

# Linux Base the CLI

Franco Arcieri

## 1 ssh

Accesso remoto da terminale.

Per verificare se il servizio è attivo: sudo systemctl status sshd.service

- Verificare l'esistenza della cartella HOME/.ssh
- nella cartella deve esserci almeno un file. Se non c'e' nulla allora da remoto nessuno puo' collegarsi alla macchina linux
- tentativo di connessione da remoto: uno volta trovato l'indirizzo ip della macchina linux, provare, da remoto, a eseguire: ssh *user@10.211.55.128*
- se non va occorre abilitare sshd: apt list | grep openssh
- sudo apt install openssh-server
- ma occorre sempre fornire la password e i terminali non si aprono
- allora utilizzare ssh -Y *user@10.211.55.128*
- per creare una chiave rsa nel client, ssh –keygen rsa. copiare la chiave pubblica (.pub) e copiarla nel file authorized\_keys della macchina server alla quale voglio connettermi

Ricapitoliamo: per collegarsi da remoto senza utilizzare password:

- sulla macchina client occorre avere una coppia di chiavi (esempio: id\_rsa e id\_rsa.pub)
- appendere id\_rsa.pub al file authorized\_keys presente nella cartella HOME/.sshd
- potrebbe avere bisogno di rivviare il servizio sshd
- Attenzione: controlli bene le proprietà del file, devono essere rw.rw.r authorized\_keys

Grazie a questa configurazione è possibile eliminare completamente l'accesso tramite password. In tal modo, anche la macchina è esposta su internet, nessuno potrà accedere a meno che non abbia la chiave privata e pubblica che è stata registrata un authorized\_keys

- andare nella cartella /etc/ssh: pushd /etc/ssh/
- aprire sshd\_config
- PubkeyAuthentication yes
- PasswordAuthentication no

Attenzione. Come funziona realmente SSH?

esempio: ssh user@192.168.213.134

macchina locale(192.168.213.133), macchina remota (192.168.213.134) Le cose marcate con 1,2,3,4,) sono eseguite sulla macchina locale Lo cose marcate con a,b,c,d,e) sono eseguite sulla macchina remota

1. SSH si collega a SSHD (porta 22) della macchina remota
  - a) SSHD sulla macchina remota mette in esecuzione il programma /bin/bash

2. Lo stdin (tutte le pressioni di tasti oppure le redirect fatte verso ssh - esempio cat file & ssh ...) sono lette dal programma ssh e inviate al programma sshd della macchina remota  
b) tutti i caratteri ricevuti via rete da SSHD (e provenienti da ssh del client (133) sono inviati al programma messo in esecuzione (in questo caso bash) il risultato di bash (lo stdout della macchina remota) viene letto da SSHD e viene rimandato al client ssh della macchina 133
3. tutti i caratteri ricevuti da SSHD (provenienti dalla macchina remota) sono messi da ssh sullo stdout (quindi a schermo oppure, se c'è una redirect, su file)

## 2 bash

Chiedo il manuale di bash: man bash.

Tramite bash si richiamano i comandi del sistema operativo. È la console di sistema. Quante console differenti ci sono in linux?

- sh: è la console classica. con poche estensioni
- bash: è la console più utilizzata, ricca di estensioni. è quella che esegue .bashrc e .profile
- csh
- zsh
- e tante altre

bash non ha programmi propri ma ha un grosso insieme di istruzioni di controllo. I comandi, in linux, sono nelle cartelle /bin, /usr/bin, /sbin, /usr/sbin. e poi in altre cartelle ma queste sono le più importanti.

Cosa è /sbin. È la cartella che contiene alcuni comandi che sono "linkati" in modo statico, non fanno cioè riferimento a librerie dinamiche. Sono utili quando sta configurando un sistema linux e non ha ancora installato le librerie dinamiche.

Dove sta un particolare comando? whereis echo, per trovare dove sta il comando echo  
echo -n : non va a capo

## 3 i comandi di linux

Tutti i comandi di linux sono fatti per comportarsi in modo atomico, nel senso che non fanno più di quello che gli viene chiesto. Esempio: comando DIR del dos.

```
Volume in drive C has no label.
Volume Serial Number is 2C7D-7820

Directory of C:\Users

09/26/2020  11:34 AM    <DIR>      .
09/26/2020  11:34 AM    <DIR>      ..
09/02/2020  07:07 PM          1, 000 applese
09/24/2020  08:59 PM    <DIR>      Public
10/20/2020  06:39 PM    <DIR>      Soap
                           1 File(s)       1, 000 bytes
                           4 Dir(s)   13, 879, 459, 840 bytes free
```

Il comando dir mostra molto di più di quello che gli viene chiesto. Mostra il nome del disco, il numero dei file, le date, ... Il comando DIR non è atomico.

Invece il comando ls è atomico: riporta esclusivamente l'elenco dei file e delle cartelle presenti nella cartella corrente.

l'atomicità mi consente di comporre facilmente comandi, concatenando l'uscita dell'uno con l'ingresso dell'altro

### 3.1 cat

man cat

## 3.2 stdin, stdout, stderr

tutti i comandi lavorano su tre flussi base di I/O: stdin, stdout, stderr.

Esempio cat senza parametri legge dallo stdin e scrive sullo stdout.

CTRL-D e' l'EOF dello stdin

### 3.2.1 Esercizio

usando cat, scriva ciao su un file.

```
cat > pippo.txt
```

```
ciao
```

```
CTRL-D
```

Il simbolo > indica che l'uscita del comando(nel nostro caso cat) deve essere rediretta sul file indicato dopo il simbolo >

cioe' lo stdout del programma cat diventa il file

Ricapitoliamo: cat legge da stdin e scrive su stdout. Con i simboli < e > noi sostituiamo stdin e stdout con il file infidicato al fianco di stdin e di stdout

### 3.2.2 Esercizio

1) usando cat, scriva due righe sul file pippo.txt , la prima contenente ciao , la seconda contenente come stai?\`

2) sempre usando cat, aggiunga al file pippo.txt una terza riga: io bene\`

Nota bene: la doppia \$>\$ indica che deve appendere a un file esistente\`

3) copiare pippo.txt nel file pluto.txt , usando solo il comando cat\`

- cat <pippo.txt >pluto.txt

cioe', con il < utilizza pippo.txt come stdin , con il > utilizza pluto.txt come stdout

rimpiazza stdin con pippo.txt e rimpiazza stdout con pluto.txt

Altrimenti potrebbe fare

```
cat pippo.txt >pluto.txt
```

in quanto cat copia su stdout tutti i file che le gli passa sulla riga di comando.

```
cat pippo.txt pluto.txt >paperino.txt
```

ed ora cosa accade?

```
cat * >paperino.txt
```

paperino.txt viene riscritto e quindi non viene appeso a se' stesso!!.

## 3.3 ls

```
man ls
```

```
ls -l
totale 12
-rw-rw-r-- 1 user user 48 ago 23 10:51 paperino.txt
-rw-rw-r-- 1 user user 24 ago 23 10:47 pippo.txt
-rw-rw-r-- 1 user user 24 ago 23 10:49 pluto.txt
```

rw-rw-r-- : indicano le proprietà di appartenenza dei file/cartelle elencati  
i primo tre elementi rw- indica che il proprietario può leggere (r), scrivere (w), e non eseguire (-)  
cioè' read write execute => rw-, quindi significa che non si può fare quella cosa

la seconda terna rwx indica le stesse cose ma per il gruppo di appartenenza  
la terza le indica per l'utente root

Esempio: voglio bloccare in lettura il file paperino.txt  
quindi voglio che il comando cat paperino.txt mi torni errore

Per modificare le proprieÀ di un file si usa chmod seguito da un numero in ottale (da 0 a 7) dove i primi tre bit sono rwx per proprietario e gli altri per gruppo e per root

rwx => 111 => 7

rw- => 110 => 6

r-- => 100 => 4

-w- => 010 => 2

chmod 266 paperino.txt

chmod u=w paperino.txt

chmod g=rw paperino.txt

chmod o=rw paperino.txt

x su una cartella significa che la cartella la puo' aprire (listare oppure entrarci dentro)

E se aggiungessi

chmod u-w paperino.txt

come potrei cancellare paperino.txt?

sempre con rm poiche' e' un file che mi appartiene

- il secondo elemento mostrato da ls -l sono i riferimenti al file

- la terza colonna Ã" il nome del proprietario (user)

Come faccio a sapere chi sono io?

1) whoami

2) id

id

uid=1000(user) gid=1000(user) gruppi=1000(user),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),122(lpadmin),134(lxd),135(sambashare)

uid: user id

gid: group id

questi due sono quasi sempre uguali e vengono creati per ogni utente

i rimanenti sono i gruppi di appartenenza dell'utente

- la quarta colonna Ã" il gruppo di appartenenza del proprietario del file

- la quinta colonna indica le dimensioni (in byte) del file

creiamo una cartella

mkdir folder

ora abbiamo

ls -l

totale 12

drwxrwxr-x 2 user user 4096 ago 23 11:45 folder

-rw-rw-r-- 1 user user 24 ago 23 10:47 pippo.txt

-rw-rw-r-- 1 user user 24 ago 23 10:49 pluto.txt

notiamo due cose:

1) il primo elemento delle proprietà cambia da - a d (directory)

2) le dimensioni sono 4096 ??? per quale motivo??? il motivo che una cartella e' un file speciale le cui dimensioni sono a blocchi da 4096 (la dimensione del blocco dipende dal file system utilizzato).

- la sesta colonna Ã" la data di ultima modifica del file/cartella  
l'ultimo elemento Ã" il nome del file/cartella

il comando touch crea o modifica la data di modifica di un file

- ls -lR mostra ricorsivamente file e cartelle
  - ll -rt mostra i file e le cartelle e le ordina per data (in fondo il più recente)
  - ll -rS ordina per dimensioni
  - ll \*
- mostra tutti i file della cartella corrente. Se trova una cartella, ne mostra il contenuto. Non mostra ricorsivamente (-R) ma mostra solo il contenuto delle cartelle presenti nella cartella corrente.
- ll -d \*
- non apre le cartelle

### 3.4 wc

man wc

- Esercizio: contare quanti file e cartelle ci sono nella directory corrente.  
Usare solo ls e wc e pipe (|)  
ls -R | wc  
la pipe prende lo stdout del comando alla sua sinistra e lo fa diventare lo stdin del comando alla sua destra

### 3.5 cp

man cp

una copia normale cambia la proprietà del file assegnandola a chi ha chiamato la copia

Supponiamo però che io voglia mantenere i diritti e le proprietà originali (esempio: sto facendo un backup, in tal caso quando lo dovrò ripristinare allora le proprietà non verranno essere quelle originali!!!)

cp -r .... copia ricorsivamente modificando le proprietà

cp -a .... copia ricorsivamente lasciando inalterate le proprietà

- copiare i file pippo.txt e pluto.txt nella cartella prova

### 3.6 mv

man mv

mv di una cartella la sposta tutta, quindi non ha bisogno di specificare il -r

esempio: spostare tutto in una cartella che si chiama nuova nella cartella corrente

```

mkdir nuova
mv * nuova

ora ripristinare i file al posto "precedente"
mv nuova/* .

— Esercizio: creare 100000 file nella cartella nuova
for ((i=0; i<100000; i++))
do
    touch nuova/$i.dat
done

— creare una nuova cartella di nome vuota e spostarci tutti i file presenti
    nella cartella prova
mkdir vuota

mv nuova/* vuota/
-bash: /usr/bin/mv: Elenco degli argomenti troppo lungo

Allora potrebbe fare: rmdir vuota

mv nuova vuota

cosa accade se prima non cancella vuota?
che in vuota viene spostata nuova.

se vuota non esiste allora la mv nuova vuota cambia il nome di nuova in vuota
ma se la cartella vuota esistesse già e non fosse possibile eliminarla?

```

### **3.7 rm**

man rm

```

con il flag -i chiede di cancellare
con il flag -f non chiede conferma
con il flag -r cancella ricorsivamente

```

### **3.8 cd**

man cd

in linux non si usa \ ma si usa /

### **3.9 mkdir**

man mkdir

```

vogliamo creare una serie di cartelle una nell'altra
vuota/f1/f2/f3
mkdir -p vuota/f1/f2/f3

```

```

mkdir -m 0 esempio
che cosa accade?
inaccessibile
ora la modifichi in modo da renderla accessibile solo a lei stesso!!!
chmod 700 esempio

```

### **3.10 rmdir**

man rmdir

cancella solo cartelle vuote!!!  
con il -p cancella una gerarchia di folder , l'importante che siano vuote!!!

### **3.11 bash wildcards**

\*

per indicare i nomi di file e cartelle puo' utilizzare \* oppure ?  
il ? e' per i singoli caratteri

ls ?  
list tutti i file i cui nomi sono lunghi un singolo carattere

ls ??  
lista tutti i file i cui nomi sono di due caratteri

lista tutti i file il cui nome inizia per 1  
ls 1\*

tutti i file che iniziano per 2 e il cui nome Ã" lungo 2 caratteri  
ls 2?

listare tutti i file che contengono nel loro nome il numero 100. Ricordare che  
nella cartella "nuova" i file sono numeri che terminano con .dat  
ls nuova/\*100\*.dat

e quanto sono?  
ls nuova/\*100\*.dat | wc

listare tutti i file il cui nome contenga 9 e 0 in questo ordine. prima 9 e poi,  
anche dopo alcuni caratteri , uno 0

ls nuova/\*9\*0\*

listare tutti i file il cui nome contiene , in qualsiasi ordine , un 9 uno 0 e un  
2!!!  
ls nuova/\*9\*0\*2\* nuova/\*9\*2\*0\* nuova/\*0\*9\*2\* nuova/\*2\*9\*0\* ...

listare tutti i file il cui nome contiene o un 9 o un 2  
ls nuova/\*9\* nuova/\*2\*  
oppure  
ls nuova/\*[92]\*

cosa fa  
rm ??  
cancella tutti i file il cui nome e' lungo 2  
rm -f ??  
cancella senza chiedere conferma!!!

### **3.12 grep**

man grep

facciamo un file che contiene i numeri da 0 a 99999

```
for ((i=0; i<100000; i++)); do echo $i >> elenco.txt; done

Cerchiamo nel file tutte le righe che contengono tre zeri (000)
grep "000" elenco.txt
con grep il . indica un carattere qualsiasi
.* indica una qualsiasi sequenza di caratteri

esempio
cerca in elenco.txt tutte le righe che contengono un 1 seguito da un due dopo
    un qualsiasi carattere: 23188266.dat c'e' un 1 poi 88 e poi un 2
grep "1.2" elenco.txt no! in questo caso tra 1 e 2 c'e' un solo carattere

. : un qualsiasi carattere
x*: il carattere x ripetuto 0 o piu' volte
grep "1..*2" elenco.txt
```

### 3.13 less

man less

```
mostra i contenuti di un file di testo. ci si puo' muovere in avanti e indietro
    sia con le frecce, sia con i comandi f e b oppure con page-up e page-down
E' possibile cercare qualcosa utilizzando il comando /
/999
cerca la prossima riga che contiene 999

/9.4
cerca la prossima riga che contiene un 9 seguito da un qualsiasi carattere e poi
    dal numero 4

q per uscire
h per help

per passare da un file all'altro
:n
```

### 3.14 pwd

man pwd

```
pwd
=> stampa il path completo della cartella corrente
```

Nota bene: Si ricordi che in bash la variabile di ambiente PS1 contiene i comandi per mostrare la prompt di bash

### 3.15 diff

man diff

Mostra le differenze tra due file  
Serve per capire come e' stato modificato un file, ad esempio.

```
diff pippo.txt pluto.txt
1c1
< ciao
```

```
—  
> ciao11  
4d3  
< aaa  
  
diff pippo.txt pluto.txt  
  
ricorda di uscire con :q, per due volte poiche' sono due file
```

### 3.16 file

man file

```
file *  
elenca tutti i file nella cartella e indica di che tipo sono
```

### 3.17 wget

man wget

consente di scaricare da internet dati, immagini, interi siti web

```
wget http://www.google.it/index.html  
less index.html
```

### 3.18 factor

man factor

```
factor 123456789012345678901234567890  
123456789012345678901234567890: 2 3 3 3 5 7 13 31 37 211 241 2161 3607 3803  
2906161
```

trova i fattori primi di un numero di qualsiasi valore

### 3.19 find

man find

cercare in tutto il computer un file che si chiama pippo.txt

potrei fare  
sudo ls -lR / | grep "pippo.txt"  
ma, per prima cosa mi mostra un sacco di errori  
Soluzione  
sudo ls -lR / 2> errori.log | grep "pippo.txt"

in questo modo gli errori di ls tramite la redirezione 2> li mando su altro file

in bash lo stream stdin si indica con 0  
in bash lo stream stdout si indica con 1  
in bash lo stream stderr si indica con 2

ls 1> elenco.txt  
questo equivale a dire metti il risultato di ls nel file elenco.txt, ma senza  
metterci gli errori

quindi posso redirigere sia l'output sia gli errori di un programma

Tornando alla ricerca , anche se la ls -R funziona , non mi dice dove stanno i file trovati!!!

```
find / -name "pippo.txt" 2>errori.log
/home/user/Documenti/pippo.txt
/home/user/Lavoro/pippo.txt
```

Esempio: cerchi nella cartella /lib tutti i file il cui nome inizia con lib e termina con so  
find /lib -name "lib\*so"

```
#find <cartella di ricerca> <opzioni per cercare: -name, -type, -size, -atime, -ctime...> <opzioni per eseguire comandi: -exec -ok -delete ...> <opzioni per stampare -print -print0 ...> <altre opzioni visita in preordine, vista in postordine, ....>
```

```
#cerca tutti i file il cui nome contiene il numero 10
find . -name "*10*"
```

```
#cercare tutti i file il cui nome contiene il numero 1000 e visualizzarne il tipo
find . -name "*1000*" -exec file {} \;
```

```
#cercare tutti i file (non le cartelle) presenti nella cartella corrente e stamparne il nome
find . -type f
```

```
#cercare tutti i file (non le cartelle) del disco con il nome che contiene la sottostringa .so e stamparne le dimensioni in byte
sudo find / -name "*.so*" -type f -exec du -b {} \;
```

```
#cercare tutti i file nella cartella corrente e stamparne il nome al contrario (se il nome era ex1.sh deve stampare hs.1xe)
echo pippo | rev
echo pippo => sullo stdout sono inviati i caratteri p i p p o
|           => tutto quello che gli arriva a sinistra sullo stdout (p i p p o) lo butta a destra sullo stdin
rev          => riceve dallo stdin e quindi riceve p i p p o
echo
rev
      |
      pippo
```

Quindi !!!!!

```
find . -type f | rev
```

```
#cercare tutti i file (non le cartelle) del disco con il nome che contiene la sottostringa .so, di dimensioni maggiori di 10 MBytes e stamparne le dimensioni in byte
sudo find / -type f -size +100M -name "*.so*" -exec du -b {} \;
```

```
#cercare tutti gli elementi della cartella corrente creati negli ultimi 5 giorni
find . -atime -4
```

```
#cercare tutti i file presenti nella cartella corrente, senza scendere nelle sottocartelle
find . -maxdepth 0 -type f
```

### **3.20 du**

man du

```
disk usage: du <file o directory>
torna le dimensioni del file passato come parametro oppure di tutta la
cartella
Opera ricorsivamente sulle directory
du -s -h . :> in megabyte/giga...
du -s -b . => in byte
```

### **3.21 locate**

man locate

```
locate pippo.txt
```

### **3.22 head**

man head

```
head torna la prima parte di un file
```

### **3.23 tail**

man tail

```
tail torna l'ultima parte di un file
```

```
tail -f <file>
mostra in diretta come cambia il file
```

```
se non so dove sta il file potrei
find . -name "lista.txt" | xargs tail -f
oppure
tail -f 'find . -name "lista.txt"'
```

```
esercizio: mettere le prime 40 righe di elenco.txt nel file inizio.txt
head -40 elenco.txt > inizio.txt
```

```
copiare le righe dalla 10 alla 30 di elenco.txt e metterle in inizio.txt
tail -n +10 elenco.txt | head -n 20
```

### **3.24 sort**

man sort

```
ordina lo stdin secondo un ordine numerico oppure lessografico
```

```
ordina le prime 20 righe del file elenco.txt in modo inverso
head -n 20 elenco.txt | sort -n -r
```

```
ordina in modo inverso le righe del file elenco.txt che vanno dalla 1000 alla
1010
```

```
tail -n +1000 elenco.txt | head -n 10 | sort -n -r
```

### 3.25 passwd

man passwd

cambia la password

nel file /etc/passwd ci sono informazioni sugli utenti e sulle shell

### 3.26 man

man man

per cercare il manuale di un comando, esempio, ls  
man ls

per cercare tutti i manuali che contengono un riferimento a una stringa, es:  
password,  
man -k password

per accedere ad un manuale secondario, utilizzare in numero del manuale  
esempio

man -k password torna:

chage (1)	– cambia le informazioni sulla scadenza della password
chgpasswd (8)	– aggiorna le password di gruppo in modalità non interattiva
...	
chpasswd (8)	– aggiorna le password in modo non interattivo
expiry (1)	– controlla e fa rispettare la scadenza della password
grpconv (8)	– convertono a e da password e gruppi shadow.
grpunconv (8)	– convertono a e da password e gruppi shadow.
login.defs (5)	– configurazione del pacchetto password shadow
passwd (1)	– cambia la password utente
passwd (5)	– il file delle password
pwck (8)	– verifica l'integrità dei file delle password
pwconv (8)	– convertono a e da password e gruppi shadow.
pwunconv (8)	– convertono a e da password e gruppi shadow.
shadow (5)	– file delle password shadow
vigr (8)	– modifica i file delle password, dei gruppi, delle password
...	
vipw (8)	– modifica i file delle password, dei gruppi, delle password
...	
apg (1)	– generates several random passwords
cpgr (8)	– copy with locking the given file to the password or group file
...	
cppw (8)	– copy with locking the given file to the password or group file
...	
cracklib-check (8)	– Check passwords using libcrack2
create-cracklib-dict (8)	– Check passwords using libcrack2
grub-mkpasswd-pbkdf2 (1)	– generate hashed password for GRUB
openssl-passwd (1ssl)	– compute password hashes
openssl-srp (1ssl)	– maintain SRP password file
pam_pwhistory (8)	– PAM module to remember last passwords
pam_pwquality (8)	– PAM module to perform password quality checking
pam_unix (8)	– Module for traditional password authentication
seahorse (1)	– Passwords and Keys
systemd-ask-password (1)	– Query the user for a system password
systemd-ask-password-console.path (8)	– Query the user for system passwords on the console
...	
systemd-ask-password-console.service (8)	– Query the user for system passwords on the console
...	

```
systemd-ask-password-wall.path (8) - Query the user for system passwords on th  
...  
systemd-ask-password-wall.service (8) - Query the user for system passwords on  
...  
systemd-tty-ask-password-agent (1) - List or process pending systemd password  
...  
unix_chkpwd (8) - Helper binary that verifies the password of the current  
...  
unix_update (8) - Helper binary that updates the password of a given user
```

Qui leggo che per la stringa passwd ci sono due manuali, uno al posto (1) l'altro al posto (5)

per leggere il manuale del primo e' sufficiente  
man passwd

oppure man 1 passwd

per gli altri sempre mettere il numero

man 5 passwd

**esercizio:**

trovare tutti i manuali che parlano di socket  
man -k socket

per quanto riguarda le password e le informazioni reali sugli utenti il file,  
leggibile solo da root, e':  
/etc/shadow

### **3.27 set**

per vedere tutte le variabili di ambiente utilizzare il comando set

per leggerne il valore di solo una, utilizzare  
echo \$variabile

### **3.28 ps**

man ps

ps x Show all of your running processes.

ps ax Show all processes on the system, not just the ones you own.

ps u Include more detailed information on processes.

ps w Show full command names, not just what fits on one line.

ps auxw | less

### **3.29 kill**

man kill

il segnale di STOp si invia anche con CRTL-Z  
e poi fg o bg per rittivarlo in foreground oppure in background.

**esercizio:** trovare tutte le bash che sono in esecuzione

ps auxw|grep "bash"|grep -v "grep"

con il flag -v grep esegue una ricerca al contrario, trova tutte le righe che non contengono grep

```
con kill -9 oppure -15 il processo viene ucciso completamente senza possibilita' di ripristino!!!
```

### 3.30 sort e unique

```
man unique
```

```
head elenco.txt > a.txt  
cat a.txt a.txt a.txt a.txt >duplicati.txt  
ora il file duplicati.txt contiene tanti numeri duplicati
```

```
con sort duplicati | uniq ottengo tutti i valori senza duplicati  
uniq elimina tutti i duplicati, adiacenti tra loro
```

### 3.31 Utilita'

```
per generare un file con 5 stringhe di 12 caratteri per linea  
for ((i=0; i<100; i++)); do echo `pwgen 12 5` >>stringhe.txt; done
```

### 3.32 compressione

```
gzip bzip2 xz
```

```
gzip duplicati.txt  
il risultato è il file  
duplicati.txt.gz  
zcat duplicati.txt.gz | less  
zgrep "100" duplicati.txt.gz  
  
gunzip duplicati.txt  
  
gzip -9 duplicati.txt massima compressione  
  
xz -9 duplicati.txt comprime molto di piu'  
  
xz puo' anche comprimere in parallelo
```

Esempio:

```
for ((i=0; i<100; i++)); do echo `pwgen 120 1000` >>folder/$i.txt; done  
  
creo 100 file abbastanza grandi  
gzip -r -v folder  
  
!!! gzip rimpiazza i file  
con -k lascia anche i file decompressi  
  
xz non ha la ricorsione  
xz -v9 folder/* !!comprime solo il primo livello di una cartella  
unxz -v9 folder/* !!decomprime
```

Esempio: cancelliamo tutti i figli di folder  
rm -rf folder/\*

```
#nel file genera.sh scrivo  
for ((i=0; i<5; i++))  
do
```

```

echo 'pwgen 10 5' >>folder/$i.txt
mkdir folder/$i.folder
cd folder/$i.folder
for ((j=0; j<5; j++)); do echo 'pwgen 10 5' >>$j.txt; done
cd ../..
done

```

Per essere sicuro che il file venga eseguito come script bash  
in testa al file scrivo  
#!/bin/bash

Per eseguirlo due possibilità  
1) bash genera.sh

2) chmod a+x genera.sh  
./genera.sh

Se vuoi provare il comando prima di eseguirlo per davvero, bash -n  
genera.sh

Dopo aver creato tutte le cartelle e i file

- 1) come posso listare tutti i contenuti della cartella folder?  
– ll -R folder  
– find folder  
– find folder -name “\*”
- 2) come posso comprimere tutti i file contenuti in folder e nelle sue sottocartelle?  
– gzip -r folder #Comprime tutte le sottocartelle di folder
- 3) dentro folder ci sono sia cartelle sia file. Voglio elencare esclusivamente tutte le sottocartelle di folder, senza elencare i file  
Ieri abbiamo visto il comando: file  
Oppure, giustamente, usando ll e leggendo il primo carattere di ogni riga (d)  
ll -R folder/ | grep “^d”  
ok, ma i risultati sono ancora troppi  
ls -l -R folder/ | grep “^d” #con questo mostro solo le cartelle
- 4) se invece volessi usare find, ci sono un insieme di “flag” che posso usare con find, uno di questi è -type (noi finora abbiamo conosciuto -name)  
– find folder -type d  
e se invece volessi listare tutti i file?  
– find folder -type f
- 5) un ulteriore flag di find è il flag -exec. Con questo tu esegui un comando sull’elemento correntemente puntato da find nel comando per indicare il nome dell’elemento si usa {}, per indicare la fine del comando, si usa \;  
find folder -type f -exec ls -l {} \;  
con questo comando io cerco tutti i file nella cartella folder e per ognuno di essi eseguo il comando ls -l
- 6) Esercizio  
– decomprimere tutti i file contenuti nella cartella <folder>  
– gunzip -r folder  
questo va bene se i file sono compressi con gzip!!!

- per ognuno dei file contenuti nella cartella folder , contare di quanti caratteri è composto
  - a) come faccio a contare i caratteri di un file?
    - `wc -c <file>` ##conta solo le righe
  - b) come faccio ad applicare il comando `wc -c` a tutti i file della cartella folder?
    - `find folder -type f -exec wc -l {} \;`

7) esercizio

- comprimere con xz tutti i file della cartella folder
  - !!!xz non ha -r
    - la logica sarebbe per ogni file della cartella folder , lo comprimo
    - `find folder -type f -exec xz -v {} \;`
- elencare tutti i file compressi con xz presenti nella cartella folder. Supponiamo che un file compresso con xz , termini con il suffisso `<.xz>`
  - !!!ce ne potrebbero essere alcuni non compressi con xz e quindi che non terminano con `<.xz>`

quale scegliamo?

- a) `find folder -type f -name "*.xz"`
- b) `find folder -name "*.xz"`

Scegliamo a) poiché volgiamo solo i file che terminano con xz

- 8) sapendo che il comando `xzcat <file.xz>` mostra i contenuti originali (decompressi) del file `<file.xz>`, cercare in tutti i file `<.xz>` la stringa "ooY"

`find folder -type f -name "*.xz" -exec xzcat {} \; | grep "ooY"`

!!ops !! , ma c'è anche `xzgrep`  
`find folder -type f -name "*.xz" -exec xzgrep "ooY" {} \;`

ok, ma non mi dice i nomi dei file !!!

In questo find mi aiuta , esiste un flag che si chiama `-print` che stampa il nome del file

`find folder -type f -name "*.xz" -exec xzgrep "ooY" {} \; -print`

- 9) cancellare da folder tutti i file che contengono un carattere 2 nel proprio nome

- a) come posso cercare tutti i file che contengono 2 nel proprio nome?

`find folder -type f -name "*2*"`

- b) per cancellare , due modi

- 1) usa `-exec rm -f {} \;`
- 2) usa `-delete`

`find folder -type f -name "*2*" -delete`

`find folder -type f -name "*2*" -exec rm -f {} \;`

### 3.33 processi in background

per eseguire un processo in background , aggiungere il carattere & al termine del comando

Esempio: scrivere sul file `<tutti.txt>` il nome e le dimensioni in byte di tutti i file presenti nel sistema .

NB: potrebbero essere così tanti che il comando per scriverli potrebbe impiegare diversi minuti!!!

Proviamolo senza background e poi con il background!  
`sudo find / -type f -exec wc {} \; -print >tutti.txt`

```

sudo find / -type f -exec wc {} \; -print >tutti.txt &
per vedere i miei processi che ho lanciato in background:
jobs
[1]+  In esecuzione          sudo find / -type f -exec wc {} \;
      -print > tutti.txt &

jobs -l mostra anche id del processo
per kill potrei fare
kill -9 <id del processo>

- se ne eseguo un altro
sudo find / -type f -exec wc {} \; -print >tuttil.txt &

- per metterne uno in foreground (ad esempio per chiuderlo)
fg <numero del job>

se poi faccio CTRL-Z, lo sospendo
e se poi faccio
bg <numero del job>
lo rимetto in background

per terminarli o kill oppure fg seguito da CTRL-C

```

### 3.34 ln

man ln

```

ln -s tutti.txt ilmiolink
ll -rt
lrwxrwxrwx 1 user user 9 ago 24 11:41 ilmiolink -> tutti.txt
- cosa accade se lo cancello?
rm ilmiolink

un hard link (senza il -s)
ln tutti.txt ilmiolink
ll -rt
-rw-rw-r-- 2 user user 7854081 ago 24 11:36 tutti.txt
-rw-rw-r-- 2 user user 7854081 ago 24 11:36 ilmiolink
sono lo stesso file con due nomi che puntano a lui

esempio:
cp tutti.txt prova.txt
ln -s prova.txt provalnk
rm prova.txt
il link rimane ma non è più valido!!!

cp tutti.txt prova.txt
ln prova.txt provalnk
rm prova.txt
prova.lnk continua a esistere ed è valido

```

Quindi:

un link simbolico quando cancello il file originale non vale più

un hard link, quando cancello il file originale, continua a puntare ai dati originali

cioè gli hardlink aggiungono nomi agli stessi dati. Linux,  
cancellerà i dati solo quando  
sarà cancellato l'ultimo nome

Nel nostro file system se fai `ls -l /`  
ottiene un insieme di informazioni sui link simbolici del FS

### 3.35 tar

`man tar`

Creazione di archivi. Consente quindi di creare un unico file che  
contiene intere cartelle

`tar cf archivio.tar elenco cartelle o file da archiviare`  
Esempio:

```
tar cf archivio.tar nuova
- se voglio verificare cosa sta facendo
tar cvf archivio.tar nuova
```

```
file archivio.tar
archivio.tar: POSIX tar archive (GNU)
```

Ma lo stesso sarebbe se facessi  
`tar cvf archivio nuova`

```
file archivio
archivio: POSIX tar archive (GNU)
```

Il comando `file` opera su una cosa che si chiama "magic numbers"  
In linux ogni tipo di file può avere all'inizio una zona in cui  
ci sono alcuni numeri che consentono di riconoscerlo come  
tipo

Ovviamente posso ingannare il comando `file` scrivendo nella zona  
dei magic numbers, gli stessi numeri di un tipo diverso di  
file

Come faccio a leggere i contenuti di un archivio `tar`?  
`tar tf <tar file>`

`tar tf archivio`

Se voglio recuperare l'archivio  
`tar xf <tar file>` e mi ripristina le cartelle e i file nella  
cartella corrente  
`tar xf archivio`  
oppure, per verificare l'avanzamento,  
`tar xvf archivio`

Se vuole anche comprimere, allora usa `z`  
`tar czf archivio.tgz nuova`

Per estrarre e decomprimere da un archivio compresso  
`tar xzf <nome archivio>`

Per verificare il contenuto di un archivio zippato  
`tar ztf <nome archivio>`

### Esercizio

- creare l'archivio <tutto.tar> dalle due cartelle <nuova> e <folder>

## 3.36 I device

I device (`/dev`) rappresentano il punto di congiunzione tra il Sistema Operativo e il mondo esterno.

Video, Audio, dischi, memoria, filesystem, reti, ecc. sono tutti rappresentati o rappresentabili come device.

ci sono device fisici e device virtuali

Esempio:

`/dev/null` è un device "pozzo": qualunque cosa venga inviata a questo device, viene buttata

I dischi

eseguendo il comando `df -l -T`, mostra i file system locali (non di rete) e il loro tipo

per ottenere informazioni complete su un device  
`udevadm info --query=all --name=/dev/sda`

`cat /dev/input/mousexxx`

`cat /dev/tty` => la tastiera correntemente collegata al sistema

Generazione di numeri casuali

`head -c 100 /dev/random | cat -v` (cat -v mostra anche i caratteri non stampabili)

Il comando `xxd`, mostra un file o uno stream in formato binario

`head -c 100 /dev/random | xxd`

`cat /proc/devices` => mostra tutti i device realmente presenti e usati dal sistema

`lsscsi` #mostra i dischi scsi collegati al sistema

creiamo un nuovo disco con vmware  
(di tipo scsi)

`sudo fdisk -l`

`sudo mkfs.ext4 /dev/sdb`

montiamo il nuovo fs

`sudo mount /dev/sdb /mnt` (umount `/dev/sdb` oppure `umount /mnt`)  
!!/`mnt` è per montarci file system temporanei, che poi smonto

In genere i dischi usb sono montati automaticamente

!!Come facciamo a fare in modo che linux monti questo disco ad ogni bootstrap?  
facciamo il reboot

```
Andiamo nel file /etc/fstab  
sudo vim /etc/fstab  
aggiungiamo il nuovo file system  
/dev/sdb /external ext4 errors=remount-ro 0 1  
  
Al boot il nuovo disco sarà montato nella cartella /external  
  
udevadm monitor  
monitorizza la connessione e al disconnessione di nuovi dispositivi  
Questo è un comando estremamente comodo  
  
blkid => fornisce tutti gli UUID dei file a blocchi  
mount UUID=a9011c2b-1c03-4288-b3fe-8ba961ab0898 /home/extra
```

```
Ubuntu monta in automatico sotto /media  
  
per un check del file system  
fsck /dev/...
```

### 3.37 dd

man dd

```
consente di copiare un device su un file o su altro device
```

```
dd if=/dev/zero of=new_file bs=1024 count=1  
Sta creando un file vuoto di dimensioni 1024
```

```
//Voglio cancellare completamente un disco  
dd if=/dev/zero of=/dev/sda1
```

```
!! Il programma zerofree azzerà tutto lo spazio "libero" del disco  
!! Utile per stringere le partizioni!!
```

### 3.38 tmux

man tmux

```
Consente di creare più terminali all'interno della stessa finestra  
per uscire CTRL-D  
per creare un terminale CTRL-B c  
Per cambiare terminale CTRL-B 0/1/2/3....
```

### 3.39 Swap space

```
swapon  
mostra cosa sta usando come swap
```

```
lo swap può essere o una partizione oppure un file su disco
```

```
## crea un file con tutti 0 di dimensioni 1GByte  
dd if=/dev/zero of=swap_file bs=1024k count=1024
```

```
## lo inizializza come swap  
mkswap swap_file
```

```

## Lo assegna al S.O. come swap area
swapon swap_file

##Di quanto swap uno ha necessità?
prima era il doppio della ram. Oggi se uno ha più di 32 GByte di ram può
usare , ad esempio , 16GByte di swap oppure anche nulla!!

```

### 3.40 File system

ext4 il file system di linux risiede parzialmente su ram e parzialmente su disco  
è un file syste journaled che si sincronizza rapidamente con lo stato del disco.  
è quasi impossibile "rompere" la struttura del file system  
Un crash del sistema difficilmente rovina il file system.

### 3.41 Il processo di boot

1. The machine's BIOS or boot firmware loads and runs a boot loader.
2. The boot loader finds the kernel image on disk, loads it into memory, and starts it.
  - se non riesce a entrare nel menu du grub, allora:  
 edit del file /etc/default/grub  
 GRUB\_DEFAULT=10  
 GRUB\_TIMEOUT\_STYLE=menu  
 GRUB\_TIMEOUT=10  
 e poi sudo update-grub  
 All'avvio parte il menu di grub  
 posizionarsi su ubuntu e selezionre modalità di ripristino  
 seguire i menu
3. The kernel initializes the devices and its drivers.  
  - insmod consente di installare un driver nuovo per un device non conosciuto dal kernel
4. The kernel mounts the root filesystem.
5. The kernel starts a program called init with a process ID of 1. This point is the user space start.
6. init sets the rest of the system processes in motion.
  - si ferma alla modalità testo
  - si ferma alla modalità grafica
L'ordine di avvio è
  - printing
  - cron
  - network
  - ...
7. At some point, init starts a process allowing you to log in, usually at the end or near the end of the boot.
  - con il login grafico , CTRL-ALT-F1/F2/F3... attivano i terminali testuali

Al termine del boot può esaminare tutti i messaggi con il comando dmesg

i run level vanno da 0 a 5  
who -r mostra il run level corrente

```

5 => graifca
4=> testo
3 => rete
2=> file system

sudo init 4 Attiva modalità testo
sudo init 6 shutdown

sudo systemctl set-default graphical.target

sudo systemctl set-default multi-user.target #modo testo

```

### 3.42 crontab

man crontab

Esegue periodicamente dei comandi  
 crontab -l #lista i servizi attivi  
 crontab -e #edit

### 3.43 date and time

man date

```

date #fornisce la data
hwclock --hctosys --utc

time <comando> #fornisce il tempo di esecuzione di un comando

time ( tar czf ex.tgz nuova; bash creacartelle.sh ) #Le parentesi tonde
consentono di raggruppare due comandi come se fossero uno solo

```

### 3.44 at

man at

con at può specificare che un particolare comando/script sarà eseguito  
 ad una certa ora

### 3.45 lsof

man lsof

```

lista i file aperti dai processi
lsof -p <pid>

include le comunicazioni su rete

```

### 3.46 strace

man strace

```

traccia lo stato di funzionamento di un programma
ltrace traccia le chiamate a librerie dinamiche

```

### **3.47 ps -m**

man ps

lista i thread

### **3.48 tree**

man tree

mostra l'albero delle directories e dei file nelal cartella corrente

### **3.49 Regex**

man regex

```
cp /usr/share/dict/american-english .
wget https://www.gutenberg.org/files/1661/1661-0.txt
mv 1661-0.txt SherlockHolmes.txt
grep <cosa cercare> <dove cercarlo>

#Cercare tutte le righe che contengono una doppia (11)
grep "11" SherlockHolmes.txt

#Cercare tutti i numeri di riga delle righe che contengono la doppia 1 (
    11)
grep -c "11" SherlockHolmes.txt #mi dice quante sono le righe che
corrispondono

#Calcolare quante righe non contengono la parola morning
grep -cv "morning" SherlockHolmes.txt

#E se morning fosse scritta in maiuscolo?
grep -cvM "morning" SherlockHolmes.txt

#Trovare tutte le righe che contengono Morning (con la M maiuscola)
grep "Morning" SherlockHolmes.txt

#Trovare tutte le righe che contengono la parola Morning (esattamente la
parola. Quindi sono escluse , ad esempio , _Morning)
grep -w "Morning" SherlockHolmes.txt

#Cercare nella cartella corrente tutti i file che contengono il numero
    100
grep "100" *

#E se volessi solo i nomi dei file
grep -l "100" *

#E se volessi cercare in tutte le cartelle e sottocartelle?
grep -rl "100" *

#E se volessi cercare in tutte le cartelle e sottocartelle e ottenere
per ogni file
quanti elementi ho trovato?
grep -cr "100" *

#Ora tra tutti quelli trovati (con la grep -cr "100" *), stampare solo
quegli che hanno almeno un match
```

```

grep -cr "100" * | grep -v ":0"

#Questo funziona come se
1) grep -cr "100" * >/tmp/elenco.txt
2) grep -v ":0" /tmp/elenco.txt
La pipe prende lo stdout del comando alla sua sinistra e lo trasforma
nello stdin del comando alla sua destra

#Cercare in SherlockHolmes tutte le righe che contengono la parola "
"morning" e, tra queste, cercare tutte quelle che contengono la parola
"the"
grep "morning" SherlockHolmes.txt | grep "the"

#Cercare in SherlockHolmes tutte le righe che contengono la parola "
"morning" e, tra queste, cercare tutte quelle che contengono la parola
"the" e, tra queste, cercare tutte quelle che contengono la parola "
Sherlock"
grep "morning" SherlockHolmes.txt | grep "the" | grep "Sherlock"

#Se lo avessi chiesto in questo modo: trovare tutte le righe che
contengono "morning", "the" e "Sherlock"
#sarebbe stato più difficile trovare la soluzione!!!!

#Per poter utilizzare un modello di ricerca più strutturato studiamo ora
le espressioni regolari (regular expressions: regex)
#!!!per utilizzare regex deve usare egrep e non grep
. rappresenta qualsiasi carattere (a volte il fine riga è escluso)
^ rappresenta l'inizio di una riga
$ rappresenta il termine di una riga
* ripete da 0 a infinite volte il simbolo alla sua sinistra (es: a* =>
" " "a" "aa" "aaaa" "aaaaaaaaaaaaaaaa" ...)
+ ripete da 1 a infinite volte il simbolo alla sua sinistra (es: a+ => "
a" "aa" ma non è "")
[abcdfeg] rappresenta uno dei caratteri elencati un simbolo ^ al primo
posto indica la negazione
| rappresenta in modo alternativo l'espressione alla sua sinistra oppure
l'espressione alla sua destra. (es: a+|b+ => a aa aaa b bb bbb bbbb
)
(expr) le parentesi tonde racchiudono un'espressione regolare in un
unico blocco.

Esempio (abc)+ matcha abc, abcabc, abcabcabc, ...
{n} oppure {n,m} indica la ripetizione del carattere/espressione
regolare alla sua sinistra per n volte oppure da n a m volte

#Nel file SherlockHolmes.txt cercare tutte le righe che contengono le
lettere
M r i g, separate l'una dall'altra, da un singolo carattere. Esempio
Marqiag è ok
M;r!iag è ok
Maarig no
egrep "M.r.i.g" SherlockHolmes.txt

#Nel file SherlockHolmes.txt cercare tutte le righe che contengono una
A seguita da 5 caratteri e poi una b

#Nel file SherlockHolmes.txt cercare tutte le righe che contengono una
"a" seguita da almeno una "b" (sto cercando quindi ab, abb, abbb, abbbb,
...)
egrep "ab+" SherlockHolmes.txt

```

```

#Nel file SherlockHolmes.txt cercare tutte le righe che contengono un
    numero di almeno 4 digit. L'operatore [] consente anche di
    rappresentare insiemi sequenziali: [0-9] rappresenta 0, 1, 2, 3, 4,
    5, 6, 7, 8, 9. [a-h] rappresenta a,b,c,d,e,f,g,h
- egrep "[0-9][0-9][0-9][0-9]+" SherlockHolmes.txt
- egrep "[0-9]{4}" SherlockHolmes.txt

egrep "\[e" SherlockHolmes.txt

#Generano il file random.txt che contiene stringhe casuali
for ((i=0; i<10000; i++)); do head -c 32 /dev/random | sha512sum >>
random.txt; done

#Dimostrare che il file random.txt contiene esclusivamente caratteri
    0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f e quindi non contiene g,h,i,j,k,l,m,n
    ,o,p,q,r,s,t,u,v,w,x,y,z e neppure i simboli di interpunkzione
    (,:;.!?...)
egrep "[g-z,;:-]" random.txt
Mi accorgo che manca tutta le linee poichè contengono il simbolo – come
    ultimo carattere
#!!!Come può trasformare il file random.txt nello stesso file ma
    togliendo la coppia di spazi e il – che sono in fondo alle righe
egrep -o "[0-9a-f]+" random.txt >random1.txt ##prendo solo la parte che
    match l'espressione regolare

#Su random1.txt mi trovi tutte le righe che contengono numeri di almeno
    8 digit
egrep "[0-9]{8}" random1.txt

#Su random1.txt mi trovi tutte le righe che contengono numeri di
    esattamente 8 digit
egrep "[0-9]{8}" random1.txt
egrep "[^0-9][0-9]{8,8}[^0-9]" random1.txt

#In sherlock holmes trovare tutte le parole/frasi che non iniziano con [
    thmpPiw .] e sono lunghe almeno 6 caratteri, senza spazi bianchi
egrep "[^thmpPiw .][a-zA-Z0-9]{6,6}" SherlockHolmes.txt

#In sherlock holmes trovare tutte le parole di almeno 10 caratteri che
    sono a fine riga
egrep "[a-zA-Z0-9]{10}\$" SherlockHolmes.txt

#In sherlock holmes trovare tutte le parole di almeno 10 caratteri che
    sono a inizio riga
egrep "^[a-zA-Z0-9]{10}" SherlockHolmes.txt

#In sherlock holmes trovare tutte le parole di esattamente 12 caratteri
    e stampare solo queste
egrep -o "[^a-zA-Z0-9][a-zA-Z0-9]{12,12}[^a-zA-Z0-9]" SherlockHolmes.txt

#In sherlock holmes trovare tutte le parole di almeno 12 caratteri e
    stampare solo queste
egrep -o "[a-zA-Z0-9]{12}[a-zA-Z0-9]*" SherlockHolmes.txt

```

### 3.50 resolv.conf

man resolv.conf

```
sudo vim /etc/resolv.conf
```

```
lasciare solo  
nameserver 8.8.8.8
```

### 3.51 sed

man sed

Nel file SherlockHolmes.txt sostituire tutte le parole "Sherlock" con la parola "Investigatore"

```
sed -e '/Sherlock/s//Investigatore/' SherlockHolmes.txt
```

/cerca questa espressione regolare/ e su questa applica il comando sostituisce  
s// quello che hai trovato  
s//jsdjsjsjsj/ con questo

```
sed -e '/Sherlock Holmes/s//Investigatore/' SherlockHolmes.txt  
Per salvare le modifiche può utilizzare la redirect > su un nuovo file  
oppure utilizzare il flag -i (che rimpiazza il file)  
!! se usa -i e sbaglia, perde il file originale !!
```

```
#nel file SherlockHolmes.txt sostituire tutti i "no" con "si", dovunque  
si trovino i "no"  
Sed quando sostituisce lavora per ogni riga e sostituisce solo il primo  
che trova. Per sostituirli tutti deve aggiungere /g  
sed -e '/no/s//sì/' SherlockHolmes.txt
```

```
sed -e '/no/s//sì/g' SherlockHolmes.txt  
!!Funziona anche sui file utf-8 (unicode)
```

```
#Cercare in sherlockholmes.txt tutte le parole "the" e appendere a  
queste la stringa "eht"  
sed -e '/the/s//theht/' SherlockHolmes.txt
```

```
#Cercare in sherlockholmes.txt tutte le parole che iniziano per "th[  
aeiou]" e terminano con " " e appendere a queste la stringa "AAAAAA"  
sed -e '/\(\(th[aeiou]\[^ ]*\)\)/s//\1AAAAAA/' SherlockHolmes.txt | less
```

Quando circondi una espressione regolare con i caratteri \(\ e \), sto definendo un "sub-match"  
nella replace i submatch che ha trovato li può indicare con \1 (il primo ) \2 (il secondo)....  
sed -e '/\(\(th[aeiou]\[^ ]\*\)\)/s//\1AAAAAA/' SherlockHolmes.txt | less  
cerca il submatch \(\(th[aeiou]\[^ ]\*\)\)  
sostituisce \1 (sè stesso) seguito da AAAAAA

```
#Cercare in sherlockholmes.txt tutte le parole che iniziano per "A" e  
scrivici in testa la stringa XXXXXX  
sed -e '/\(\(A[^ ]*\)\)/s//XXXXXX\1/' SherlockHolmes.txt | less
```

```
#cercare tutte le parole che iniziano per "p" seguite da parole che  
iniziano per "q" e scambiarne la posizione  
Esempio
```

La prova quando vale => La quando prova vale

```
random2.txt contiene  
afdf1235f27a28e
```

```
afd12f12a12e78a19a26  
afd24f12a12e78a19a26
```

```
#cercare tutti numeri 1 seguiti da 2 e scambiarli tra loro nel file  
random2.txt
```

Esempio

```
afd1243fae => afd2143fae  
sed -e '/12/s//21/g' random2.txt
```

```
#Cercare tutti i numeri 1 o 2 o 3 o 4 o 5 seguiti da numeri 6 o 7 o 8 o  
9 o 0 e scambiarli tra loro
```

Esempio

```
afd12a39fae => afd12a93fae  
afd16a48fae => afd61a84fae
```

### 3.52 ascii

man ascii

```
man iso_8859 -1/2/3/4...
```

```
man utf-8
```

### 3.53 info

man info

Manuali estesi spesso sono nella sezione info

esempio:

```
info sed #Le mostra il manuale esteso di sed
```

La sezione info è in genere in /usr/share/info

### 3.54 Visual Studio code

Anche in code possono utilizzare le espressioni regolari!!!

i submatch sono delimitati da ( ) e non da \(\ )

Le sostituzioni non sono \1 ma \$1

### 3.55 Protocolli di rete

browser parla con google

- 1) protocollo http
  - contiene il protocollo applicativo (dell'applicazione web che sto usando)
- 2) protocollo TCP
  - contiene il protocollo HTTP
- 3) protocollo IP
  - contiene il protocollo TCP
- 4) Ethernet (mac addresses)
  - contiene il protocollo IP

Sotto linux posso utilizzare tcpdump che analizza la rete

C'è anche wireshark (sotto windows e sotto unix) che fornisce un'interfaccia grafica verso tcpdump

```
sudo tcpdump -i <scheda di rete/any> vari flag <comandi di  
filtraggio delle comunicazioni>
```

```
sudo tcpdump -i any -e (mostra lo strato ethernet) -n (usa  
indirizzi ip e non nomi di dominio) -s <quanti dati voglio  
leggere per ogni pacchetto> -X <mostra i dati in formato  
esadecimale> /-A (mostra i dati in formato ascii)
```

con ifconfig si ottiene il numero della scheda di rete  
sudo tcpdump -i ens33 -X -e -n not port 22 ##in questo modo stampa i  
pacchetti tranne quelli diretti alla porta 22 (ssh)

##Come fa a sapere che ssh è sulla porta 22 (a livello standard)?  
vim /etc/services

```
#dalla macchina virtuale alla macchina host faccio  
ping 10.211.55.1 (la macchina virtuale ha indirizzo 10.211.55.128)  
Il mac address della virtuale è: 00:0c:29:69:eb:57  
Il mac address dell'host è: 00:50:56:c0:00:08
```

Destinazione (sta per primo per motivi di efficienza e velocità. La  
scheda di  
rete legge questi 6 byte per prima cosa e se non sono il suo indirizzo  
MAC, non legge il resto ! si ricordi che il protocollo ethernet vuole  
che sulla rete viaggino tutti i pacchetti e quindi ogni macchina che  
sta sulla rete riceve anche i pacchetti indirizzati alle altre  
macchine)

!!! Differenza tra hub e switch

un Hub per esempio a 6 porte, duplica tutto quello che riceve su una  
porta verso le altre 5 (quindi se riceve un pacchetto sulla porta 1,  
lo invia alle 2,3,4,5 – Se riceve un pacchetto sulla porta 2, lo invia  
alle 1, 3,4,5) questo perché gli hub rispecchiano a pieno la  
specifica del protocollo ETHERNET che richiede che su una rete  
ethernet i pacchetti che viaggiano siano letti da tutte le macchine  
affacciate sulla rete.

ho un filo che parte dal router e che va all'hub. dall'hub partono tanti  
fili quanti sono i PC da collegare in rete.

Immagini un hub a 6 ingressi

- il PC A è collegato all'ingresso 1
- il PC B è collegato all'ingresso 2

A vuole inviare un pacchetto a B: come fa? A non sa su che  
ingresso è collegato B

Se A e B fossero collegati allo stesso filo di rete, allora  
qualunque cosa A metta sul file, viene letta da B, quindi se A  
manda un pacchetto al MACADDR di B allora dato che B legge  
tutto, leggerà che c'è un pacchetto per lui e quindi se lo  
prenderà.

Ma se fossimo in una situazione in cui A è collegato all'  
ingresso 1 e B è collegato all'ingresso 2 di un dispositivo  
(supponiamo un HUB). In che modo B potrebbe leggere il  
messaggio? Con gli HUB è facile. Tutto quello che l'HUB  
riceve su un ingresso lo manda a tutti gli altri.

Quindi Quando A invia un pacchetto a B, l'HUB, che non sa dove  
sta B, duplica il pacchetto su tutte le sue uscite e quindi  
anche sull'uscita dove è collegato B.

Invece cosa fanno gli Switch?

- gli switch si accorgono su che ingresso stanno A, B, ecc.
- Quando legge che A ha inviato un pacchetto i cui primi 6 byte  
sono il MAC address di B, allora lo switch mette il pacchetto  
solo sull'uscita a cui B è collegato!!!

```
//Pacchetto preso da tcpdump (whreshark)  
Analisi del protocollo  
00 50 56 c0 00 08 MAC destinazione
```

```

00 0c 29 69 eb 57 MAC Sorgente
08 00 2 byte di tipo di protocollo contenuto: IPV4

Header del pacchetto IP
45 Il 4 indica che è IPv4. Il 5 indica che la parte header del pacchetto
IP è lunga  $5*4 = 20$  byte (compreso il primo)
00 service type
00 54 Lunghezza totale del pacchetto IP (compreso header che nel nostro
caso è 20 byte) dato che sono 16 bit, un pacchetto IP può essere
lungo fino a  $2^{16} = 65536$  byte. Questo su ethernet non è mai vero. Su
ethernet la lunghezza del pacchetto IP è esattamente la lunghezza
massima del pacchetto ethernet (1500 byte).
ora 0054 in esadecimale è  $16*5+4=84$  byte - 20 byte dell'header = 64 byte

80
97 identification
40 00 flag e offset
40 time to live. Quanti router/gateway ci sono tra il suo PC e google?
01 protocollo icmp: PING!!!!!
35 eb
0a d3 37 80 => 10.211.55.128
0a d3 37 01 => 10.211.55.1

//Parte dati Header è terminato. I dati sono 64 byte.
08 00 a2 08 00 01 00 0b 24 6c 07 63 00 00
00 00 62 49 09 00 00 00 00 00 10 11 12 13 14 15
16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35
36 37

```

```

#Esame di un pacchetto TCP
45 IPV4 +  $5*4=20$  header
00
003c
6364 4000
40
06 PROTOCOL: TCP
4de7
0ad3 3780 : src address: 10.211.55.128
8efa b823 : dst address: 142.250.184.35

#La parte dati dell'IP!!! è il protocollo contenuto e quindi è il
#protocollo TCP
cf0 source port: 53200
0050 dest port : 80
8f82 1165 0000 0000
a002 faf0 899f 0000 0204 05b4 0402 080a
298e 099c 0000 0000 0103 0307

```

### 3.56 Introduzione alla configurazione di rete

```
https://ubuntu.com/server/docs/network-configuration
```

```
#Con il client del network manager
nmcli connection show
```

```
#Create a Static connection
```

```

sudo nmcli con add type ethernet con-name 'static-ip' ifname enp0s3 ipv4
    .method manual ipv4.addresses 192.168.1.102/24 gw4 192.168.1.1

#Add DNS IP to static-ip
sudo nmcli con mod static-ip ipv4.dns 192.168.1.1

#Activate the Static connection
sudo nmcli con up id 'static-ip'

#Aggiungere un address secondario
ip addr add <ip-address> dev <interface-name>
sudo ip addr del <ip-address> dev <interface-name>

#Oppure tramite interfaccia grafica !!!

#Per ripartire tutto
sudo nmcli networking off
sudo nmcli networking on

```

### 3.57 Alcuni comandi di uso comune

```

lsb_release -a Release corrente del sistema operativo
lsinitramfs lslogins lsns lspcmcia lsusb
lsattr lscpu lsipc lsmem lspgpot
lsblk lshw lslocks lsmod (insmod per installare un
    nuovo modulo)
lspci lsscsi

split #splitta un file in due o più parti!
cmp confronta due file
cut ps aux| cut -c 1-16,55-65 --output-delimiter=" - "

Esempio: stoppa il task genera.sh
#Il modo più semplice è: killall genera.sh
#Questo è un modo programmatico che lavora anche per altri scopi di
utilizzo
ps aux|grep "genera.sh"|grep -v "grep" #Con questo lo trovo ma devo
evidenziare il solo PID
ps aux|grep "genera.sh"|grep -v "grep"|cut -c 8-16 #può andare ma non è
così buona
ps aux|grep "genera.sh"|grep -v "grep"|awk -e '{ print $2; }'
awk
ps aux|grep "genera.sh"|grep -v "grep"|awk '{ print $2; }'
#in questo modo ho estratto da un file o dal risultato di un programma,
un elenco delle informazioni che mi servono
#Se il risultato è "semplice" (una sola riga con un solo numero o
similari)
ricordare che il back-quote ` , trasforma il risultato in un programma in
una stringa che può utilizzare come input di un altro programma

kill -9 `ps aux|grep "genera.sh"|grep -v "grep"|awk '{ print $2; }'` 

#Alcuni richiami da msdos
A=Ciao
echo $A #in msdos sarebbe echo %A
visualizza Ciao

#Voglio memorizzare nella variabile A l'elenco dei file che ho nella
cartella corrente
A='ls '

```

```

echo $A
0.dir 1661-0.txt american-english aplay archivio archivio.tgz a.txt
creacartelle.sh duplicati.txt.xz elenco.txt errori.log esempio ex1.sh
ex.tgz folder genera.sh index.html inizio.txt italiano lista.txt logo
.jpg new_file nuova nuova.tgz paperino.txt pippo.jpeg pippone.txt
pippo.txt pluto.txt provalnk random1.txt random2.txt random.txt
SherlockHolmes1.txt SherlockHolmes2.txt SherlockHolmes.txt stringhe.
txt tmp tuttil.txt tutti.txt vuota

#Preparare il file allfiles.txt
sudo find / -type f -name "*.txt" >allfiles.txt

#Supponiamo che un operatore abbia predisposto il file allfiles.txt
#contenente un insieme di file di testo di interesse specifico
#le chiedono di contare per ognuno di questi file , di quanti caratteri è
#composto!!!
sudo wc -c `cat allfiles.txt` 

xargs: prende in input un insieme di valori e li passa come parametri a
un comando
cat allfiles.txt | xargs wc

cat allfiles.txt | sudo xargs wc -c | head -n -1
#preso il risultato di questo comando
63 /etc/brltty/Input/lb/all.txt
2071 /etc/brltty/Input/mn/all.txt
1485 /etc/brltty/Input/tn/all.txt
1605 /etc/brltty/Input(mb/all.txt
66 /etc/brltty/Input/vr/all.txt
#calcolare la somma di tutti i campi numerici iniziali
sudo wc -c `cat allfiles.txt` | head -n -1| awk 'BEGIN {print "Inizio
conteggio"; tot=0}; {tot+=$1}; END {print "Totale: ", tot}' 

cat allfiles.txt | sudo xargs wc -c | head -n -1| awk 'BEGIN {print "
Inizio conteggio"; tot=0}; {tot+=$1}; END {print "Totale: ", tot}' 

tee #quando le serve di scrivere il risultato di un comando su un file e
#, contemporaneamente, di inviarlo allo stdout
ls | tee file.txt | wc
ls| tee uscita | tail -f

per fare la stessa cosa
ls >file.txt
wc file.txt

#in questo modo memorizza l'elenco defi file nel file files.txt e conta
quanti file sono

#trova tutti i file di dimensioni comprese tra 1M e 10MBytes e stampa le dimensioni
sudo find . -type f -size +1M -and -size -10M -exec wc -c {} \;

#esempio: a partire dal file BPD_Officier_Involved_Use_Of_Force.csv
#stampare in modo ordinato e senza ripetizioni
tutti i CC presenti
egrep -o "[0-9]{3,3}[A-Z][0-9]{5,5}" BPD_Officier_Involved_Use_Of_Force.csv | sort | uniq -c
##OK

```

```

cut -d, -f 6,7 BPD_Officier_Involved_Use_Of_Force.csv

#Se volessi aggiungere i numeri di riga davanti alle righe e stamparli
grep -n "" BPD_Officier_Involved_Use_Of_Force.csv | sed '/:/s//,/,' | cut
-d, -f 1,7,8

awk -F, 'BEGIN {linea=0} {print linea++,$6,$7}''
BPD_Officier_Involved_Use_Of_Force.csv

```

### 3.58 Web Server

Installazione di APACHE

```

sudo apt install apache2

#è in esecuzione apache?
ps aux | grep apache

#Su che porta TCP è in ascolto apache?
LISTEN: in attesa che qualcuno si colleghi
ESTABLISHED: connessione stabilita

come collegarsi
http://10.211.55.128:80/

#di default apache ha la sezione htdocs (le pagine)
/var/www/html/

#Questa sua versione di apache, che velocità ha? Può farne un benchmark?
ab -n 10000 -c 128 http://127.0.0.1:80/
ab -n 10000 -c 128 http://127.0.0.1:80/
ab -n 10000 -c 128 http://127.0.0.1:80/
sudo netstat -anp --tcp | grep -c "TIME_WAIT"
14114

sudo netstat -anp --tcp | grep "TIME_WAIT" | wc -l
14114

#Quando il server chiude i socket, questi rimangono (TIME_WAIT)
#indisponibili per un certo tempo

#in un terminale inviare
ab -n 10000000 -c 128 http://127.0.0.1:80/

#in un altro contare quante connessioni attive verso apache
sudo netstat -anp --tcp

#Entriamo nel merito di apache
#Apache appoggia configurazione e certificati in /etc/apache2
ll /etc/apache2/
-rw-r--r-- 1 root root 7224 giu 14 14:30 apache2.conf
drwxr-xr-x 2 root root 4096 ago 26 11:58 conf-available/
drwxr-xr-x 2 root root 4096 ago 26 11:58 conf-enabled/
-rw-r--r-- 1 root root 1782 mar 23 03:00 envvars
-rw-r--r-- 1 root root 31063 mar 23 03:00 magic
drwxr-xr-x 2 root root 12288 ago 26 11:58 mods-available/
drwxr-xr-x 2 root root 4096 ago 26 11:58 mods-enabled/
-rw-r--r-- 1 root root 320 mar 23 03:00 ports.conf
drwxr-xr-x 2 root root 4096 ago 26 11:58 sites-available/
drwxr-xr-x 2 root root 4096 ago 26 11:58 sites-enabled/

```

```

Aggiungiamo una porta TCP per apache
sudo vim /etc/apache2/ports.conf
Aggiungo la porta 8097

Ora dev fare il restart di apache
sudo apache2ctl restart

#By default varie installazioni di apache creano un sito per ogni utente
    del sistema (tutti gli utenti che sono in /home possono accedere al
    loro sito personale digitando http://dominio/~username)

#Attiviamo un nuovo sito
sudo mkdir /var/www/sito1
creiamo il file index.html in /var/www/sito1

nella cartella /etc/apache2/sites-available/ creiamo un nuovo sito ,
esempio , 001-default.conf
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-
-available/001-default.conf

edit la configurazione: vim /etc/apache2/sites-available/001-default.
conf

abilitare il sito

#!Il sito dei moduli per la 2.4
https://httpd.apache.org/docs/2.4/mod/

```

### 3.59 INSTALLIAMO DOCKER

```

sudo apt update
sudo apt upgrade
sudo apt install lsb-release ca-certificates apt-transport-https
software-properties-common -y
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [ arch=$(dpkg --print-architecture) signed-by=/usr/share/
keyrings/docker-archive-keyring.gpg] https://download.docker.com/
linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.
list.d/docker.list > /dev/null
sudo apt update
sudo apt install docker-ce
sudo groupadd docker
sudo usermod -aG docker $USER
docker ps

#E ora installamo nginx
https://hub.docker.com/_/nginx
docker run -it -p 8888:80 -v /var/www/sito2/:/usr/share/nginx/html:ro
nginx
-i: interattivo
-t: terminale (ctrl-c...)
-p port mapping la porta interna 80 corrisponde alla porta host 8888
-v collega file system interno a fs esterno (l'ultimo :ro significa che
è read only)
nginx è il container e per default prende l'ultimo disponibile (latest)

```

```

docker images #lista le immagini
docker container ls -a #lista i container
docker volume ls #lista i volumi

#Supponiamo di voler svuotare l'ambiente docker
docker images ls
docker image rm <id>
docker volume rm <id>
docker container rm <id>

```

### 3.60 Shell scripting

```

ex1.sh
#!/bin/bash
echo -n "Enter your name: "
read user_name
if [ -n "$user_name" ]
then
    echo "Hello $user_name!"
    exit 0
else
    echo "Greetings, nameless one!"
    exit 1
fi

#per testare se il comando ha dato errore: $?

#Un modo per eseguire script interni dopo find
find . -print0 | while IFS= read -r -d '' file; do dosomething "$file"; done

#Oppure richiamare script esterni!!!
errno -l #Lista tutti i possibili errori del SO linux
con echo $? mostra il codice di errore dell'ultimo comando eseguito

#memorizzare il codice di errore dell'ultimo comando eseguito nella variabile
lasterror
lasterror = $?

#Cancella dalla cartella nuova tutti i file di profondità max=1 e ne stampa il
# nome
find nuova -maxdepth 1 -type f -delete -print

for (( i=0; i<100; i++))
do
    head -c 32 /dev/random | sha512sum >>nuova/$i.rnd
done

#Crea 100 file con nome (0.rnd, 1.rnd, ...) nella cartella <nuova> e ci scrive
# dentro
#stringhe di 128 caratteri generate tramite numeri casuali e hash512 di questi

#Per ogni file della cartella <nuova> cancellare i caratteri < -> posti a fine
# riga
sed -i -e '/^-/s///' nuova/*

```

```

#Per ogni file della cartella <nuova> splittare la stringa casuale in
sottostringhe lunghe 16, separate da <>
#Esempio: file 0.rnd =
de8c996399f258d05fcc2e905fcf7dc9ebaedbe1fb04d4fa41b6396f0e5f36e34ea0fbfebfc52925d3cf11

Diventa 0.rnd = de8c996399f258d0 5fcc2e905fcf7dc9ebaedbe1fb04d4fa41b6396f0e5f36e34ea0fbfebfc52925d3cf11
a41b6396f0e5f36e 34ea0fbfebfc529 25d3cf11f01776d8 53d1e7dd20ac3d72 43
c8b57bc632e79c

#Prima possibile soluzione: data una variabile bash (es: str) che contiene una
stringa con ${str:3:10} prendo la stringa dal posto 3 per 10 caratteri

sed -i -e '/\(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ \(\.\.\.\.\.\.\.\)\ /s//\1 \2 \3 \4 \5 \6 \7 \8/' nuova/*

```

x = y x -eq y x is equal to y  
x != y x -ne y x is not equal to y  
x < a y x -lt y x is less than y  
n/a x -le y x is less than or equal to y  
x > a y x -gt y x is greater than y  
n/a x -ge y x is greater than or equal to y  
-n x n/a x is not null  
-z x n/a x is null

-d file file exists and is a directory  
-e file file exists  
-f file file exists and is a regular file  
-r file User has read permission on file  
-s file file exists and is not empty  
-w file User has write permission on file  
file1 -nt file2 file1 is newer than file2  
file1 -ot file2 file1 is older than file2

```

case $message_level in
    0) message_level_text="Error" ;;
    1) message_level_text="Warning" ;;
    2) message_level_text="Info" ;;
    3) message_level_text="Debug" ;;
    *) message_level_text="Other"
esac
echo "${message_level_text}: $message_itself

while read line; do
    echo "$((counter++)): $line"
done

```

### 3.61 adduser

man adduser

```
sudo adduser franco #esempio per creare  
sudo adduser franco user #esempio per aggiungere un gruppo
```

### 3.62 bit, byte e affini

Un bit è l'unità minima di informazione: #t/#f, true/false, 0/1

Un bit nei computer è rappresentato da un solo 1 o un solo 0

Byte: un byte è un gruppetto di bit. Nei computer a causa della storia dell'elettronica, dell'informatica ecc, il minimo gruppetto di bit è fatto da 8 bit

Quindi un Byte è una cosa del tipo 10100111

La memoria dei computer è fatta di celle grandi 1 byte. A volte sono anche esse organizzate in gruppetti

- WORD : 16 bit oppure 2 Byte
- DWORD (long word) : 32 bit oppure 4 byte
- LONG LONG WORD : 64 bit oppure 8 byte
- WORD128 : 128 bit oppure 16 byte

Oggi come oggi per motivi di efficienza e velocità le memorie dei PC e dei server sono organizzate in LONG LONG WORD oppure in WORD128.

Un PC con una memoria organizzata su base LONG LONG WORD, può leggere e scrivere in memoria numeri grandi fino al massimo intero rappresentabile con 64 bit ( $2^{64}$ )

Come faccio sotto terminale linux a eseguire in modo interattivo calcoli aritmetici e matematici in genere?

- bc (-l se voglio usare anche i decimali e le operazioni scientifiche e trigonometriche)

bc -l

```
s(10) => seno di 10 radianti  
sqrt(2) => radice quadrata di 2  
l(2) => logaritmo naturale di 2  
e(2) => e di eulero elevata al quadrato (2.718  
28182845904523536)
```

le cifre decimali di default sono 20 . se vuole cambiare scale=120 => mette a 120 le cifre decimali!!!

Esempio:

```
scale=500  
sqrt(2^500+sqrt(2))
```

### 3.63 ufw User firewall

man ufw

<https://www.cyberciti.biz/faq/ubuntu-22-04-lts-set-up-ufw-firewall-in-5-minutes/>

```
sudo ufw status  
sudo ufw default allow outgoing  
sudo ufw default deny incoming
```

```
grep IPV6 /etc/default/ufw  
sudo ufw allow ssh  
##ssh non è conosciuto in quanto protocollo SSH ma poiché è listato nel  
file /etc/services  
equivale a quanto scritto in service  
sudo ufw allow 22/tcp
```

NB: i firewall in genere non bloccano il protocollo di comunicazione ma bloccano tutte le comunicazioni su una specifica porta TCP

```

sudo ufw limit ssh #Limita sulla porta traffico , velocità , dimensioni
    dei dati , frequenza di accesso , numero di connessioni contemporanee ,
    ..

sudo ufw enable

#ora abilitiamo le porte di apache (ricorda che in /etc/apache2/ports.conf)
conf) abbiamo abilitato 80 e 8097
sudo ufw allow 80/tcp
sudo ufw allow 8097/tcp
sudo ufw reload

nmap 10.211.55.128
riporta tre porte aperte

Ora vorrei bloccare la posta 80 in inbound
sudo ufw deny 80/tcp
sudo ufw reload

#Modifichiamo le schede di rete , aggiungiamone una nuova (con ip addr
individua la scheda di rete da modificare)
sudo ip addr add 10.211.55.200 dev ens33

#Ora modifichiamo apache per attivare l'accesso solo sull'IP primario
(10.211.55.128)
vim /etc/apache2/ports.conf

```

### 3.64 vim

man vim

vim è un editor per linea di tipo testuale al quale è stata aggiunta con  
gvim un'interfaccia grafica  
vim ha due modi di funzionamento  
– comando  
– insert  
per passare dalla modalità insert alla modalità comando si digita il tasto ESC  
per andare nella insert si possono digitare molti comandi i, a, o, O, ...

```

Inizio riga: 0
Fine riga: $
Fine file: G
Inizio file: 1G
Seconda riga: 2G
singoli caratteri: frecette oppure hjkl
andare avanti di una word: w
tornare indietro di una word: b
andare a fine parola: e

copia e incolla
per copiare la riga corrente: yy
per incollare , sotto al cursore: p oppure Np
per incollare , sopra al cursore: P oppure NP

per copiare uno o più caratteri

```

- y<spazio> yN<spazio>  
per copiare parole  
- yw yNw
- i: per inserire un carattere alla posizione del cursore  
I: inserisce a inizio riga  
a: inserisce a destra del carattere corrente  
A: appende in fondo alla riga
- O: Aprire una nuova riga sopra la riga corrente  
o: Aprire una nuova riga sopra la riga corrente
- :! Eseguire un comando esterno
- !! eseguire un comando esterno e inserire il risultato nell'editor
- Cancellazione  
x: cancella il carattere sotto il cursore (e lo mette nel buffer di yank )  
X: cancella il carattere precedente
- parole, righe ecc  
dd: cancella la riga corrente  
D: cancella la riga senza togliere lo spazio di riga  
dw: cancella la parola a destra  
dG dalla posizione corrente fino a fine file
- Per cercare  
/<pattern da cercare/>  
n: il prossimo  
N: torna indietro
- d/<pattern>invio  
. : ripete l'ultimo comando dato  
u: undo  
U: undo della riga
- :g

### 3.65 nc NetCat

man nc

```
nc -l 1234 #su una macchina
nc -N 127.0.0.1 1234 #seconda macchina
```

### 3.66 ssh tunneling

man ssh

```
ssh -N -L 192.168.214.150:8080:192.168.214.133:80 user@192.168.214.133
```

### 3.67 tasksel

man tasksel

le consente di selezionare un tipo di installazione

### 3.68 xxd

man xxd

```
xxd le mostra un file binario in formato esadecimale e ascii (dump)  
usualmente: xxd <file> | less
```

### 3.69 openssl

man openssl

```
Il wiki di openssl: https://wiki.openssl.org/index.php/  
Command_Line_Utils  
Il manuale dei comandi: https://www.openssl.org/docs/manmaster/man1/  
Per aggiungere alternative names: <alternative names in openssl.pdf>
```

- OpenSSL Command to Generate Private Key  
openssl genrsa -out yourdomain.key 2048

- OpenSSL Command to Check your Private Key  
openssl rsa -in privateKey.key -check

- openssl to generate the public key  
openssl rsa -in privata.key -pubout

```
#Per cifrare un messaggio con una chiave pubblica  
openssl pkeyutl -encrypt -in messaggionuovo.txt -pubin -inkey pubblica-marco.pem  
-out messaggionuovo-permarco.cfr  
openssl pkeyutl -decrypt
```

- OpenSSL Command to Generate CSR

If you have generated Private Key:

```
openssl req -new -key yourdomain.key -out yourdomain.csr
```

- Once you execute this command, you'll be asked additional details. Enter them as below:

Country Name: 2-digit country code where your organization is legally located.

State/Province: Write the full name of the state where your organization is legally located.

City: Write the full name of the city where your organization is legally located

Organization Name: Write the legal name of your organization.

Organization Unit: Name of the department (Not Compulsory. Press Enter to skip)

Common Name: Your Fully Qualified Domain Name (e.g., www.yourdomainname.com.)

Email: The email ID through which certification will take place (Not Compulsory. Press Enter to skip)

- If you haven't generated your Private Key yet:

- This command will generate CSR and private key in a single shot.

```
openssl req -new \  
-newkey rsa:2048 -nodes -keyout yourdomain.key \  
-out yourdomain.csr \  
-subj "/C=US/ST=Florida/L=Saint Petersburg/O=Your Company, Inc./OU=IT/CN  
=yourdomain.com"
```

Country Name: 2-digit country code where your organization is legally located.

State/Province: Write the full name of the state where your organization is legally located.

City: Write the full name of the city where your organization is legally located

Organization Name: Write the legal name of your organization.

Organization Unit: Name of the department

Common Name: Your Fully Qualified Domain Name

– OpenSSL Command to Check CSR

```
openssl req -text -noout -verify -in CSR.csr
```

– OpenSSL Commands to Convert Certificate and Key Files

OpenSSL commands to convert PEM file :

– Convert PEM to DER

```
openssl x509 -outform der -in certificate.pem -out certificate.der
```

– Convert PEM to P7B

```
openssl crl2pkcs7 -nocrl -certfile certificate.cer -out certificate.p7b -certfile CACert.cert
```

– Convert PEM to PFX

```
openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt -certfile CACert.crt
```

– OpenSSL commands to convert DER file

– Convert DER to PEM

– To convert certificate file :

```
openssl x509 -inform DER -in yourdomain.der -outform PEM -out yourdomain.crt
```

– To convert private key file :

```
openssl rsa -inform DER -in yourdomain.key.der -outform PEM -out yourdomain.key
```

– OpenSSL commands to convert P7B file

– Convert P7B to PEM

```
openssl pkcs7 -print_certs -in certificate.p7b -out certificate.cer
```

– Convert P7B to PFX

```
openssl pkcs7 -print_certs -in certificate.p7b -out certificate.cer
```

```
openssl pkcs12 -export -in certificate.cer -inkey privateKey.key -out certificate.pfx -certfile CACert.cert
```

– OpenSSL commands to convert PKCS#12 (.pfx) file

– Convert PFX to PEM

– To convert certificate file :

```
openssl pkcs12 -in certificate.pfx -out certificate.cer -nodes
```

– To convert private key file :

```
openssl pkcs12 -in yourdomain.pfx -nocerts -out yourdomain.key -nodes
```

– OpenSSL Command to Check a certificate

```
openssl x509 -in certificate.crt -text -noout
```

– OpenSSL Command to Check a PKCS#12 file (.pfx file)

```
openssl pkcs12 -info -in keyStore.p12
```

#Un uso semplificato per cifra , firma e generazione di certificati WEB  
/usr/lib/ssl/misc/CA.pl –help

```
/usr/lib/ssl/misc/CA.pl –newcert
```

Rispondere alle domande. Come common name mettere www.linuxbase.com

```

ora ci sono due file: newkey.pem, newcert.pem

#Per esaminare in dettaglio un certificato
openssl x509 -text -noout -in newcert.pem

Se emettitore e subject sono gli stessi ,allora è un self signed certificate

#Svuotiamo la cartella corrente e questa diventerà la cearrella in cui
archiviamo tutti i certificati che genereremo

openssl req -new -keyout ./demoCA/private/cakey.pem -out ./demoCA/careq.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:IT
State or Province Name (full name) [Some-State]:LAZIO
Locality Name (eg, city) []:ROME
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Corso Linux Base
Organizational Unit Name (eg, section) []:linuxbase
Common Name (e.g. server FQDN or YOUR name) []:Root CA linuxbase

#generiamo una nuova richiesta di certificato firmato da questa CA root
/usr/lib/ssl/misc/CA.pl -newreq

Per esaminare la richiesta appena inoltrata:
openssl req -text -noout -in newreq.pem

Firma della richiesta da parte della CA locale (demoCA)
/usr/lib/ssl/misc/CA.pl -sign

#Copiare i certificati generati in /etc/ssl certs e private
#Modificare /etc/apache2/sites-available/default-ssl...
    - inserire il certificato e la chiave privata
        !! Ricordare di togliere la password dalla chiave privata (
            openssl rsa -out nopwd.pem -in key.pem)
    - inserire il certificato della root ca
    - Modificare eventuale root del sito
#abilitare il sito default.ssl (a2ensite ...)
#abilitare il modulo ssl (a2enmod ssl)

#se necessario, aprire la porta 443 sul firewall ufw enable 443/tcp, e poi il
reload delle regole!

#Ricordarsi di utilizzare netstat -anp --tcp | grep LISTEN per verificare che
apache sia effettivamente in listen
#Esempio di comandi che lei ha utilizzato:
1265 /usr/lib/ssl/misc/CA.pl -newcert
1266 ls
1267 openssl x509 -text -noout -in newcert.pem
/usr/lib/ssl/misc/CA.pl
1273 head /usr/lib/ssl/misc/CA.pl
1274*

```

```

/usr/lib/ssl/misc/CA.pl --newca
/usr/lib/ssl/misc/CA.pl --newreq
/usr/lib/ssl/misc/CA.pl --sign
mv newcert.pem linuxbase-cert.pem
mv newkey.pem linuxbase_key.pem
sudo vim /etc/apache2/sites-available/default-ssl.conf
sudo cp linuxbase-cert.pem /etc/ssl/certs/
sudo cp linuxbase-key.pem /etc/ssl/private/
sudo vim /etc/apache2/sites-available/default-ssl.conf
apache2ctl restart
sudo apache2ctl restart
sudo ufw status
sudo ufw allow 443/tcp
sudo netstat -anp --tcp
sudo a2ensite default-ssl.conf
sudo vim /etc/apache2/ports.conf
sudo apache2ctl restart
sudo a2enmod ssl
sudo vim /etc/apache2/sites-enabled/default-ssl.conf
sudo mkdir /etc/apache2/ssl.crt
sudo cp demoCA/cacert.pem /etc/apache2/ssl.crt/server-ca.crt
openssl x509 -text -noout -in /etc/ssl/certs/linuxbase-cert.pem
sudo apache2ctl restart
sudo ufw status

#Per ottenere un certificato digitale di sito web gratuitamente
#e riconosciuto da tutti i browser
https://letsencrypt.org/docs/

#Inoltre (ma questo funziona quasi come quello che abbiamo fatto noi) Il
#certificato root va inserito nel browser!
http://www.cacert.org/

```

### 3.70 DNS

```

https://www.digitalocean.com/community/tutorials/how-to-configure-bind-
as-a-private-network-dns-server-on-ubuntu-22-04
https://technologyrss.com/how-to-install-and-configure-dns-bind9-on-
ubuntu-22-04-server/

```

### 3.71 samba

man samba

Un manualetto in linea: <https://computingforgeeks.com/install-and-configure-samba-server-share-on-ubuntu/>

```

#Installare samba (apt install samba)

#Creare una cartella con tutti gli share
sudo mkdir -p /home/share
sudo chmod 777 /home/share

sudo vim /etc/samba/smb.conf
[global]

# Configure correct UTP
unix charset = UTF-8

```

```

# Change this to the workgroup/NT-domain name your Samba server
# will be part of
workgroup = WORKGROUP
bind interfaces only = yes

# Set share configuration at the end
[Docs]
path = /home/share
writable = yes
guest ok = yes
guest only = yes
create mode = 0777
directory mode = 0777

#Inoltre:
# Un-comment the following (and tweak the other settings
# below to suit)
# to enable the default home directory shares. This will
# share each
#[homes]
comment = Home Directories
browseable = yes

# By default, the home directories are exported read-
# only. Change the
# next parameter to 'no' if you want to be able to write
# to them.
read only = no

# File creation mask is set to 0700 for security reasons
# . If you want to
# create files with group=rw permissions, set next
# parameter to 0775.
create mask = 0700

# Directory creation mask is set to 0700 for security
# reasons. If you want to
# create dirs. with group=rw permissions, set next
# parameter to 0775.
directory mask = 0700

# By default, \\server\username shares can be connected
# to by anyone
# with access to the samba server.
# Un-comment the following parameter to make sure that
# only "username"
# can connect to \\server\username
# This might need tweaking when using external
# authentication schemes
valid users = %S

#Set group ownership of Samba share directory to sambashare group.
sudo chgrp sambashare /home/share

#Se vuole può creare nuovi utenti che non hanno login sotto linux
#Si ricordi che gli utenti linux già sono condivisi sotto samba!!!
#Create samba share users. Replace user1 with samba share user to create
.

```

```

sudo useradd -M -d /home/share/user1 -s /usr/sbin/nologin -G sambashare
        user1
sudo mkdir /home/share/user1
sudo chown user1:sambashare /home/share/user1
sudo chmod 2770 /home/share/user1

Set the user password and confirm .
sudo smbpasswd -a user1
        New SMB password:
        Retype new SMB password:
        Added user user1.

#Enable samba account after setting the password using the command
sudo smbpasswd -e user1
        Enabled user user1.

#Inoltre
#We created a user's home directory inside Samba share parent directory
        and set ownership to the samba share group. Let's create another user
.

sudo useradd -M -d /home/share/smbadmin -s /usr/sbin/nologin -G
        sambashare smbadmin
sudo mkdir /home/share/smbadmin
sudo smbpasswd -a smbadmin
sudo smbpasswd -e smbadmin
sudo chown smbadmin:sambashare /home/share/smbadmin
sudo chmod 2770 /home/share/smbadmin

#Configure Secure Samba share. per utenti solo samba

[user1]
path = /home/share/user1
read only = no
browseable = no
force create mode = 0660
force directory mode = 2770
valid users = @user1 @sambashare

[smbadmin]
path = /home/share/smbadmin
read only = no
browseable = yes
force create mode = 0660
force directory mode = 2770
valid users = @sambashare @smbadmin

Restart samba daemon after making the changes.
sudo systemctl restart smbd nmbd

If you have an active firewall , run:
sudo ufw allow 'Samba'

### Install Samba client on Ubuntu / Debian ####
sudo apt update
sudo apt -y install smbclient cifs-utils

### Install Samba client on CentOS / RHEL / Fedora ####
sudo yum -y install samba-client cifs-utils
sudo dnf -y install samba-client cifs-utils

```

```
#Per elencare gli share attivi
/usr/bin/smbclient -L 10.211.55.128 -U user

Accessing Samba share ? Temporarily
$ smbclient //sambaserver/share -U sambauusername
```

Example:

```
$ smbclient //192.168.122.52/user1 -U user1
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\user1's password:
Try "help" to get a list of possible commands.
smb: \> list
0: server=192.168.122.52, share=user1
smb: > ls
.
D          0 Sun Jul 14
    01:10:39 2019
..
D          0 Sun Jul 14
    00:49:13 2019
latest.zip
N 12121295 Tue Jun 18
    10:52:03 2019

9544896 blocks of size 1024. 7918732 blocks available

smb: > pwd
Current directory is \192.168.122.52\user1\
```

You can mount a samba share to a directory in your local Linux system using the mount and cifs type option.

```
$ mkdir -p ~/mounts/shares
$ mount -t cifs -o username=user1 //192.168.122.52/user1 ~/mounts/shares
$ df -h
Filesystem      Size   Used  Avail Use% Mounted on
/dev/vda1        40G   2.9G   38G   8% /
devtmpfs         488M     0   488M   0% /dev
tmpfs            496M     0   496M   0% /dev/shm
tmpfs            496M   6.7M   489M   2% /run
tmpfs            496M     0   496M   0% /sys/fs/cgroup
tmpfs            100M     0   100M   0% /run/user/1000
//192.168.122.52/user1  9.2G   1.6G   7.6G  18% /home/computingforgeeks/mounts/shares
```

### 3.72 NFS

man nfs

<https://linuxhint.com/install-and-configure-nfs-server-ubuntu-22-04/>  
<https://ibexoft.com/mounting-nfs-share-on-windows-10-with-write-access/>

```
#installazione
sudo apt update
sudo apt dist-upgrade
sudo apt install nfs-kernel-server

sudo mkdir /home/nfsshare
sudo chown -R nobody:nogroup /home/nfsshare/
sudo chmod 777 /home/nfsshare/
sudo vim /etc(exports
```

```

Aggiungere: /home/nfsshare 10.211.55.132(rw,sync,no_subtree_check)

poi attivare la nuova esportazione
sudo exportfs -a

#Dov'è il log di NFS?
#Sempre nello stesso punto
/var/log/xxx

#!Si ricordi di attivare il transito dentro il firewall, se il firewall
è attivo !!

showmount mostra i filesystem montati da macchine remote

#Su una macchina windows:
Da esplora risorse: \\10.211.55.128\home\nfsshare

#Su una macchina linux: (ricordarsi di modificare il file /etc/exportfs
per aggiungere il nuovo client)
(sudo apt install nfs-common)
da console: sudo mount -o rw,hard,intr,bg 10.211.55.128:/home/nfsshare
/mnt

```

### 3.73 RFC

Sono le cosiddette Request For Comment che con il corso del tempo e degli accordi tra organismi di gestione degli standard internazionali (ISO, FIPS, ...), posso essere tramutati in SRD (standard).  
 Tutto Internet (tutti i servizi o protocolli definiti per Internet) sono stati descritti tramite RFC.  
 Ad oggi pochissimi RFC sono stati congelati come STD, la maggior parte degli RFC resta in tale stato.  
 Ad esempio HTTP, SMTP, DNS, ... sono ancora degli RFC.  
 Gli RFC nascono all'incirca negli anni 70 con la specifica del TCP/IT.  
 Jon Postel è l'editor dei primi rfc che poi sono stati modificati da altri.

### 3.74 apt

man rm

Dove sono gli indirizzi dei repository dai quali scarico i pacchetti e gli aggiornamenti  
 /etc/apt/sources.list  
 /etc/apt/sources.list.d/\*

### 3.75 ldd

man ldd

```

ldd – print shared object dependencies

– Esempio: compilo un file da github
https://github.com/halturin/multitail
git clone https://github.com/halturin/multitail.git

#Necessario installare gli ambienti di sviluppo

```

```

sudo apt install make g++ pkg-config cppcheck libncurses-dev

#eseguo un ubnuntu vecchio
docker run -it --network host -v "$PWD":/data ubuntu:18.04 /bin/bash -i
- installo lsb-core (apt install lsb-core)

mi sposto nella cartella /data
eseguo ./multitail

!! Errore ./multitail: error while loading shared libraries: libpanelw.so
.6: cannot open shared object file: No such file or directory

#con ldd trovo le librerie dinamiche necessarie
ldd multitail
    linux-vdso.so.1 (0x00007ffefdf000)
    libpanelw.so.6 => not found
    libncursesw.so.6 => not found
    libtinfo.so.6 => not found
    libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f52d0ade000
        )
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f52d06ed000
        )
    /lib64/ld-linux-x86-64.so.2 (0x00007f52d0e7c000)

Me le copia dalla macchina host (altro terminale)
cp /lib/x86_64-linux-gnu/libpanelw.so.6 .
cp /lib/x86_64-linux-gnu/libncursesw.so.6 .
cp /lib/x86_64-linux-gnu/libtinfo.so.6 .

#Sulla macchina target
LD_LIBRARY_PATH=. ./multitail

#Se continua a dare errori in quanto non trova GCC..., allora
    copiare anche
cp /lib64/ld-linux-x86-64.so.2 .

#Ora devo mandare in esecuzione multitail ma dicendogli che
    alcune librerie dinamiche sono nella cartella corrente
LD_LIBRARY_PATH=. ./ld-linux-x86-64.so.2 ./multitail

```

### 3.76 cygwin

<https://www.cygwin.com/>  
 è un sottosistema linux per windows. Nulla a che vedere con il pacchetto  
 ubuntu per windows. è un insieme di programmi aggiornabili riscritto  
 per windows.

### 3.77 stty

man stty

```

stty intr ^P
stty sane ####Per rpristinare

```

### 3.78 Installazione di pacchetti da codice sorgente github

Supponiamo di voler installare un software da github sulla macchina  
 linux

Normalmente i software usano due o al pi3 modelli di compilazione  
– configure \*è uno dei più utilizzati  
– cmake  
– pacchetti java o python di varia natura.

Proviamo a installare <https://download.libsodium.org/libsodium/releases/libsodium-1.0.18-stable.tar.gz>

```
wget https://download.libsodium.org/libsodium/releases/libsodium
```

```
-1.0.18-stable.tar.gz
```

```
#al termine:
```

```
decomprimere e poi
```

```
per decomprimere e far untar di un tgz oppure di un tar.gz  
gzip -c -d libsodium-1.0.18-stable.tar.gz|tar xf -  
cd libsodium
```

Spesso configure ha la necessità di essere costruito. A tal fine è  
presente un file che si chiama autogen.sh

Quindi se non è presente il file configure ed è presente il file  
configure.ac

, mettere in esecuzione autogen.sh

se non va

```
auto apt install autoconf autogen
```

```
./configure --prefix=/usr/local/libsodium  
make -j8
```

```
make check
```

```
make sample/samples
```

```
make pdf/doc
```

```
sudo make install
```

### 3.79 scp

man scp

```
scp <file , ... / folder (usare -r)> utente@ip:folder di destinazione
```

```
scp -r <folder> utente@ip:folder
```

```
-C per comprimere
```

Se deve trasferire cartelle molto grandi, allora meglio utilizzare le  
pipe e ssh

```
tar czf - elencofolder | ssh user@192.168.214.134 "tar xzf -"  
In questo modo senza creare file temporanei in locale o in remoto
```

### 3.80 rsync

man rsync

rsync consente backup incrementali e altre modalità di copia  
rsync copia solo i file modificati dall'ultima operazione di copia

### 3.81 chown, chgrp, chmod

man chown, chgrp, chmod

### 3.82 Elenco degli utenti del sistema

```
cat /etc/passwd

solo gli utenti con bash (due modi diversi)
grep "bash" /etc/passwd | cut -f 1 -d :
grep "bash" /etc/passwd | egrep -o "^[^:]+"
```

### 3.83 yes

man yes

```
yes stampa infiniti yes su stdout
yes stringa stampa infinite stringhe su stdout

yes Ciao | head -n 10000 >ciao.txt
scrive 10000 volte ciao nel file ciao.txt
```

### 3.84 Comandi vari

```
Per prendere solo la parte nome di un file:
for i in [pP]*; do echo ${i##%.*}; done #di tutti i file che iniziano per
# o P prendi solo il nome
for i in [pP]*; do echo ${i##*.}; done #di tutti i file che iniziano per
# o P prendi solo l'estensione
```

```
Questa si capisce da quello che stampa!!!
rename -n 'y/a-z/A-Z/;s/(P.*O)/$1A/;s/^/my_new_dir\//' p*.*
```

### 3.85 Setuid, Setgid and Sticky Bits

Setuid, Setgid and Sticky Bits are special types of Unix/Linux file permission sets that permit certain users to run specific programs with elevated privileges. Ultimately the permissions that are set on a file determine what users can read, write or execute the file. Linux provides more advanced file permissions that allow you to do more specific things with a file, or directory. Typically, these file permissions are used to allow a user to do certain tasks with elevated privileges (allow them to do things they normally are not permitted to do). This is accomplished with three distinct permission settings. They are setuid, setgid, and the sticky bit.

Symbolic Permission Notation

— no permission	
—x execute	#!!! per le cartelle significa "apertura consentita"
-w- write	
-wx write and execute	
r— read	
r-x read and execute	
rw- read and write	
rwx read, write and execute	

## Numeric Permission Notation

The numeric notation system uses the digits 1 through 7, with each corresponding to a different symbolic permission set:

0	— no permission
1	—x execute
2	-w- write
3	-wx write and execute
4	r— read
5	r-x read and execute
6	rw- read and write
7	rwx read, write and execute

rwx

000 => 0
001 => 1
010 => 2
011 => 3
100 => 4
101 => 5
110 => 6
111 => 7

## What Each Type Controls

Read: Display the contents of a file. You may be able to open it in text editors like Vim, but you will not be able to save any changes to the file unless you also have write permissions.

Write: Make changes to a file or folder/directory.

Execute: Execute (a.k.a. Run) a file. Examples include compiled binaries and shell scripts.

ESEMPIO: wall (man wall)

## What is Setuid?

Setuid is a Linux file permission setting that allows a user to execute that file or program with the permission of the owner of that file. This is primarily used to elevate the privileges of the current user. If a file is ?setuid? and is owned by the user ?root? then a user that has the ability to execute that program will do so as the user root instead of themselves. The most common example of this in Linux is ?sudo?. In this example, the user ?test? located the executable ?sudo? and did a full listing of it with the ?ls -l? command.

```
root@host [~]# id  
uid=1002(test) gid=1002(test) groups=1002(test)
```

```
root@host [~]# which sudo  
/usr/bin/sudo
```

```
root@host [~]# ls -l /usr/bin/sudo  
-rwsr-xr-x 1 root root 136808 Jan 31 13:37 /usr/bin/sudo
```

```
root@host [~]#
```

If you look at the permissions level of the ?sudo? executable, you can see the ?s? in the permissions for the user where normally there would be an ?x?. Also, notice that this file is owned by the user ?root? (the super-user) and that the file is executable by the world (the last ?x? in the permissions). This indicates that when a user executes this program, the operating system

will execute that file not as the user ?test?, but as the user ?root'. In the matter of using the ?sudo? command, this allows a normal user to perform elevated system functions without having to log in as the root user.

### How Do I Set Up Setuid?

Setting the ?setuid? permission is as simple as setting any other permission in Linux. The file ownership is modified using the command. An example command to set this would be as follows.

```
root@host [~]# chmod u+s <filename>
```

In this example, we will create a file called ?myfile? using the command ?touch? and then we will examine its permissions with the ?ls -l' command.

```
root@host [~]# touch myfile
root@host [~]# ls -l myfile
-rw-rw-r-- 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
```

Notice that the file does not have the execute permissions for user, group, or world. We will add the setuid bit as seen below.

```
root@host [~]# chmod u+s myfile
root@host [~]# ls -l myfile
-rwSr-wr-- 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
```

This output looks a little different from what we were expecting. The lowercase ?s? we were looking for is now a capital ?S.? This signifies that the setuid IS set, but the user that owns the file does not have execute permissions. We can add that permission using the ?chmod u+x? command.

```
root@host [~]# chmod u+x myfile
root@host [~]# ls -l
total 0
-rwsrw-r-- 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
```

What is Setgid?

Setgid, when used on files, is very similar to setuid. A process, when executed, will run as the group that owns the file. A typical example of a file that uses this is the ?crontab? command.

```
root@host [~]# which crontab
/usr/bin/crontab
root@host [~]# ls -l /usr/bin/crontab
-rwxr-sr-x 1 root crontab 36080 Apr 5 2016 /usr/bin/crontab
root@host [~]#
```

How Do I Set Up Setgid?

Similar to ?setuid?, ?setgid? is inserted with the ?chmod g+s? command. Let's create a new file called ?myfile2?.

```
root@host [~]# touch myfile2
root@host [~]# ls -l myfile2
-rw-rw-r-- 1 test test 0 Mar 2 19:30 myfile2
root@host [~]#
```

Now we will run the 'chmod g+s' command and review the results.

```
root@host [~]# chmod g+s myfile2
root@host [~]# ls -l myfile2
-rw-rwSr-- 1 test test 0 Mar 2 19:30 myfile2
root@host [~]#
```

Again we see the capital ?S? is set, but we can modify that.

```
root@host [~]# chmod g+x myfile2
root@host [~]# ls -l
total 0
-rwsrw-r— 1 test test 0 Mar 2 17:59 myfile
-rw-rwsr— 1 test test 0 Mar 2 19:30 myfile2
root@host [~]#
Setgid on Directories
```

Applying the setgid permission on a directory has a different behavior. A directory that has ?setgid? on it will cause all files that are created in that directory to be owned by the group of the directory as opposed to the group of the owner. First, we create a directory.

```
root@host [~]# mkdir mydir
root@host [~]# ls -ld mydir
drwxrwxr-x 2 test test 4096 Mar 2 19:36 mydir
root@host [~]#
```

Then we change the group ownership of the directory by using the 'chgrp' command, and then we can add the ?setgid? permission like before.

```
root@host [~]# chgrp test2 mydir/
root@host [~]# chmod g+s mydir
root@host [~]# ls -ld mydir/
drwxrwsr-x 2 test test2 4096 Mar 2 19:36 mydir/
root@host [~]#
```

Let's test it out by creating a file in that directory. All other files in this tutorial were created this way and had ?test? as the group. Because ?setgid? is set on the directory and it is owned by group ?test2?, this file will get ?test2? as its group.

```
root@host [~]# touch mydir/myfile3
root@host [~]# ls -l mydir/myfile3
-rw-rw-r— 1 test test2 0 Mar 2 19:59 mydir/myfile3
root@host [~]#
```

#### What Is A Sticky Bit?

The final special permission is the ?sticky bit.? When this is set on a directory, the files in that directory can only be removed by the owner. A typical use of this is ?/tmp/.? The /tmp directory can be written to by any user, but other users cannot delete the files of others.

```
root@host [~]# ls -ld /tmp
drwxrwxrwt 8 root root 4096 Mar 2 20:17 /tmp
root@host [~]#
```

Notice that /tmp can be written to by everyone but has the ?t? in place of the ?x? at the end of the permissions list. This means it has the sticky bit.

#### How Do I Set Up A Sticky Bit?

The sticky bit is set with ?chmod +t? command.

```
root@host [~]# mkdir mydir2
root@host [~]# ls -ld mydir2
drwxrwxr-x 2 test test 4096 Mar 2 20:17 mydir2
root@host [~]# chmod +t mydir2
root@host [~]# ls -ld mydir2
drwxrwxr-t 2 test test 4096 Mar 2 20:17 mydir2
root@host [~]#
```

#### Setting Special Permissions With Number Notation

You may remember from the definitions above that permissions can be set with a series of three numbers. The numbers represent the permissions for owner, group, and world, respectively. To determine the number you want to set, you can use x=1, w=2, and r=4. You add the numbers together to get the permission

number. If we wanted to have read, write, and execute permissions, we would use 7. Read and write would be 6. Just read is 4. An example to set the file to read, write, and execute for owner, read and execute for group and world would look like this:

```
root@host [~]# chmod 755 myfile
root@host [~]# ls -l myfile
-rwxr-xr-x 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
For the special permissions, you prepend these numbers with another number where 4 is setuid, 2 is setgid, and 1 is the sticky bit. The following commands are all the same (assuming the file has the permissions we set above).
```

```
root@host [~]# chmod 4755 myfile
root@host [~]# chmod u+s myfile
root@host [~]# ls -l myfile
-rwsr-xr-x 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
root@host [~]# chmod 2755 myfile
root@host [~]# chmod g+s myfile
root@host [~]# ls -l myfile
-rwxr-sr-x 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
```

```
root@host [~]# chmod 1755 mydir
root@host [~]# chmod +t mydir
root@host [~]# ls -ld mydir
drwxr-sr-t 2 test test2 4096 Mar 2 19:59 mydir
root@host [~]#
Removing Special Permissions
```

To remove special permissions, we can use the same chmod commands with a ?-? instead of a ?+?.?

```
root@host [~]# chmod u-s myfile
root@host [~]# chmod g-s mydir
root@host [~]# chmod -t mydir2
root@host [~]# ls -l
total 8
drwxr-xr-x 2 test test2 4096 Mar 2 19:59 mydir
drwxrwxr-x 2 test test 4096 Mar 2 20:17 mydir2
-rw-r-xr-t 1 test test 0 Mar 2 17:59 myfile
root@host [~]#
```

## 3.86 sockets

### 1. Overview

Sockets are endpoints for communication. Importantly, sockets enable communication between programs running on separate machines or programs running on the same machine.

In this tutorial, we'll learn about TCP/IP sockets and Unix sockets. We'll also look at the difference between the two.

I socket consentono sia su rete, sia sulla stessa macchina di fare Interprocess communication!!!

### 2. TCP/IP Sockets

TCP/IP stands for Transmission Control Protocol/Internet Protocol. These sockets facilitate communication between two computers. As a result, the socket

consists of the IP address and the port number that the machines use to transmit the data. In fact, all applications transmitting data use the socket to send and receive data.

In particular, TCP/IP supports three types of sockets: stream sockets, datagram sockets, and raw sockets.

## 2.1. Stream Sockets

Corrispondono al meccanismo della telefonata (il client richiede una connessione, il server gli dice OK, il client manda dati, il server risponde, a un certo punto uno dei due peer riattacca)

Caratterizzato da

IP sorgente: porta sorgente: IP destinazione: porta destinazione

$2^{32}$  ip src:  $2^{16}$  psrc : il numero di IP assegnati alla sua macchina: le porte su cui tali IP sono in ascolto

Per il tcp aggiungo: 4 byte di sequence, 4 byte di ack: totale è leggermente superiore a 20 byte (quindi 100 come limite superiore è più che sufficiente)

Sono almeno  $2^{48}$  (sono quelli che provengono da Internet) \* i suoi IP e le sue porte

Il vantaggio delle comunicazioni è che non lasciano traccia. Al massimo deve ricordarsi quante ne ha aperte. e quindi è una dimensione lineare rispetto alla quantità di connessione aperte

Supponendo che una singola connessione costi 100 byte di memoria sulla sua macchina, allora 1000000 di connessioni contemporaneamente aperte consuma al più 100 MByte. 10000000 (centomilioni) di comunicazioni contemporaneamente aperte consumano di memoria al più 10GByte.

Stream sockets are the most commonly used type of communication transport protocol over TCP/IP. Also, these sockets provide a reliable and nearly error-free data pipe between two endpoints. In addition, the data pipe can be bidirectional. Both endpoints transmit and receive through the socket.

Moreover, stream sockets are also relied upon to deliver sequenced and unduplicated data. This means packets are delivered in the order they are sent and that computers receive a particular packet only once.

However, if packets arrive on the physical network out of order, the network adapter and the host operating system ensure they are assembled in the correct sequence for processing by an application.

For example, a use case for stream sockets is a web server like Apache handling HTTP requests on port 80. For HTTP, a socket address would be similar to 23.227.38.65:80.

## 2.2. Datagram Sockets

Sono le comunicazioni di tipo UDP. Significa che non c'è una "telefonata" tra i due peer.

Il peer A trasmette sulla rete un singolo pacchetto UDP con destinazione peer B, se il peer B lo riceve, tutto ok, altrimenti tutto ok lo stesso. Non conta che il ricevitore riceva il pacchetto. È ovvio che se sta trasferendo un file (ad esempio) il protocollo Datagram non funziona in quanto il ricevitore potrebbe non aver ricevuto tutto il file che il trasmettitore ha trasmesso.

Però funziona benissimo nella trasmissione di un brano musicale in temporeale, di un movie in tempo reale: in tal caso se il ricevitore perde un singolo pacchetto, la qualità del filmato o della musica non viene compromessa.

In questo caso il cosiddetto SERVER (quello che riceve la richiesta datagram) non deve archiviare nulla rispetto alla comunicazione poiché i datagram non creano un canale stabile di comunicazione (tipo la telefonata). Un esempio di datagram è quando in mezzo alla folla A dice qualcosa a gran voce a B. Se B

lo sente , ok , altrimenti A urla di nuovo .

Datagram sockets define a connection-less service . In fact , these sockets use the User Datagram Protocol ( UDP ) to encapsulate and transmit data . The packets are sent independently , and there are no guarantees . This means the packets can be lost , duplicated , and arrive out of order .

In addition , the size of the packets is limited to the size that can be sent in one transaction . There is no disassembly or assembly of packets in datagram sockets . To transmit data successfully , applications that use datagram sockets must have their own error handling and data ordering logic .

For example , Domain Name System ( DNS ) servers use the User Datagram Protocol to send and receive requests for domain names . Since the DNS server ? s default port is 53 , a socket address would be similar to 23.227.38.65:53 .

### 2.3. Raw Sockets

A raw socket is a type of network socket that enables a software application on the computer to send and receive packets from the network without using the computer ? s primary operating system . Note that raw sockets bypass the normal TCP/IP processing system and send packets to the specific user application .

Furthermore , raw sockets allow applications to directly access lower-level protocols such as the Internet Protocol ( IP ) . Along with direct access to lower-level protocols , raw sockets also receive un-extracted packets . Therefore , raw sockets are mainly used to test new protocol implementations .

## 3. Unix Sockets

Unix sockets , also known as Inter-process Communication ( IPC ) sockets , are data communication endpoints that allow bidirectional data exchange between processes running on the same computer .

These sockets can be stream-based or datagram-based . In addition , these sockets exchange data between processes directly in the operating system ? s kernel through files on the filesystem . Processes read and write to their shared socket files to send and receive data .

Lastly , Unix sockets are widely used by database systems that don ? t need to be connected to a network interface . For instance , on Ubuntu , MySQL defaults to using /var/run/mysqld/mysql.sock for communication with local clients .

## 4. Unix Sockets vs. TCP/IP Sockets

Let ? s look at the differences between Unix sockets and TCP/IP sockets :

Unix socket example :

```
nc -U pippsock -u -l #Crea uno unix socket ( il file pippsock )
```

```
nc -U pippsock -u #Si connette come client a uno unix socket
```

```
ssh -N -L 1234:/var/run/mysqld/mysqld.sock demo #Un esempio di SSH che fa  
tunneling da INET socket ( internet ) a UNIX socket ( locali alla macchina )
```

Comparison Factor	Unix Sockets	TCP/IP Sockets
-------------------	--------------	----------------

Full Name	Unix sockets are also known as Inter-process Communication ( IPC )	TCP/IP stands for Transmission Control Protocol/Internet Protocol
-----------	--	---

Functionality	Unix sockets are used for communication between processes running on the same computer	TCP/IP sockets are used for programs running on different computers
---------------	--	---

Requirements	There is no need for a hostname and port number for processes to communicate	There is a need for a hostname and port number for programs
--------------	--	---

to communicate using TCP/IP sockets  
Speed Unix sockets connections are fast because the processes are running on the same system, and they can avoid some checks and operations TCP/IP connections are slow relative to Unix sockets

!!!!!!!!!!!!!!

Abbiamo visto che ssh consente di fare tunneling. Ci sono altre soluzioni, a volte più semplici, per il tunneling.

A che serve il tunneling:

La macchina A è esposta su una lan. La macchina B è collegata solo alla macchina A.

Se lei vuole fare in modo che un client che si collega alla macchina A possa accedere a un servizio di rete (TCP) che è esposto dalla macchina B, allora deve fare un tunneling tra A e B. Il tunneling nel tcipip va fatto per IP e per PORTA.

oltre a ssh ad esempio abbiamo l'app: socat

Esempio: apriamo due terminali: uno sul server, l'altro sul desktop

Sul desktop configuriamo il firewall per consentire solo al server 8134) di accedere ad apache sulla porta 80...

ufw deny 80 e 8080

ufw allow from <server> to any port 80 e 8080

ufw delete ... per cancellare le precedenti deny

ufw status numbered per visualizzare le regole numerate e per poterle poi cancellare

Esempio dalle slide Architetture di rete

1) attiviamo un firewall sulla macchina 134

ssh -N -L 192.168.214.134:80:192.168.214.133:80 user@192.168.214.133

Losvantaggio è che la 134 deve poter accedere a un account della 133

Con socat non ha necessità di fare accessi.

Da questo punto di vista socat è più simile al firewall.

Esempio 1) sudo socat TCP4-LISTEN:80 STDOUT

Si mette in ascolto e riporta sullo stdout

Esempio 2) sudo socat TCP4-LISTEN:80 TCP4:192.168.214.133:80

Si mette in ascolto e gira sulla 133

Va bene ma fa solo una comunicazione

Esempio 3) sudo socat TCP4-LISTEN:80,fork,reuseaddr TCP4:192.168.214.133:80  
Ora è ok!!!

La fork sotto linux significa: duplica il processo

### 3.87 mysql - mariadb

man mariadb

```
apt install mysql-server
```

Per installare il workbenh: andare sul sito e scaricare mysqlworkbenc.deb

```

per installare i .deb

sudo dpkg -i mysql-workbenh.....

mysql va in ascolto sulla porta tcp n. 3306
sudo mysql
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY "
root"; #La password dovrebbe essere un pochino più forte!!!

Fare partire mysql-workbench
Fare il login con root/root
Una volta fatto: create new schema
Poi, all'interno dello schema: create table
ecc.

sudo systemctl stop/start mysql

Inoltre può installare sqlite3
un db locale con tutte le qualità di un rdbms completo.
apt install sqlite3

```

### 3.88 Domande

- A cosa serve il comando Unix "mount"? montare volumi interni o esterni
- Che cos'è un programma filtro sulla Shell bash di Linux ??è un meccanismo legato alle pipe che trasferiscono lo stdout di un programma nello stdin di un altro programma
- Che differenza c'è tra un socket Unix e un socket Internet (*AF\_INET*)? Un socket Unix è solo interno alla macchina e si appoggia su un nome file nel filesystem. Un socket internet utilizza il protocollo IP e TCP per mettere in comunicazione due programmi sulla stessa macchina oppure su due macchine in rete.
- Con quale comando in Linux si configura la macchina "192.168.0.1" come router di default?? route add default gw 192.168.0.1
- Con quale comando Unix si possono vedere i permessi della directory "/tmp"??ls -ld /tmp
- Con quale comando Unix si possono visualizzare i messaggi del file "messages.log" mentre arrivano?? tail -f messages.log
- Con quale comando Unix si possono salvare tutti i files della directory corrente nell'archivio "arch1"?? tar cvf arch1 \*
- Con quali utilities Linux può condividere stampanti con sistemi Windows??Samba
- Cosa accade quando il sistema Unix va a 'run level' 6?? Compie un reboot
- Cosa compie, nella shell Bash di Linux, il comando "cd -"??Porta alla directory precedente all'ultimo comando "cd"
- Cosa esegue il comando Unix "vi -r rapporto"?? Recupera il file "rapporto" salvato dal sistema
- Cosa esegue la seguente linea di comando di una Shell Unix ? chown Tom FileName? ? Assegna la proprietà del file Filename a Tom
- Cosa esegue la seguente linea di comando di una Shell Unix? cat file1 file2 file3? ? Combina in concatenazione ordinata il file1, il file2 ed il file3 restituendo il risultato su stdout
- Cosa restituisce il comando "history" di una Shell Unix?? ? Gli ultimi comandi inseriti sulla Shell
- Cosa si intende per "distribuzione" Linux??Una collezione completa e installabile di software di sistema e utilities
- Cosa si ottiene se si lancia il seguente comando Unix? echo "3+4" ; bc? ? Un messaggio di errore
- Cosa si ottiene se si lancia il seguente comando Unix? echo "3+4" > bc ??

- Cosa visualizza la linea di comando di una shell linux "man ls — less"? La pagina di manuale del comando ls mostrando una pagina per volta
  - E' necessaria l'interfaccia grafica su un server di rete Linux?? No, anzi appesantisce il sistema
  - Il comando di una Shell Unix "chmod 764 filename":? ? Assegna ogni permesso al proprietario, di r/w al gruppo proprietario, solo di read a tutti gli altri
  - Il comando Unix "sort" può essere utilizzato per:? ? Ordinare le linee di un file di testo
  - Il file 'lista' contiene su tre linee le parole 'uno', 'due' e 'tre'. Cosa mostra il comando Unix "tac lista"? Tre due uno
  - Il Kernel di Linux supporta IPv6?? ? Si, da molti anni
  - In "vi" il comando Unix "j" minuscolo sposta il cursore in basso di una linea. Cosa fa il comando "J" maiuscolo?? Esegue il "join" della riga successiva a quella corrente
  - In quale linguaggio di programmazione è principalmente scritto uno Unix moderno?? ?C Il kernel di linux è tutto scritto, tranne pochissime parti, in linguaggio C.
  - In Unix, il file ".profile" è il file nascosto che contiene:? ? Comandi per la shell Bourne, e che viene eseguito automaticamente ad ogni login due file (.bashrc e .profile) sono eseguiti ad ogni login di linux/bash se fosse un'altro cli allora: se csh il file è .cshrc se ksh, allora .kshrc Usualmente .profile inizializza i path .bashrc ( o simili) inizializza la shell
  - In Unix, un alias è il comando per l'assegnazione di un diverso nome a:? Un comando di shell. per assegnare un alias, in .bashrc: alias newll='!bin/ls -laF' per vedere gli alias : alias
  - La pipeline Unix "ls -l — lpr" serve a:? ? Generare e stampare un elenco di Files Differenza tra ls -l — lpr ls -l — xargs lpr 1) nel primo ls -l costruisce un elenco di nomi di file. Questo elenco diventa lo stdin del comando successivo. lpr se non ha parametri legge lo stdin e lo mette in stampa. QUindi con questo comando stampo i nomi dei file presenti nella cartella corrente. 2) nel secondo comando, l'elenco dei nomi dei file è passato per xargs che li prende uno per uno e li fa diventare parametri per lpr che quindi stamperà i contenuti di ognuno dei file presenti nella cartella. Per chiarire ls torna =\_ a.out bc ex1.sh ex2-0.sh ex2.sh ex3.sh ex4.sh ex5.sh ex6.sh pippsock try.c Quindi la stringa "a.out bc ex1.sh ex2-0.sh ex2.sh ex3.sh ex4.sh ex5.sh ex6.sh pippsock try.c" diventa stdin di lpr che la stampa sulla stampante Con xargs, la stringa "a.out bc ex1.sh ex2-0.sh ex2.sh ex3.sh ex4.sh ex5.sh ex6.sh pippsock try.c" viene decomposta in nomi e quindi lpr farà: lpr a.out, lpr bc, lpr ex1.sh, .... e quindi stampa tutti i file
- Ancora un esempio ls — wc > numero di file presenti ls — xargs wc => mi torna il numero di caratteri in ogni file
- Altro esempio:
- ```

find . -type f -name "*.sh"
torna :
./ex2.sh
./ex5.sh
./ex2-0.sh
./ex4.sh
./ex1.sh
./ex6.sh
./ex3.sh
Se faccio: find . -type f -name "*.sh" | wc
wc gli arriva lo stdin contenente
" ./ex2.sh
./ex5.sh
./ex2-0.sh
./ex4.sh
./ex1.sh
./ex6.sh
./ex3.sh"
e quindi wc torna 7 linee
Invece quando uso xargs lo stdin viene dato a xargs
che lo prende riga per riga e lo fa diventare parametro di wc

```

```

find . -type f -name ".*.sh" | xargs wc
30 100 661 ./ex2.sh
23 140 772 ./ex5.sh
13 78 378 ./ex2-0.sh
22 112 671 ./ex4.sh
10 20 108 ./ex1.sh
10 64 375 ./ex6.sh
3 7 47 ./ex3.sh
111 521 3012 totale

```

In questo caso wc si comporta come il seguente programma  
`find . -type f -name ".*.sh" | while read riga; do wc $riga; done`  
`find . -type f -name ".*.sh" | while read riga; do echo $riga; done`

Completemente equivalente a: `find . -type f -name ".*.sh" | xargs wc`  
oppure a `find . -type f -name ".*.sh" | xargs echo`

Il vantaggio di linux è che tutti i file , dispositivi , comunicazioni

La Shell rappresenta:?? Il programma Unix interprete dei comandi dell 'utente  
Nel comando Unix "ls -lR / 2> lista", cosa indicano i caratteri "2>"?? La redirezione  
Nella maggioranza delle versioni Unix, con quale ambiente interagisce l'utente di una  
Nell 'ambito di sistemi operativi Linux/Unix, qual è la caratteristica principale del Kernel  
Nell 'ambito di sistemi operativi Unix/Linux, con quali componenti di sistema il Kernel  
Nell 'ambito di sistemi operativi Unix/Linux, con quale utility comunicano i processi tra loro  
Nell 'ambito di sistemi operativi Unix/Linux, cosa avviene tipicamente ai processi in background  
Nell 'ambito di sistemi operativi Unix/Linux, cosa identifica i poteri totali dell 'ammministratore  
Nell 'ambito di sistemi operativi Unix/Linux, cosa occorre fare dopo aver ricompilato i driver  
Nell 'ambito di sistemi operativi Unix/Linux, cos'è uno "inode"?? La struttura rappresentata  
Nell 'ambito di sistemi operativi Unix/Linux, il file ".cshrc":?? E' un file della home  
Nell 'ambito di sistemi operativi Unix/Linux, in quale directory sono raccolti i file dei programmi  
Nell 'ambito di sistemi operativi Unix/Linux, in quale file è configurato lo User ID del root  
Nell 'ambito di sistemi operativi Unix/Linux, in quale file si devono aggiungere i nomi dei gruppi  
Nell 'ambito di sistemi operativi Unix/Linux, per poter cancellare un file da una directory  
Nell 'ambito di sistemi operativi Unix/Linux, per rinominare l'utente esistente "mario"  
Nell 'ambito di sistemi operativi Unix/Linux, qual è il miglior metodo per cambiare la password  
Nell 'ambito di sistemi operativi Unix/Linux, qual è il nome del processo di sistema  
Nell 'ambito di sistemi operativi Unix/Linux, quale tra le seguenti è una corretta definizione  
Nell 'ambito di sistemi operativi Unix/Linux, quale variabile d'ambiente identifica un file  
Nell 'ambito di sistemi operativi Unix/Linux, quando si esegue un file con il bit SUID  
Nell 'ambito di sistemi operativi Unix/Linux, se si aggiungono pagine di manuale nella directory  
Nell 'ambito di sistemi operativi Unix/Linux, se si estraе un pacchetto software della repository  
Nell 'ambito di sistemi operativi Unix/Linux, se si verifica un errore fatale in un programma  
Nell 'ambito di sistemi operativi Unix/Linux, su un sistema con due schede di rete, quale interface  
Nell 'ambito di sistemi operativi Unix/Linux, un segmento di memoria condivisa appartiene al  
Per aggiungere 2000 nuovi utenti in modalità batch, tutti con la stessa password inizializzata  
Per attivare a mano il protocollo DHCP come client sulla scheda eth1, che comando Unix eseguire  
Per ispezionare la tabella di routing senza risolvere gli indirizzi col DNS, che comando  
Per massima portabilità ad altre versioni Unix, quale shell è meglio usare per scrivere script  
Per terminare in modo ottimale il processo 413, il comando Unix migliore è:?? Kill 413  
Per vedere l'occupazione totale sul disco della directory "/var/log", che comando Unix eseguire  
Perché è necessario mantere aggiornato il codice Linux?? Per porre rimedio alle vulnerabilità  
Qual è la dimensione massima di un Kernel Linux versione 2.4?? Non vi è alcuna dimensione  
Qual è l'utilizzo primario del comando "file" di una Shell Linux?? Indica se l'argomento  
Quale comando avrebbe l'effetto di visualizzare la lista dei login id riconosciuti dal sistema  
Quale è l'utilizzo primario del comando "grep" di una Shell Unix?? Trova le linee che  
Quale è l'utilizzo primario del comando "ln" di una shell Linux?? Crea un nuovo collegamento  
Quale è l'utilizzo primario del comando "mv" di una shell Linux?? Rinomina un file o  
Quale è l'utilizzo primario del comando "nice" di una Shell Bash Linux?? Esegue un comando

Quale è l'utilizzo primario del comando "ps" di una Shell Bash Linux?? ? Visualizza in  
Quale file di Linux contiene le password crittografate degli utenti?? ? /etc/shadow  
Scenario Linux: nella directory appropriata mancano sia il file "cron.allow" che il fi  
Scenario Unix: in un file system remoto, montato con NFS, perché root non può creare u  
Scenario Unix: l'utente "pippo" crea una directory coi permessi 777. Chi può rimuovere  
Scenario Unix: l'utente "pippo" crea una directory pubblica coi permessi 1777. Chi può  
Scenario Unix: un file system pieno di files molto piccoli, dà errore "Full" quando am  
Scenario Unix: una nuova libreria dinamica viene installata nella directory "/root/pro  
Se si da il comando Unix "echo \$\$", cosa si ottiene?? ? L'identificativo di processo di  
Si può usare liberamente del codice sorgente Unix/Linux con licenza GNU?? ? Si, ma tut  
Tutto il software che gira su Linux è gratis??No, può anche essere a pagamento e/o con  
Uno stesso utente Unix può compiere login da più terminali simultaneamente?? ? Si

### 3.89 xdg-open

```
xdg-open <nome file>
in base al tipo di file attiva il programma che, di default, lo manipola
```

### 3.90 Montaggio dischi

```
per montare un disco ntfs
sudo mount -t ntfs -o rw /dev/sdcNNN /mnt (il posto dove montarlo)
per smontare
sudo umount /mnt
```

```
Per vedere tutti i dischi collegati alla mia macchina
sudo fdisk -l
```

```
sudo parted -l
sudo parted -l | grep Disco (Disk in inglese)
```

### 3.91 rm

```
man rm
```

alle 16.10