COMANDO	FUNZIONE
date	stampa la data
who	stampa la lista degli account collegati al server
who | wc	stampa il numero di account connessi al server
pwd	stampa la CWD (Current Working Directory, ovvero la directory in cui siamo attivi al momento)
clear	ripulisce lo schermo e porta in cima alla pagina
CTRL-C	forza l'uscita da un comando
cd	porta alla propria home
cd ~	porta alla propria home
cd /	porta alla Root
cd -	porta alla directory precedente
cd .	porta alla CWD
cd ..	porta alla directory superiore rispetto a quella in cui siamo (Parent Directory)
cd ../..	porta nella directory due livelli sopra alla corrente
cd /dir1/dir2/dir3	porta nella directory dir3 a partire dalla Root (Absolute Path)
cd dir3	porta nella directory dir3 a partire dalla CWD (solo se esiste) (Relative Path)
mkdir mydir	crea una nuova directory denominta mydir nella CWD
rmdir mydir	elimina la directory mydir (la directory deve essere vuota)
touch myfile	crea un nuovo file vuoto denominato myfile
rm myfile	elimina (definitivamente) il file myfile
mv myfile1 myfile2	rinomina il file o la directory myfile1 in myfile2
mv myfile mydir	sposta il file myfile nella directory mydir
mv myfile1 mydir/myfile2	sposta il file myfile1 nella directory mydir e lo rinomina in myfile2
ls	stampa la lista dei file presenti nella CWD
ls -a	stampa la lista di tutti i file (compresi quelli che iniziano col punto)
ls -l	stampa la lista dei file presenti nella CWD in long format
cat myfile	stampa il contenuto del file myfile
cat	ripetere quello che si digita fino a che non si preme CTRL+C
ls myfile	risponde se esiste il file myfile oppure no

cp mydir1/myfile mydir2	copia il file myfile dalla directory mydir1 in mydir2
man mycommand	stampa il manuale del comando mycommand
q	(Quit) esce dal manuale
\	(Escape) disabilita il valore di metacarattere al testo che segue immediatamente (es. per scrivere lo spazio si usa "\ ", escluse le virgolette)
history	stampa la lista di comandi dati in precedenza
!numcomand	esegue il comando relativo al numero della lista di history
echo txt	ripete il testo digitato
echo $SHELL	stampa il nome dello Shell in uso
bash -v	stampa la versione di Bash
which mycommand	stampa che tipo di file è il comando mycommand
whoami	stampa chi sono io in veste di proprietario di file (User)
nano myfile	apre il file myfile con un editor di testo da terminale
chmod g+w myfile	aggiunge il privilegio di scrittura (w) al gruppo (g) al file myfile
chmod u+w myfile	aggiunge il privilegio di scrittura (w) al proprietario (u) al file myfile
chmod g-rw myfile	toglie il privilegio di lettura (r) e scrittura (w) al file myfile al gruppo (g)
chmod g+r,o-r myfile	aggiunge il privilegio di lettura (r) al gruppo (g) e toglie quello di lettura (r) a tutti gli altri (o) al file myfile
echo $PATH	stampa le directory dalle quali il sistema prende le utility (comandi)
chmod 700 myfile	rende eseguibile il file myfile
./myfile	esegue il file myfile precedentemente reso eseguibile (qualsiasi file non presente nelle PATH, ovvero uno script)
wc myfile	stampa in ordine il numero di linee, di parole e di caratteri del file myfile
wc -l myfile	stampa solo il numero di linee del file myfile
cut -fn -dchar	taglia il campo numero n usando come delimitatore di campi il carattere digitato dopo -d (es. -d' ' usa lo spazio come delimitatore di campi)
env	stampa la lista di tutte le variabili istanziate
MYVAR=myvalue	istanzia la nuova variabile MYVAR col valore myvalue
echo $MYVAR	stampa il valore della variabile MYVAR
echo !!	esegue l'ultimo comando eseguito
scp myfile myserver:mydir	copia il file myfile nella directory mydir della macchina connessa al server myserver
scp myserver1:myfile myserver2:mydir	copia il file myfile dal server myserver1 nella directory mydir del server

	myserver2
echo $((4+5))	(Arithmetic Expansion) esegue il calcolo aritmetico digitato e lo stampa
MYVAR=$(mycommand)	(Command Substitution) avvalora la variabile MYVAR con l'output del comando mycommand
read MYVAR	salva l'input da tastiera nella variabile MYVAR
# mycomment	rende il testo mycomment solo commento
$n	in uno script indica la parola n-esima immessa da tastiera subito dopo il richiamo dello script (es. "./myscript.sh mydir1 mydir2", $1=mydir1 e $2=mydir2)
top	stampa l'elenco dinamico dei processi in esecuzione sulla macchina
ps	stampa l'elenco statico dei processi in esecuzione sulla macchina
mycommand | less	permette di scorrere l'output del comando mycommand con la tastiera (premere Q per uscire)
id	stampa lo user id
ls -la $(which ps)	si nota che al posto della r nei permessi ha una s (se c'è la s il proprietario del processo continua a essere il proprietario del file)
$?	stampa l'Exit Status dell'ultimo comando eseguito (0=true, 1=false)
mycommand1; mycommand2; ...	lista di comandi (l'Exit Status si riferisce sempre all'ultimo)
'text'	disabilita tutti i metacaratteri all'interno (Strong Quoting)
"text"	disabilita i metacaratteri all'interno tranne alcuni tipo $ (Weak Quoting)
test mycondition	restituisce un Exit Status positivo (0) se la condizione è vera oppure negativo (1) viceversa (es. "test 11 > 9" restituisce un Exit Status negativo poichè l'operatore ">" opera alfabeticamente, mentre "test 11 -gt 9" restituisce un Exit Status positivo poichè "-gt", ovvero greater than, opera algebricamente)
test A -eq B	vero se A algebricamente uguale a B (equal)
test A -ne B	vero se A algebricamente non uguale a B (not equal)
test A -gt B	vero se A algebricamente maggiore di B (greater than)
test A -ge B	vero se A algebricamente maggiore o uguale di B (greater or equal)
test A -lt B	vero se A algebricamente minore di B (less than)
test A -le B	vero se A algebricamente minore o uguale di B (less or equal)
[mycondition]	ha lo stesso effetto di test
stat myfile	stampa alcune proprietà del file myfile (solo su Linux)
stat -x myfile	stampa alcune proprietà del file myfile (solo su Mac OS)
true	restituisce un Exit Status positivo (0)

false	restituisce un Exit Status negativo (1)
mycommand; sleep n	disabilita l'esecuzione del comando mycommand per n secondi subito dopo il primo avvio
\	scritto durante un comando ti permettere di continuare a scrivere il comando alla riga successiva
ls -i myfile	stampa l'inode number del file myfile
tty	stampa il nome del terminale che stiamo utilizzando
file *	stampa le proprietà di tutti i file presenti nella CWD
mycommand 2> myfile	reindirizza lo Standard Error al file myfile
mycommand 2> /dev/null	annulla l'output di errore
/dev/random > myfile	crea bit a caso nel file myfile
head myfile	stampa le prime 10 righe del file myfile
head –n myfile	stampa le prime n righe del file myfile
tail myfile	stampa le ultime 10 righe del file myfile
tail -n myfile	stampa le ultime n righe del file myfile
head -n myfile | tail -1	stampa la n-esima righa del file myfile
cat -n myfile	stampa il contenuto del file myfile con il numero di riga
tail -f myfile	lascia in sospeso la stampa del contenuto del file myfile e la aggiorna se si salva altro
cat >> myfile	permette di scrivere dati nel file myfile (Append)
grep mychar myfile	stampa tutte le righe del file myfile che contengono il carattere mychar
/usr/share/dict/words	file che contiene le parole inglesi del sistema
basename myfile	stampa l'ultimo ramo della path di myfile
dirname myfile	stampa la directory che contiene il file myfile
ls -id mydir	stampa l'inode della directory mydir
stat -f%i myfile	stampa l'inode del file myfile (qulasiasi) su Mac OS
stat -c%i myfile	stampa l'inode del file myfile (qualsiasi) su Linux
stat -f%z myfile	stampa la dimensione del file myfile su Mac OS
stat -c%s myfile	stampa la dimensione del file myfile su Linux
find mydirlist -name myfile -print	cerca i file che si chiamano myfile nelle directory mydirlist e li stampa
find mydirlist -name myfile* -print	cerca i file il cui nome comincia con myfile e li stampa
find mydirlist -name myfile -type d -print	cerca il file myfile di tipo directory (d) e lo stampa
find mydirlist -name myfile \(-type d -o -type f \)	cerca sia directory che regular file
find mydirlist -name myfile -exec mycommand {} \;	cerca ed esegue il comando mycommand sui file trovati

sort	ordina in ordine alfabetico
sort -n	ordina in ordine numerico crescente
sort -nr	ordina in ordine numerico decrescente
tr mychar1 mychar2	trasforma tutti i caratteri char1 in char2
\n	indica a capo
grep -r mydir	esegue il grep ricorsivamente cioè anche all'interno di directory interne a mydir
grep -E mydir	esegue il grep nella modalità estesa
grep '[abc]' mydir	prende le parole che contengono almeno una lettera tra la a, la b e la c
grep '[abc]..[yz]' mydir	prende le parole che contengono almeno una lettera tra abc poi subito a seguire due lettere qualsiasi e poi una tra yz
grep '[^ab]' mydir	prende le parole che contengono tutte le lettere tranne ab
grep '^a...z$' mydir	prende le parole la cui prima lettera è la a e l'ultima è la z
MYVAR=$MYVAR\mychar	aggiorna la variabile MYVAR con il testo digitato dopo il backslash accodandolo al testo già presente nella variabile (se presente)
MYVAR=${n:-myval}	istanzia la variabile MYVAR con il valore digitato nel campo $n, se $n è vuoto la istanzia con il valore di default myval
MYVAR=${MYVAR:=myval}	istanzia la variabile MYVAR con il valore myval
echo ${MYVAR:n}	stampa i caratteri successivi ai primi n (Offset) della variabile MYVAR
echo ${#MYVAR}	stampa la lunghezza della variabile MYVAR
echo ${MYVAR/mychar1/mychar2}	stampa la variabile MYVAR con il primo match del carattere mychar1 sostituito da mycahr2
date +%myopt1-%myopt2-%myopt3	stampa la data nel formato myopt1-myopt2-myopt3
date +%myopt1/%myopt2/%myopt3	stampa la data nel formato myopt1/myopt2/myopt3 (Si può usare un carattere qualsiasi per delimitatore al posto dei trattini o dello slash, basta sostituirlo nel codice)
%Y	anno
%m	mese
%d	giorno
%H	ora
%M	minuti
%S	secondi
%s	secondi dalla EPOCH (1 Gennaio 1970)
cal myyear	stampa il calendario dell'anno myyear (es. cal 1991)
mycommand &	lascia sospeso il comando in background una volta avviato

jobs	stampa la lista di processi in background
CTRL+Z	avvia il processo attuale in background
fg %n	avvia in foreground il processo n
bg %n	avvia in background il processo n
PS1	variabile che contiene il testo del prompt del terminale
PS1='\t ->'	così settata stampa un prompt di questo tipo (hh:mm:ss ->)
alias mycommand2=mycommand1	sostituisce il nome del comando mycommand1 con il nuovo nome mycommand2
open myfile –a myapp	apre il file myfile con l'applicazione myapp
md5 myfile	applica un algoritmo al file myfile che restituisce un digest di 32 caratteri (solo su Mac OS)
md5sum myfile	stesso effetto su Linux
diff myfile1 myfile2	stampa le differenze tra i file (non stampa nulla se non trova differenze)
type mycommand	stampa l'alias del comando mycommand
xargs	funziona come il pipe, cioè esegue un comando all'output di un altro comando
locale	stampa l'encoding utilizzato dal terminale
openssl dgst -sha1 myfile	crea il digest sha1 del file myfile
openssl enc -des3 < myfile	cripta il file myfile con il cipher des3
vmmap mypid	stampa la gestione della memoria del processo col pid mypid
ps -p mypid	mostra le caratteristiche del pid mypid
hexdump myfile	stampa i caratteri presenti nel file nel formato ASCII (codice ASCII)
hexdump -c myfile	stampa i caratteri presenti nel file nel formato ASCII (carattere ASCII)
iconv -f myenc1 –t myenc2 myfile	converte il file myfile dall'encoding myenc1 a quello myenc2
iconv -f L1 -t UTF8 myfile	converte il file myfile dall'encoding latin iso 1 a utf-8
iconv -l	stampa tutti gli encoding disponibili
tr -d ' '	elimina (d) tutti gli spazi (' ')
unset MYVAR	elimina qualsiasi predefinizione della variabile MYVAR
echo ${#myarray[*]}	stampa la lunghezza dell'array myarray (solo se definito)
myarray=(myval1 myval2 ...)	avvalora l'array myarray con i valori digitati in seguenza
ps	stampa i processi legati ai terminali
ps -e	stampa tutti i processi
ps -ef	stampa tutti i processi in formato lungo

pstree	stampa tutti i processi in un grafico ad albero (solo su Linux)
echo $$	stampa il Process id (PID) di Bash
passwd	permette di cambiare la password del proprio sistema
ps -e -ocomm	stampa solo il campo col nome dei processi
ps -e -opid,comm	stampa solo i campi pid e nome del processo
ps -e -opid=,comm=	stesso effetto ma non stampa la prima linea di intestazione con i titoli
ps -e -opid=mytit1,comm=mytit2	da come titoli ai campi pid e comm rispettivamente mytit1 e mytit2
ps -e -opid,ruid,uid,comm	stampa il pid, il real uid, l'effective uid e il nome del processo
[$# -ne n]	test che verifica se il numero di tutti gli inserimenti sulla riga dello script è uguale a un valore di default n
[-z $MYVAR]	test che verifica se la variabile MYVAR è uguale a zero
exec mycommand	eseguito in Bash dice a Bash di trasformare lui stesso in quel comando mentre lanciando il comando normalmente, Bash si duplica e il Bash Child si trasforma nel comando lanciato
tr [:punct:] '\n'	sostituisce tutti i segni di punteggiatura con un accapo
uniq -c	stampa il numero di ripetizioni delle righe uguali
lynx myurl	browser web a linea di comando
export MYVAR	esporta la variabile MYVAR in modo da poter essere letta anche all'interno di script
gzip myfile	comprime il file regolare myfile in un file zippato (myfile.gz)
gunzip myfile.gz	decomprime il file myfile.gz
tar myfile	comprime il file myfile (anche directory) in un file tar (myfile.tar)
tar cvfz myfile2.tgz myfile1	comprime il file myfile1 in un file tar zippato (myfile2.tgz)
exec mycommand 1>myfile	invia l'output del comando mycommand al file myfile
readlink mylink	stampa il source file del link mylink
SIGINT	CTRL+C
SIGTSTP	CTRL+Z
SIGQUIT	CTRL+\
SIGSTOP	stoppa il processo
SIGCONT	riprende il processo stoppato
SIGKILL	termina il processo (non gestibile)
trap -l	stampa la lista di segnali disponibili
trap mycommand mysignal	indica a Bash di eseguire il comando mycommand se riceve il segnale mysignal
trap - mysignal	reimposta l'azione di default alla ricezione del segnale mysignal

trap '' mysignal	indica a Bash di non eseguire niente alla ricezione del segnale mysegnal
trap -p mysignal	stampa quale operazione è associata al segnale mysignal
trap -p	stampa la lista di attribuzioni comando-segnale
ulimit -a	stampa la dimensione di memoria disponibile per tutti i file
ulimit -c	setta la dimensione del file core
kill -mysignal mypid	invia il segnale mysignal al processo associato al pid mypid
kill -9 mypid	uccide il processo associato al pid mypid, stesso effetto di kill -SIGKILL mypid
kill -SIGSTOP mypid	stoppa il processo con il pid mypid
kill -SIGCONT mypid	riprende il processo con il pid mypid
mdfind mychar	corrispettivo di spotlight a linea di comando, cerca il carattere mychar nel contenuto dei file nel Hardisk
mdfind -0 mychar	separa i risultati con un null
mdls myfile	stampa le proprietà del file myfile
strings myfile	stampa le stringhe stampabili del file binario myfile
mdfind 'mycriterion1<n && mycriterion2>m'	cerca i file con quei criteri (es. mdfind -0 'mycriterion<n' | xargs -0 mdls -name mycriterion)
tee myfile	all'interno di una linea di comando devia lo standard output al file myfile (es. ps -ef | tee myfile | grep mychar) invia l'output del ps al file myfile e al grep
jot n x y	stampa n repliche di numeri da x a y (numeri)
jot -r -c n x y	stampa n volte i caratteri (c) casuali (r) da x a y (caratteri)
rs r c	(Reshape) stampa lo standard input in r righe e c colonne (r=0 stampa fino alla fine dell'input)
rs -g r c	non separa le colonne con lo spazio
dd ibs=n obs=m count=i skip=j if=myfile1 of=myfile2	copia il file myfile1 in myfile2
ibs	input block size
obs	outout block size
count	caratteri da copiare
skip	caratteri tagliati dall'inizio del file
if	input file
of	output file
ls [^az]*	stampa l'output di ls di tutte le righe che non cominciano per a e finiscono per z
tr -d -C [:alpha:]	elimina (d) tutto ciò che "non è" (C) alfabeto
tr \\064 mychar	sostituisce il carattere con codice ottale \064 (4) con mychar

tr -d \\000-\\037	elimina tutti i caratteri di controllo da \000 a \037
curl myurl	restituisce il codice dell'url myurl
ping myip	manda un pacchetto icmp echorequest e riceve un pacchetto icmp echorequire all'ip myip
ping -c n myip	invia n pacchetti e poi termina
fuser myfile	stampa il pid del processo che sta usando il file myfile
lsof -p mypid	stampa la lista dei file che il processo associato al pid mypid ha aperti
diff myfile1 myfile2 > mypatch	crea il file patch con le differenze
patch myfile mypatch	aggiorna il file myfile aggiungendo le differenze salvate nel file mypatch
killall mycommand	chiude tutte le istanze del comando mycommand (es. killall nano)
mkfifo myfile	crea un file fifo (first input first output) (named pipes)
whois myip	stampa informazioni sull'indirizzo ip myip
netstat -f inet	stampa le connessioni aperte nella propria rete inet
netstat -f inet -p tcp	stampa le connessioni tcp aperte nella propria rete
netstat -b -i en0	visualizza i byte scambiati dalle connessioni aperte nella rete en0
ifconfig -a	visualizza tutti i driver di rete con i relativi indirizzi
source	stesso effetto di include in C
gcc	compilatore di codice C e C++ (idem cc)
gcc myfile.c	compila il file sorgente myfile.c in un eseguibile nominato a.out
gcc myfile.c -o myexec	compila il file sorgente myfile.c in un eseguibile nominato myexec
/usr/include	directory contenente le librerie del C (proptotipi)
/usr/lib	(implementazioni)
gcc -H myfile.c	stampa la lista delle librerie incluse nel file sorgente myfile.c ad albero
gcc -Wall myfile.c	stampa tutti (all) i warning (W), precompilazione
wget --recursive --no-clobber --page-requisites --html-extension --convert-links --restrict-file-names=windows --domains example.com --no-parent www.example.com/folder1/folder2/	scarica l'intero sito www.example.com/folder1/folder2/ nella cwd

Dichiarazioni condizionali e sintassi dei cicli iterativi

if

if command1; then command2 ; fi
if list1; then list2; fi
if list1; then list2; else list3; fi
if list1; then list2; elif list3; then list4; fi

while

while list1; do list2; done
until list1; do list2; done

for

for var in parameterlist; do list; done
for ((v=p1; v < p2; v++)); do list; done