamp di ultimo accesso
 - **-m** → altera il timestamp di ultima modifica dei dati
- **mkdir [opzioni] D₁ D₂ ... D_N** → crea directory vuote
 - **-p** → permette di creare directory inestate, non da errori se manca una cartella nel percorso, le crea tutte
- **rm [opzioni] F₁ F₂ ... F_N** → senza opzioni cancella file. Se si prova a cancellare una directory (vuota e non) **rm** segnala errori.
 - **-d** → permette di rimuovere una directory vuota
 - **-r** → ricorsiva di una cartella e del suo contenuto, (possibile anche per un file, ma non cambierebbe nulla dal comando rm senza opzioni)
 - **-i** → modalità interattiva: a ogni cancellazione viene chiesta una conferma
 - **-f** → modalità forzata, si ignorano file e directory non esistenti e non viene chiesta alcuna conferma
- **rmdir [opzioni] D₁ D₂ ... D_N** → per eliminare directory vuote
 - **-p** può cancellare una gerarchia di directory
- **cp [opzioni] F_{SRC} F_{DST}** → lanciato senza opzioni, copia file e/o directory.
 - **-a FD₁ FD₂ ... FD_N D** → abilita la modalità di copia archiviale.
Se i primi N argomenti sono file o directory e l'ultimo argomento è una directory, copia fedelmente (ovvero, cercando di preservare i metadati originali) tutti i file/directory nella directory destinazione.
- **cp F₁ F₂ ... F_N D** → Se i primi N argomenti sono file **F₁, F₂, ... F_N** e l'ultimo argomento è una directory **D**, cp copia tutti i file nella directory.
- **mv FD_{SRC} FD_{DST}** → sposta il file/directory **FD_{SRC}** nel file/directory **FD_{DST}**.
È possibile cambiare il nome del file contestualmente al suo spostamento. È sufficiente usare un nome diverso in **FD_{DST}**

- **cat [opzioni] F₁ F₂ ... F_N** → lanciato senza opzioni, stampa il contenuto di F₁, F₂, ... F_N in sequenza, visualizzazione non interattiva.
 - **-E** ⇒ stampa il carattere di fine linea con **\$**
 - **-T** ⇒ stampa il carattere di tabulazione con il simbolo **^I**
 - **-v** ⇒ stampa i caratteri non stampabili ossia caratteri di controllo ASCII e caratteri non ASCII
 - **-A** ⇒ elenca tutti i caratteri non stampabili (aggiunge anche i caratteri di fine linea e tab) . Essa è equivalente a **-vET**.
 - **n** ⇒ stampa il file e, per ogni riga, mette un numero progressivo
- **hexdump [opzioni] F₁, F₂, ... F_N** → senza opzioni stampa il contenuto F₁, F₂, ... F_N in esadecimale, a parole di 16 bit.
 - **-C** ⇒ abilita la visualizzazione in forma canonica. Di fianco a ogni parola da 16 byte c'è la corrispondente traduzione in forma di stringa.
 - **-v** ⇒ abilita la visualizzazione verbosa eliminando le compattazioni. Di default infatti per motivi di efficienza e compattezza **hexdump** non stampa righe ripetute di output ma le sostituisce col carattere *****.
- **head [opzioni] F₁, F₂, ... F_N** senza opzioni stampa le prime dieci righe di F₁, F₂, ... F_N
 - **-n** → riceve in argomento un numero N e stampa
 - Se N è positivo le prime N righe
 - Se N è negativo fino alle ultime N righe
 - **-c** → uguale a -n ma riceve in ingresso un numero di byte
- **tail [opzioni] F₁, F₂, ... F_N** → senza opzioni stampa le ultime dieci righe di F₁, F₂, ... F_N
 - **-n** → riceve in argomento un numero N e stampa
 - le ultime N righe
 - se specificato con il carattere + davanti (+N) stampa a partire dalla riga N.
 - **-c** → uguale a -n ma in byte
 - **-F** → visione dei file di log. Effettua in ordine queste operazioni
 - continua a provare l'apertura del file fino a quando non viene creato (se non esiste già)
 - mostra le ultime righe del file
 - si blocca in attesa di nuove righe da mostrare
- **cut [opzioni] F₁, F₂, ... F_N** → estrae elementi selezionati da uno o più file esistenti, Il carattere di delimitazione di default è **<TAB>**
 - **-d " "** → modifica del carattere delimitatore
 - **-f** → permette di selezionare campi (colonne) specifici di un file.
 - **N** ⇒ l'N-mo campo
 - **N-M** ⇒ dall'N-mo campo all'M-mo campo
 - **N,M** ⇒ l'N-mo campo e l'M-mo campo
 - **-M** ⇒ dal primo campo all'M-mo campo
 - **-c** → analoga ma specifica intervalli di caratteri invece che campi.
 - **-b** → analoga ma specifica intervalli di byte invece che campi.

- **tr [opzioni] F** → permette di effettuare operazioni di filtro sui file. Viene messo in | con un file ed eliminare gli eventuali doppi spazi per poi fare il cut sempre in |
 - **-s** " " → comprime il carattere passato, in questo caso lo spazio
- **paste F₁, F₂, ... F_N** → fonde uno o più file esistenti in un unico output multicolonnare.
- **sort [opzioni] F₁, F₂, ... F_N** → ordina file secondo specifici criteri. Senza opzioni svolge le operazioni seguenti:
 1. concatena i file **F₁, F₂, ... F_N**;
 2. considera l'intera riga come una singola chiave di ordinamento;
 3. ordina tutte le chiavi alfabeticamente.
 - **-n** → ordinamento numerico
 - **-d** → ordinamento alfanumerico (dizionario)
 - **-h** → ordinamento numerico "umano" (confronta numeri leggibili dagli umani, ad es. 2G e 1K)
 - **-M** → ordinamento del mese (confronta JAN, FEB, ...)
 - **-k** riceve come argomento una specifica testuale **KEY** (corrisponde a una porzione di una riga) del formato di una chiave, che illustra come recuperarla da una riga

Formato di **KEY**: **POS1 [, POS2]**

dove **POS2** è opzionale e specifica la posizione finale, **POS1** quella iniziale.

POS1 e **POS2** hanno un formato del tipo **F [. C] [OPTS]**

- **F (obbligatorio)** ⇒ indica il numero del campo desiderato
- **C (facoltativo)** ⇒ posizione del carattere nel campo
- **OPTS (fac.)** ⇒ lettera che corrisponde ad una posizione di ordinamento

Esempio

L'ID numerico richiesto è contenuto nel primo campo:

-k 1

L'ID numerico inizia al quarto carattere:

-k 1.4

L'ID numerico termina al sesto carattere:

-k 1.4,1.6

Si imposta l'ordinamento numerico per questa chiave:

-k 1.4,1.6n

⇒ Se **POS2** non è specificato si assume il fineriga.

- **-t** specifica il carattere separatore dei campi (di default uno o più spazi).
- **find [opzioni] [percorso] [espressione]** → individua file a partire dai suoi metadati.
 - **opzioni**: configurano le modalità operative di find
 - **percorso**: percorso iniziale di ricerca
 - **espressione**: criterio di individuazione su metadati

Senza opzioni e argomenti si assume che la directory di ricerca sia quella attuale e il criterio di individuazione sia "elenca tutto".

Se si passa invece solo una cartella allora elenca tutto quello che c'è in quella cartella

- **-[i]name pattern** → Per aggiungere un criterio di ricerca secondo uno specifico **pattern**
- **-[i]regex 'regex'** → Per individuare file in base al match di una espressione regolare
- **-atime n** → il file è stato acceduto l'ultima volta **n***24 ore fa.
- **-perm mode** → il file ha i permessi **mode**
- **-size n[cwbkMG]** → il file usa **n** unità spazio su disco
 - **+n** ⇒ più grande di **n**
 - **-n** ⇒ più piccolo di **n**
 - **n** ⇒ esattamente pari a **n**
- **-print** → (default) vengono stampati i percorsi dei file
- **-printf "%... "** → accetta una stringa di formato che permette di personalizzare la stampa delle informazioni relative ai file individuale
 - **%p** → nome file
 - **%s** → dimensione file in byte
 - **%m** → permessi dei file in ottale
 - **%M** → permessi del file in lettere
- **-exec COMMAND '{ }' \;** → accetta un comando di shell da eseguire su ciascun file individuato.
 - La stringa **'{ }'** viene sostituita col nome del file trovato.
 - I caratteri **\;** servono per terminare il comando.

Esempio utile

Per ogni file che finisce con .conf all'interno della cartella etc si stampa il tipo del file.

```
find /etc -name "*.conf" -exec file "{}" \;
```

● Caratteri speciali (regex):

- **.** → un carattere qualunque
- **+** → il carattere o l'espressione regolare precedente, ripetuta almeno una volta
- **?** → il carattere o l'espressione regolare precedente, ripetuta zero o una volta
- **\+** → il carattere letterale +
- **\?** → il carattere letterale ?
- **[c1-c2]** → un carattere nell'intervallo [c1, c2]
- **^** → inizio riga
- **\$** → fine riga
- **R1\|R2** → espressione regolare R1 o R2
- **\(REGEX\)** → REGEX viene associata ad un blocco e può essere riferita in seguito
- **\N** → riferimento blocco N (N=1, 2, 3, ...)
- **\w** → un carattere dentro una parola
- **\W** → un carattere non dentro una parola
- **\<** → inizio di una parola
- **\>** → fine di una parola
- **\b** → bordo (inizio o fine parola)
- **\B** → interno (non inizio, non fine parola)
- **[:alnum:]** → carattere alfanumerico
- **[:alpha:]** → carattere alfabetico

- **[[:blank:]]** → carattere di separazione (spazio/tab)
 - **[[:digit:]]** → cifra
 - **[[:lower:]]** → carattere minuscolo
 - **[[:space:]]** → un carattere space (spazio tab newline form fed carriage return ...)
 - **[[:upper:]]** → un carattere alfabetico maiuscolo
 - **{N}** → carattere o espressione regolare precedente ripetuta N volte
 - **{N, }** → carattere o espressione regolare precedente ripetuta almeno N volte
 - **{N,M}** → carattere o espressione regolare precedente ripetuta da N a M volte
- **grep [opzioni] pattern [file]** → individua file a partire dal suo contenuto.
Se file non è presente cerca nella working directory
 - **-n** → stampa il numero di riga in cui è avvenuto un match con il pattern
 - **-H** → stampa il nome del file in cui è avvenuto un match col pattern
 - **-R** → (include -H) effettua una ricerca ricorsiva nel sottoalbero specificato come argomento
 - **-i** → effettua una ricerca case insensitive
 - **--color = yes** → evidenzia i match di colore rosso
 - **-E 'REGEX'** → per introdurre le espressioni regolari.
 - **-o** → stampa esclusivamente la porzione di riga che verifica il match
 - **-v** → inverte il match impostato da linea di comando, stampa tutte le righe che non contengono l'argomento passato.
- **ln[opzioni] TARGET LINK_NAME** → lanciato senza opzioni crea collegamenti fisici dove:
 - TARGET** è il percorso del file originale;
 - LINK_NAME** è il nome del collegamento fisico
 - **-s** → per creare collegamenti simbolici
- **readlink [opzioni] F₁, F₂, ... F_N** → risolve collegamenti simbolici, ovvero se **F_i** è un collegamento simbolico, stampa il file da esso puntato; altrimenti, non viene stampato nulla.
 - **-f** → abilita la risoluzione ricorsiva. Nel caso in cui un collegamento simbolico punti a un altro collegamento simbolico si evita quindi di risolvere ogni collegamento a mano.
- **tar[opzioni] FD₁, FD₂, ... FD_N** → manipola archivi in formato
 - **-f NAME** → per riferirsi all'archivio **NAME**
 - **-c** → per creare un archivio (**tar -c -f NAME FD₁, FD₂, ... FD_N**)
 - **-t** → Per vedere l'elenco dei file e directory in un archivio (**tar -tf NAME**)
 - **-x** → Per estrarre un archivio (**tar -x -f etc.tar**)
 - **-v** → abilita la modalità verbosa, che stampa file e directory manipolati durante una operazione (**tar -v -x -f etc.tar**)

- **-C DIRECTORY** → specifica la directory da cui tar preleva i file (creazione) o in cui tar scrive i file (estrazione) (**tar -C DIRECTORY -f NAME <opzioni>**)
 - **-z** → compressione GZIP (**.gz**)
 - **-j** → compressione BZIP2 (**.bz2**)
 - **-J** → compressione XZ (**.xz**)
- ↓ Meno comprimente
 ↓ Più comprimente
- **7z COMANDO ARCHIVE_NAME FD₁ ... FD_N** → gestisce archivi in questi formati 7-ZIP, TAR, ZIP, RAR
 - **a** → per creare un archivio (**7z a config.7z ~/.config**)
 - **l** → per elencare il contenuto dell'archivio
 - **x** → per estrarre un archivio
 - **-m** → per abilitare le opzioni aggiuntive
 - **-mhe = on** → abilitare la cifratura di un archivio e, tramite l'opzione **-p**, si imposta la chiave simmetrica
7z -mhe=on -ppassword a config.7z ~/.config

● Operatori unari sui file

BASH mette a disposizione tanti operatori unari per la verifica di proprietà su file.

- **-d FILE** VERO se **FILE** esiste ed è una directory
 - **-e FILE** VERO se **FILE** esiste
 - **-f FILE** VERO se **FILE** esiste ed è un file regolare
 - **-h FILE** VERO se **FILE** esiste ed è un link simbolico
 - **-r FILE** VERO se **FILE** esiste ed è leggibile
 - **-w FILE** VERO se **FILE** esiste ed è scrivibile
 - **-x FILE** VERO se **FILE** esiste ed è eseguibile
- test -r /etc/passwd**
echo \$?
0 → /etc/passwd è leggibile

● Operatori di redirectione

STDIN → Descrittore di file 0

STDOUT → Descrittore di file 1

STDERR → Descrittore di file 2

- **< FILENAME** → associa il canale **STDIN** ad un file.
cat < /etc/passwd
grep --color=yes -n -H root < /etc/passwd
- **> FILENAME** → associa il canale **STDOUT** ad un file.
- **2> FILENAME** → associa il canale **STDERR** ad un file. Redirezionando **STDERR** sul file speciale di dispositivo **/dev/null** (ossia un cestino virtuale) si ha uno scarto degli errori.

- **>> FILENAME** → appende l'output di **STDOUT** ad un file
- **2>>FILENAME** → appende l'output di **STDERR** ad un file
- **N>&M** → copia il descrittore di file **N** nel descrittore di file **M**. D'ora in avanti, il canale puntato da **N** scrive sullo stesso file puntato da **M**.
- **2>&1** oppure **&>**: **STDERR** usa lo stesso file di **STDOUT**

- **exec**

Il builtin `exec`, coadiuvato da opportuni operatori di redirectione, apre, redirectiona, chiude file su descrittori arbitrari.

Ad esempio, per aprire il file `/etc/passwd` ed associarlo al descrittore di file 3, si esegue il comando:

```
exec 3< /etc/passwd
```

- Ora è possibile leggere una riga di `/etc/passwd` con l'opzione `-u` del comando interno `read` (che specifica il descrittore di file da cui leggere):

```
read -u 3 line
```
- Si stampa il contenuto della variabile `line`:

```
echo $line
```
- Si ottiene la riga letta:

```
root:x:0:0::/root:/bin/bash
```
- Si può anche aprire un file in scrittura:

```
exec 4> my-output.txt
```
- Si scrive il contenuto della variabile `line` nel file appena aperto:

```
echo $line >&4
```
- Si stampa il file: `cat my-output.txt` Si ottiene il contenuto di `line`:

```
root:x:0:0::/root:/bin/bash
```

Infine, si chiudono entrambi i file aperti: `exec 3<&-` e `exec 4>&-`

- **Espansione di percorsi (pathname expansion).**

- | | |
|--|--|
| ? | → un qualunque carattere |
| * | → una qualunque sequenza di caratteri |
| [C₁C₂ . . .] | → i caratteri C₁, C₂, ... |
| [!C₁C₂ . . .] | → NON i caratteri C₁, C₂, ... |
| [^C₁C₂ . . .] | → NON i caratteri C₁, C₂, ... |
| [C₁-C₂] | → tutti i caratteri da C₁ a C₂ |

- `ls a*` → visualizza i file che iniziano con la lettera `a`
- `ls 'a*'` → visualizza i metadati del file di nome letterale `a*`
- `ls "a*"` → visualizza i metadati del file di nome letterale `a*`

- **find /etc -name "*.conf"** → trova ricorsivamente tutti i file in **/etc** terminanti con l'estensione **.conf**.

Ai fini dell'esecuzione, nell'esempio proposto l'uso di apici singoli (quoting forte) o doppi (quoting debole) è del tutto indifferente.

Il quoting debole espande le variabili, quello forte no.

Qui non sono presenti valori di variabili (ad esempio, **\$var**) nel pattern.

- **find /etc -name *.conf** → esegue una espansione dell'argomento non quotato ***.conf**. L'espansione sostituisce il pattern ***.conf** con il file che lo soddisfa (**a.conf**). Nel caso in cui ci sia anche un file **b.conf** allora BASH segnalerebbe errore poiché il comando eseguito sarebbe **find /etc -name a.conf b.conf** sintatticamente errato

Autenticazione

- **adduser nome_utente** → permette di inserire un nuovo utente
- **deluser nome_utente** → per rimuovere un utente
 - **--remove-home** → rimozione della home directory
- **useradd** → simile a **adduser** ma maggiore difficoltà uso
 - **-m** → crea la home directory
- **userdel** → simile a **deluser** ma maggiore difficoltà uso
 - **-r** → rimuove la home directory
- **usermod [opzioni] nome_di_login** → permette di modificare le proprietà e le risorse di un utente esistente
 - **-l** → cambia il nome di login
 - **-u** → cambia l'identificatore utente
 - **-d** → la nuova home in **/etc/passwd**
 - **-m** → spostare il contenuto della vecchia directory nella nuova
 - **-L** → disabilita login utente, blocca l'account
 - **-U** → riabilita login utente, sblocca l'account
- **chfn** → cambia il nome dell'utente (se eseguito da utente root) e le altre informazioni
- **passwd** → modifica la password
- **chsh** → modifica l'interprete dei comandi

Autorizzazione

- **addgroup nome_gruppo** → per creare un gruppo
- **delgroup nome_gruppo** → per rimuovere un gruppo
- **groupadd / groupdel** → analoghi ai comandi precedenti
- **groupmod [opzioni] nome gruppo** → modifica le proprietà e le risorse di un gruppo esistente
 - **-g** → cambia l'identificatore gruppo
 - **-n** → cambia il nome del gruppo
- **groups [nome_di_login]** → mostra i gruppi di appartenenza di un utente
 - se non specifico parametri mostra i gruppi dell'user attualmente in uso
- **usermod** → per modificare gruppi di lavori primari e secondari
 - **-g nome_gruppo** → specifica il nuovo gruppo di lavoro primario
 - **-G ng_1,ng_2,...** → specifica il nuovo insieme completo dei gruppi secondari (NON LAVORA IN APPEND)

- **-a** → da aggiungere all'opzione **-G** se vogliamo aggiungere i gruppi secondari
- IN APPEND
- **gpasswd** → aggiunge di default gruppi senza troncarli
 - **-a nome_di_login gruppo** → si aggiunge un utente ad un gruppo
 - **-d nome_di_login gruppo** → si rimuove un utente da un gruppo
- **newgrp gruppo** → soluzione poco "pulita" per non aspettare il reboot della macchina per rendere effettive le modifiche sui gruppi
- **chown [opzioni] nome_di_login file** → modifica l'utente creatore di un file e/o di una directory
 - **-R** → opera ricorsivamente su una cartella
 - **chown utente_creatore:gruppo_creatore file** → permette di modificare utente e gruppo creatore di un file in un colpo solo
- **chgrp [opzioni] nome_gruppo file** → modifica il gruppo di un file e/o di una directory
 - **-R** → opera ricorsivamente su una cartella
- **chmod [opzioni] permessi file** → modifica i permessi del file
 - **u, g, o, a** → entità su cui applicare i permessi (utente creatore, gruppo del file, utenti restanti, tutti gli utenti)
 - **r, w, x, s** → permessi che possiamo applicare sulle entità precedenti (lettura, scrittura, esecuzione/accesso a cartella, applicazione bit SETUID/SETGID)
 - **+** → per aggiungere il permesso
 - **-** → per rimuovere il permesso
 - accetta anche una sequenza di 4 numeri al posto dei metodi sopracitati
 - LAVORA IN APPEND

File System

- **/dev/[tipo]d[id_disco]** → formato file speciali dei dischi
 - Tipo: h (dischi IDE), s (dischi SCSI o SATA).
 - ID disco: una lettera univoca associata al disco, partendo dalla a.
 - ID partizione: un numero intero univoco associato al disco, partendo dall'1.
- **fdisk [opzioni] disco_rigido** → **fdisk (/sbin/fdisk)** è un comando che gestisce la tabella delle partizioni di un disco rigido (disco rigido viene inserito mediante virtual box). Si occupa di partizionare il disco. Dopo averlo lanciato si presenta un menù.
Inserendo il carattere **m** si presenta una lista dei possibili comandi eseguibili.
 - **p** → stampa la descrizione del disco
 - **n** → si aggiunge una nuova partizione
 - **t** → gestisce i tipi di partizione
 - **L** → si vedono tutti i tipi di partizione
 Per salvare le modifiche **wq**
- **ls -l /sbin/mkfs.*** → individua i file system disponibili nel sistema (**ext2, ext3, ext4**)
- **mkfs -t tipo_fs file_speciale_dispositivo** → viene creato un file system di **tipo_fs** in file_speciale_dispositivo (**mkfs -t ext3 /dev/sdb1**)
- **mount -o options -t fs_type special_file mount_point** → Comando che associa alla directory **mount_point** un file system di tipo **fs_type** contenuto nella partizione **special_file**. Se non si specificano opzioni e tipo, mount prova a

scoprire il tipo del file system e a montarlo con opzioni di default (read-write, esecuzione).

- **lsblk** → visualizza i file system memorizzati su supporti fisici o file con annessi mount point
- **mountpoint [directory]** → controlla se directory sia o meno un mountpoint
- **umount [special_file | mount_point]** → Il file system può essere staccato dal suo punto di aggancio tramite l'operazione di sgancio, sostanzialmente l'inversa di mount specificando il file system oppure il mount point.
- **fsck disk** → comando wrapper che invoca il comando giusto di controllo della consistenza e riparazione. Il controllo avviene al boot automaticamente oppure manualmente con possibilità di riparazione.

Processi

- **pidof [nome]** → stampa il PID di tutti i processi attivati a partire da un nome esatto
- **pgrep '[regex]'** → stampa i PID di tutti i processi attivati a partire da una espressione regolare estesa
 - **pgrep -n bash** → stampa il PID del processo più recente contenente la stringa "bash" nel nome
 - **pgrep -o bash** → stampa il PID del processo meno recente contenente la stringa "bash" nel nome
 - **pgrep -l bash** → stampa il PID e il nome dei processi contenenti la stringa "bash" nel nome
 - **pgrep -u username** → stampa il PID di tutti i processi lanciati dall'utente username
- **pstree** → stampa una rappresentazione compatta dell'albero dei processi
 - **pstree | less -Mr** → necessario per impaginare l'output a causa della sua grande dimensione
 - **-a** → stampa i nomi dei processi in esecuzione ed i relativi argomenti
 - **-c** → disabilita la "compattazione", esplicitando la rappresentazione dell'albero
 - **-p** → mostra i PID dei processi
 - **-H PID** → evidenzia in neretto il ramo dell'albero dal processo init al processo identificato da PID. Evidenzia quindi un processo specifico e rispettivi antenati. Utile all'amministratore per capire come è stato generato un processo
 - **-s** → in alternativa a **-H** si può utilizzare l'opzione -s per visualizzare solo la catena di processi
- **/bin/kill [options] PID...** → invia segnali ai processi in base al PID
 - se non si specifica nulla il segnale di default è SIGTERM
 - **-l** → stampa elenco dei segnali disponibili
 - **-L** → come il comando precedente ma con descrizioni più esaustive
- **/usr/bin/pkill** → può selezionare il processo a cui inviare il segnale tramite il suo nome o tramite filtri di ricerca

- **pkill --signal SIGNAL NOME** / **pkill -SIGNAL NOME** → invia il segnale SIGNAL ad un processo identificato da NOME
 - **-n** → permette di identificare il processo più recente
- **Operatore |** → implementa il meccanismo della pipe concatenando i comandi
 - questo operatore lascia passare solo lo STDOUT
- **tee** → introduce un meccanismo di tubazione dove lo STDIN è scritto su STDOUT e su un file
 - **C1 | tee out1.txt | C2 | tee out2.txt | ...**
 - utile per verificare correttezza output intermedi
- **mkfifo FILENAME** → crea un file speciale per una named pipe
- **wget** → client HTTP per il download di brani (pacchetto software wget)
- **mpv** → riproduttore multimediale "streaming" (pacchetto software mpv)
- **stty tostop** → blocca l'accesso all'STDOUT ai processi in background
- **jobs** → elenca i job lanciati
 - **-l** → stampa i PID di tutti i processi della pipeline
- **Operatore &** → se inserito al termine di un comando esso verrà eseguito in background
 - es. **sleep 7200 &**
- **fg JOB** → permette di ripristinare in foreground un certo JOB specificato
 - i job possono essere rappresentati nei seguenti modi: %JOB_ID, %COMANDO, %+ ◦ %, %-
 - es. per ripristinare il job numero uno: **fg %1**
- **Operatore \$(COMMAND)** → introduce l'espansione di comando, viene eseguito COMMAND e il suo output è sostituito all'espressione \$(COMMAND)
- **Operatori <(COMMAND) e >(COMMAND)** → introducono la sostituzione di processo, viene eseguito COMMAND e il suo input/output finisce in /dev/fd/N (file temporaneo, N intero)
 - **<(COMMAND)** → l'output di COMMAND viene fatto vedere come un file al comando che invoca <()
 - **>(COMMAND)** → l'output scritto sul file /dev/fd/N dal comando che usa >() viene fatto vedere come input (STDIN) a COMMAND
- **exec [opzioni] COMANDO** → consente di sostituire l'immagine del processo corrente con quella di un nuovo comando
- **Fork bomb** → **:() { :|:& };;**

Elevazione privilegi

- **id [opzioni] [utente]** → stampa gli identificatori utente, di gruppo primario e dei gruppi secondari per uno specifico utente. Senza argomenti considera l'utente attivo sul terminale.
- **su [opzioni] [nome_utente]** → esegue comandi generici con credenziali di un altro utente. Se lanciato senza opzioni e argomenti ci riferisce all'utente root e all'interprete di default /bin/bash
 - **su [nome_utente]** → comando per diventare l'utente nome_utente

- **su - login, su -l, su -, su - prova** → comando per diventare prova e caricare il suo ambiente
 - **-c** → lancia un comando specifico (mettere `` se ci sono più argomenti)
- **sudo** → esegue comandi generici con le credenziali dell'utente di partenza (ricevendo un solo argomento il comando viene eseguito da root). Configurazione contenuta nel file **/etc/sudoers** (% indica un gruppo) **<who> <where> = <as whom> <what>** .
 - **-l** → specifica cosa può fare l'utente attuale (**-ll** output più prolisso)
 - **-U** → specifica il nome di login dell'utente di cui si vogliono ottenere i privilegi di lancio
 - **-g** → specifica il nome del gruppo primario sotto cui si eseguirà il comando
 - **-u** → permette di specificare il nome di login dell'utente che eseguirà il comando
 - **Defaults timestamp_timeout=5** → da inserire nel file **/etc/sudoers** per definire ogni quanto mettere la password
 - **NOPASSWD: ALL** → etichetta che definisce quali comandi richiedono password.
- **getcap [opzioni] file** → mostra le capability di un file
- **setcap [opzioni] capability file** → setta capability in file

Gestione software in Debian (merda)

- **apt install PACCHETTO1 PACCHETTO2 ...** → utilizzato per installare pacchetti
- **apt update** → per aggiornare l'elenco dei repository
- **apt remove PACCHETTO1 PACCHETTO2 ...** → per rimuovere i pacchetti
 - **--purge** → cancella anche tutti i file di configurazione generati dagli script eseguiti dal pacchetto binario durante l'installazione (se non ancora rimossi)
- **apt purge PACCHETTO1 PACCHETTO2** → esegue le stesse operazioni di apt remove --purge
- **apt autoremove --purge PACCHETTO1 ...** → rimuove il pacchetto, la sua configurazione e tutti i pacchetti servivano solo per installarlo ed eseguirlo (con le relative configurazioni)
- **du** → "disk usage", dice quanto occupano i file/directory all'interno del disco
 - **-h** → stampa human readable
 - **-s** → consumo totale
- **apt clean** → rimuove i file *.deb dalla directory cache del sistema APT
- **apt upgrade** → Per aggiornare tutti i pacchetti installati
 - **-u** → forza la stampa dell'elenco dei pacchetti da aggiornare
- **apt full-upgrade/apt dist-upgrade** → per aggiornare ad una nuova distribuzione
- **apt search REGEX1 REGEX2 ...** → ricerca un pacchetto che soddisfa le regex specificate, stampa nome e descrizione breve del pacchetto
 - **--names-only** → esclude la descrizione breve dal pacchetto del parco stringhe da controllare
- **apt show PACCHETTO1 PACCHETTO2...** → stampa i metadati dei pacchetti specificati
- **apt depends PACCHETTO1 PACCHETTO2...** → stampa le dipendenze dirette dei pacchetti specificati

- **apt rdepends PACCHETTO1 PACCHETTO2...** → stampa le dipendenze inverse dei pacchetti specificati
- **CALCOLO E DISEGNO GRAFO DELLE DIPENDENZE**
 - **debtree** → calcola il grafo delle dipendenze di un pacchetto
 - **dot** → disegna grafi
 - **inkscape** → per visualizzare l'immagine prodotta
- **debtree --no-recommends --no-conflicts bash > bash.dot**
dot -Tsvg bash.dot > bash.svg
inkscape bash.svg
- **apt-file** → permette di effettuare ricerche più sofisticate sui pacchetti (funziona anche su pacchetti non installati)
 - **list -x [pacchetto]** → ricerca dei file forniti da un pacchetto software, tramite regex
 - **search -x [file]** → per capire quale pacchetto fornisce un file specifico, tramite regex
- **update-alternatives** → elenca tutti i pacchetti virtuali e l'alternativa di default corrispondente, elenca tutte le alternative di un pacchetto virtuale, seleziona come default una fra le alternative disponibili
 - **--get-selections** → elenca i nomi di tutti i pacchetti virtuali disponibili sul SO Debian
 - **--list PACCHETTO** → elenca tutti i pacchetti alternativi già installati per un dato pacchetto virtuale sul SO Debian
 - **--config PACCHETTO** → permette di modificare l'alternativa di default attuale del pacchetto
- **debtree --no-recommends --no-conflicts bash > bash.dot**: produce un file testuale rappresentante il grafo delle dipendenze di bash
- **dot -Tsvg bash.dot > bash.svg**: produce un'immagine vettoriale mostrante il grafo
- **inkscape bash.svg**: visualizza l'immagine vettoriale