

# **Academic Report**

**LAS**

Laboratorio di Amministrazione di Sistema

Mattia Scarpa, Claudio Giacomello, Luca Palmetti

Fabrizio Romano

8 May 2024

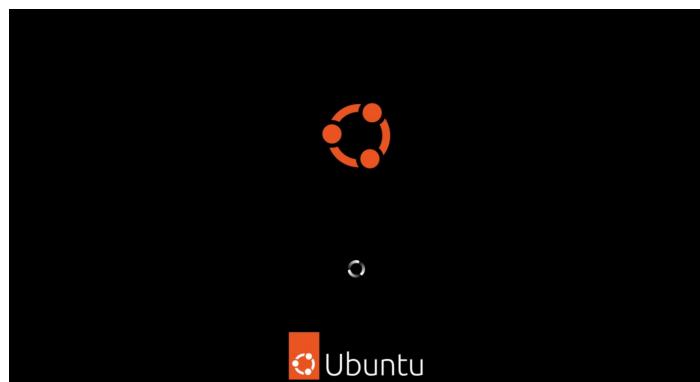
# Introduzione

Ad ogni gruppo è stato assegnato un portatile hp g6, una chiavetta USB con l'installazione di Ubuntu e una tastiera USB.

I laptop sono etichettati come hp-xx (con xx = 12 nel nostro caso) e questo sarà il nome del nostro server. L'indirizzo IP sarà 192.168.100.yy (con yy = 62 nel nostro caso). **[192.168.100.62]**.

## Esercitazione 1 : Installazione di Ubuntu Server

Accendiamo il computer e inseriamo la chiavetta. Ci apparirà la seguente schermata iniziale di Ubuntu :

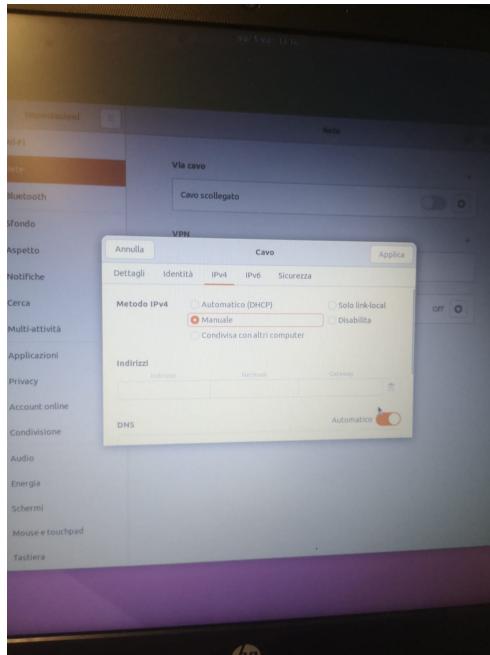
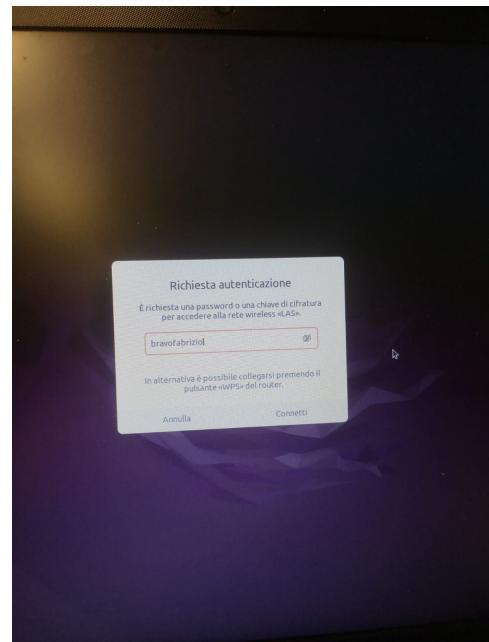
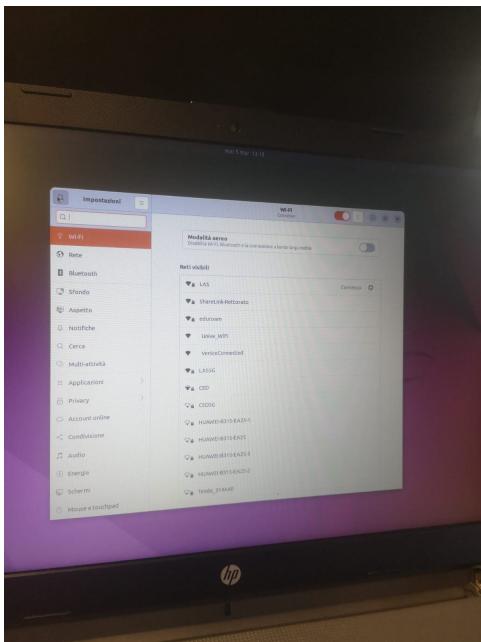


Dopodiché verrà chiesto di scegliere la lingua (italiano nel nostro caso) e scegliere anche se fare "Prova Ubuntu" o "Installa Ubuntu". Confermiamo la prima opzione :



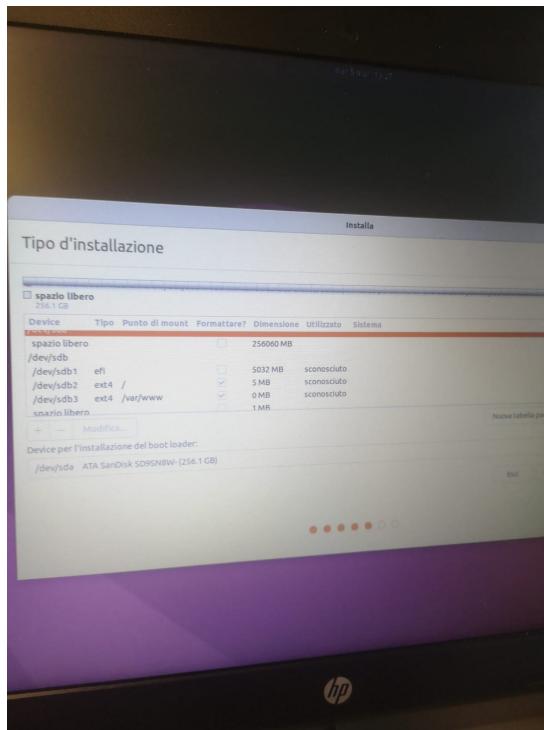
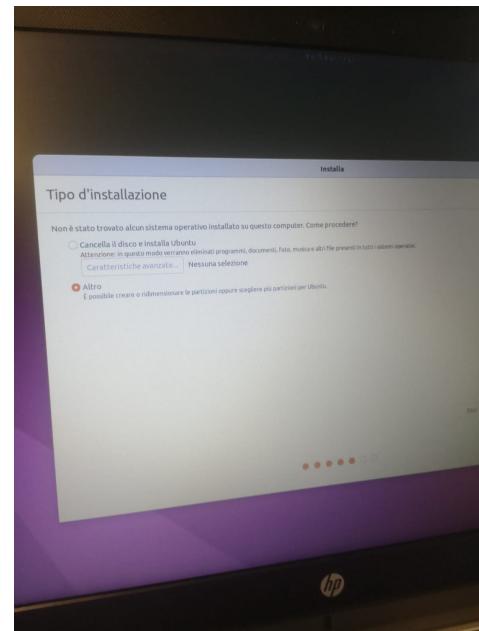
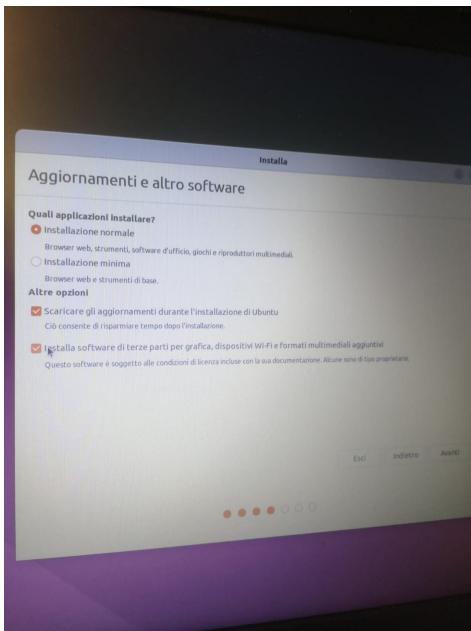
Dopo aver schiacciato “Prova Ubuntu”, sul desktop apparirà l’icona del pacchetto di installazione di Ubuntu da utilizzare più tardi.

Il prossimo passo sarà quello di cercare la rete wifi LAS (oppure LAS5G) con password “bravofabrizio” e, dopo aver inserito in rete il laptop, dobbiamo modificare le impostazioni della scheda di rete ed inserire la configurazione manuale :



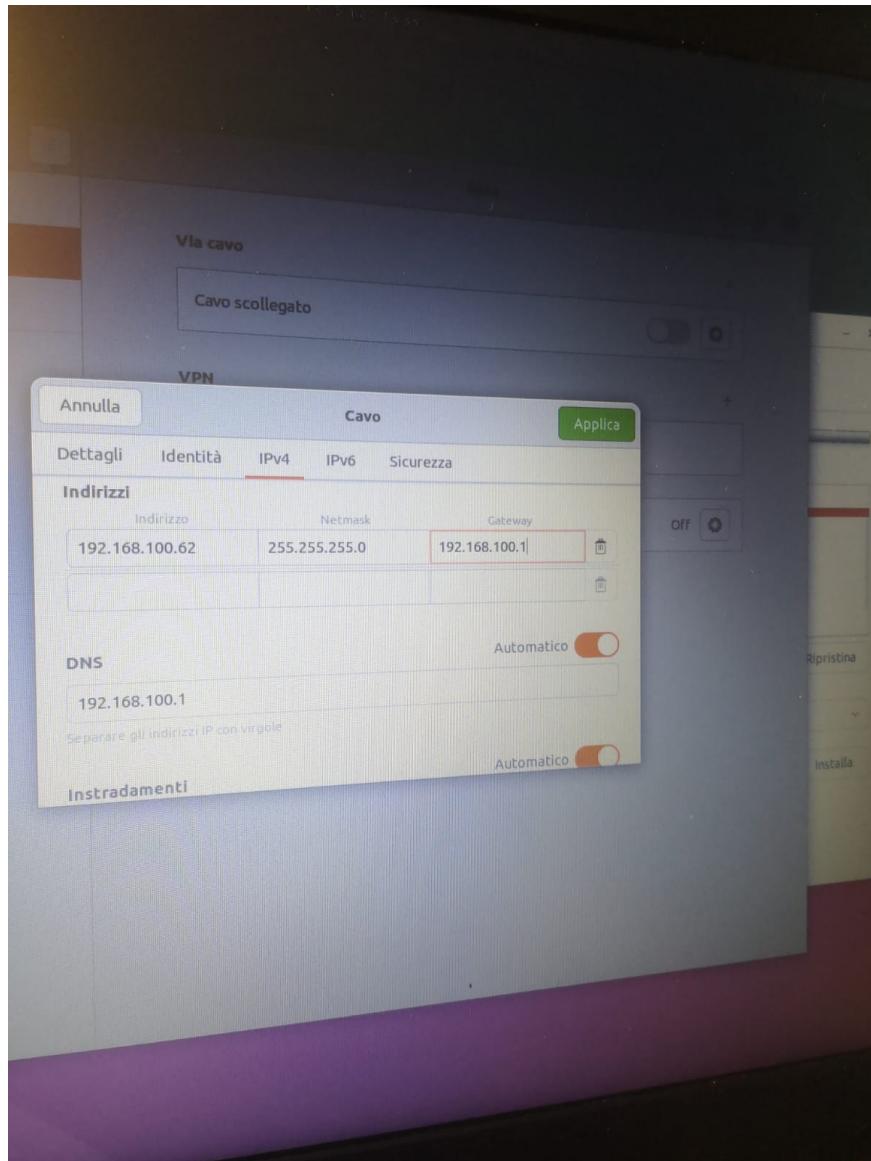
Ora possiamo far eseguire il pacchetto ed iniziare la vera installazione.

La prima facciata che compare è “Aggiornamenti e altro software” dove dobbiamo selezionare alcune opzioni. Poi verrà chiesto il tipo di installazione che vogliamo attuare e dobbiamo scegliere “Altro”. Proseguendo con “il tipo di installazione” verrà richiesto di modificare la tabella delle partizioni.

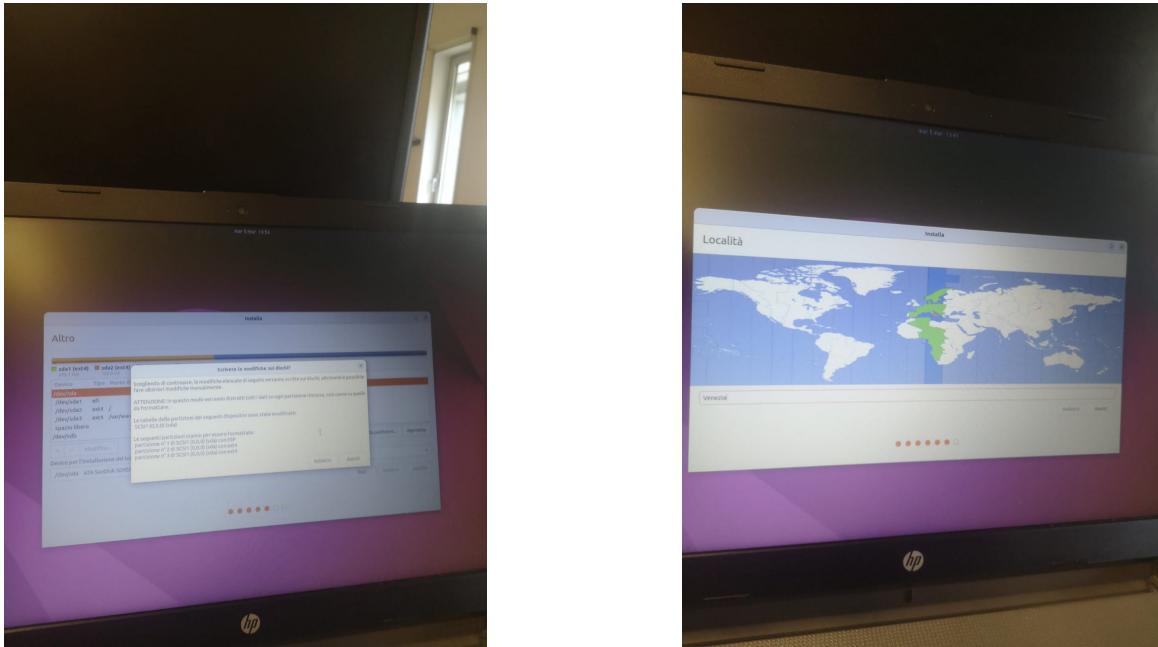


Arrivati a questo punto, dobbiamo tornare sulle impostazioni di rete e andare sulla sezione IPv4 (IP versione 4), dove dovremo selezionare il metodo “Manuale” e modificare gli “Indirizzi” (che comprendono Indirizzo, Netmask e Gateway) ed il DNS (Domain Name System).

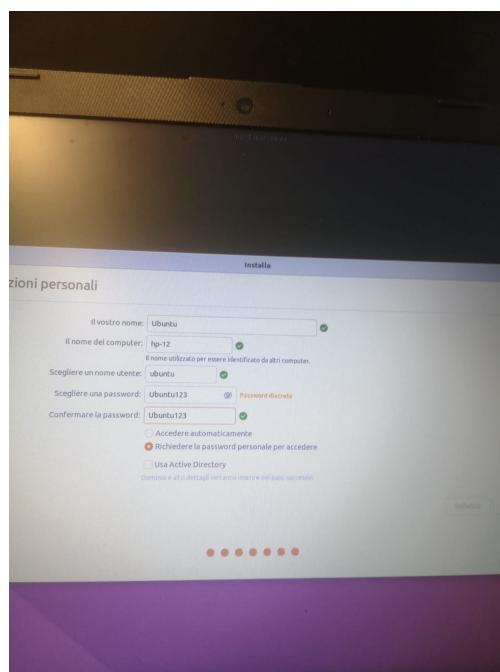
Rispettivamente, dobbiamo inserire : Indirizzo = 192.168.100.62, Netmask = 255.255.255.0 (Classe C in particolare), Gateway = 192.168.100.1 ed infine il DNS = 192.168.100.1 .



Tornando al tipo di installazione, dopo aver cliccato su “Installa” comparirà sullo schermo un messaggio che fa un recap della situazione sulle partizioni e chiede di confermare. Confermate le modifiche, scegliamo la nostra località attuale (Italia) :

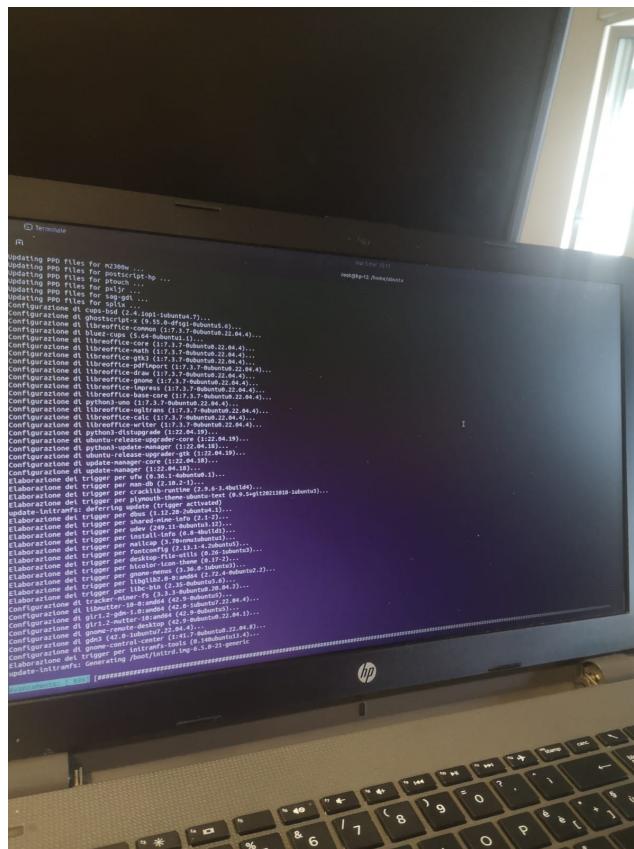
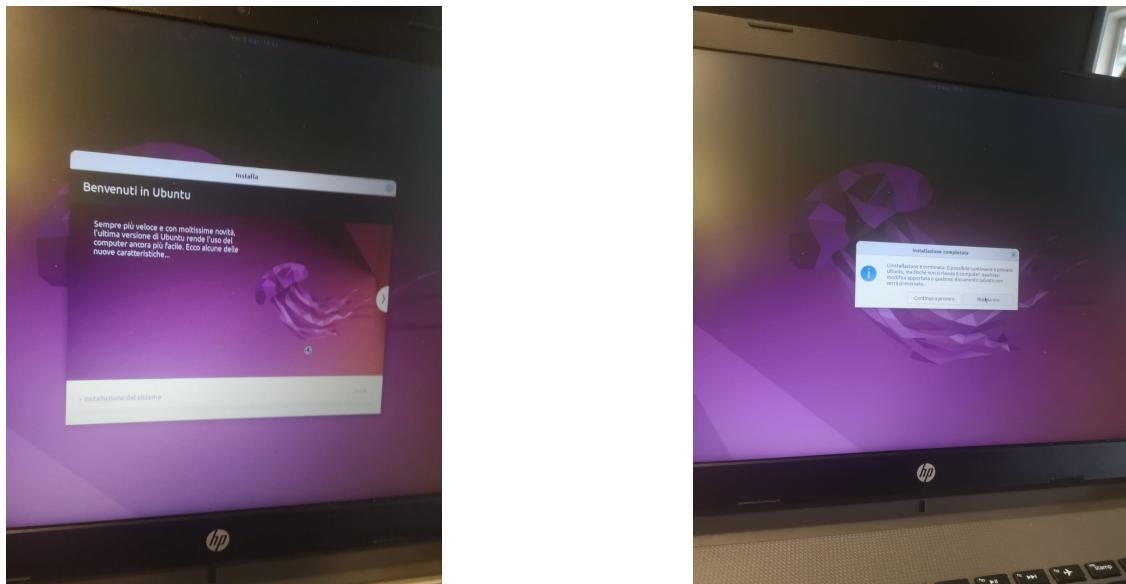


Infine, appare una finestra in cui viene richiesto di inserire le informazioni personali come nome gruppo password ecc...



L'installazione è completata. Appare la schermata di Benvenuto e inizia a scaricare tutti i file necessari. Basta riavviare il pc e Ubuntu si aprirà.

L'ultimo passo è quello di installare **ssh** attraverso il terminale con il comando : **sudo apt install ssh**.



## Esercitazione 2 : Comandi e SHELL

In questa esercitazione sarà richiesto di testare vari comandi Linux :

### Esercizi :

#### Hello World

Direttamente dal prompt dei comandi : # echo "Hello World!"

```
ubuntu@ubuntu-hp12:~$ echo "Hello World"  
"Hello World"  
ubuntu@ubuntu-hp12:~$
```

#### Creare uno script

Utilizzare un editor di testo per creare un file contenente il nostro script :

```
# nano mio_script.sh
```

Inserire il contenuto dello script :

```
#!/bin/bash
```

```
echo "Hello world!"
```

The screenshot shows a terminal window with the title 'UW PICO 5.09'. The file being edited is 'mioscript.sh'. The content of the file is:

```
#!/bin/bash  
echo "Hello world!"
```

At the bottom of the screen, there is a menu bar with various keyboard shortcuts for text editing.

```
ubuntu@ubuntu-hp12:~$ nano script.sh  
ubuntu@ubuntu-hp12:~$ bash script.sh  
Hello world!  
ubuntu@ubuntu-hp12:~$
```

Per eseguire lo script c'è anche un modo alternativo più macchinoso : invece di usare bash nome\_file.sh, possiamo dettare i permessi del file con "**chmod a+x nome\_file.sh**" e poi eseguirlo tramite "**./nome\_file.sh**".

## Stampare messaggi a video

Per stampare messaggi a video utilizziamo il comando echo (salviamo il file come echo.sh) :

The screenshot shows a terminal window with two panes. The left pane displays the code for echo.sh:

```
UW PICO 5.09          File: echo.sh          Modified
echo "Testo con newline"
echo -n "Testo senza newline"
echo -e "\n Testo \t senza \'t backslash \n"■
```

The right pane shows the output of running the script:

```
ubuntu@ubuntu-hp12:~$ nano echo.sh
ubuntu@ubuntu-hp12:~$ bash echo.sh
Testo con newline
Testo senza newline
Testo senza backslash
ubuntu@ubuntu-hp12:~$ ■
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts.

## Uso dei commenti e espressioni algebriche

Il seguente script, commenti.sh, mostra l'uso dei commenti ed effettua la somma di due numeri qualsiasi :

The screenshot shows a terminal window with two panes. The left pane displays the code for commenti.sh:

```
UW PICO 5.09          File: commenti.sh          Modified
# Somma due numeri a piacere
# ((sum=25+35))
# Stampa il risultato
echo $sum■
```

The right pane shows the output of running the script:

```
ubuntu@ubuntu-hp12:~$ nano commenti.sh
ubuntu@ubuntu-hp12:~$ bash commenti.sh
60
ubuntu@ubuntu-hp12:~$ ■
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts.

The screenshot shows a terminal window with two panes. The left pane displays the code for commenti.sh:

```
UW PICO 5.09          File: commenti.sh          Modified
# Somma due numeri a piacere
# ((sum=25+35))
# Stampa il risultato
echo $sum■
```

The right pane shows the output of running the script:

```
ubuntu@ubuntu-hp12:~$ nano commenti.sh
ubuntu@ubuntu-hp12:~$ bash commenti.sh
60
ubuntu@ubuntu-hp12:~$ ■
```

# Ciclo WHILE

Incrementiamo una variabile partendo da 1 fino a 5 (salviamo il file come while.sh) :

The screenshot shows two terminal windows side-by-side. The left window is a text editor titled 'File: while.sh' with the following content:

```
UW PICO 5.09
File: while.sh
Modified
valid=true
count=1
while [ $valid ]; do
    echo $count
    if [ $count -eq 5 ]; then
        break
    fi
    ((count++));
done
```

The right window shows a terminal session with the command `ubuntu@ubuntu-hp12:~$ nano while.sh`, followed by the numbers 1, 2, 3, 4, and 5, indicating the output of the script.

# Ciclo FOR

Stampiamo il conto alla rovescia per il lancio di un missile (salviamo il file come for.sh) :

The screenshot shows two terminal windows side-by-side. The left window is a text editor titled 'File: for.sh' with the following content:

```
UW PICO 5.09
File: for.sh
Modified
for (( counter=10; counter>0; counter-- )); do
    echo -n "$counter"
done
echo ""
```

The right window shows a terminal session with the command `ubuntu@ubuntu-hp12:~$ nano for.sh`, followed by the numbers 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, indicating the output of the script.

The screenshot shows two terminal windows side-by-side. The left window is a text editor titled 'File: for.sh' with the following content:

```
UW PICO 5.09
File: for.sh
Modified
for (( counter=10; counter>0; counter-- )); do
    echo -n "$counter"
done
echo ""
```

The right window shows a terminal session with the command `ubuntu@ubuntu-hp12:~$ bash for.sh`, followed by the numbers 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, indicating the output of the script.

# Costrutto IF

Un blocco IF inizia con la parola chiave **if** e termina con la parola chiave

**fi.** La clausola è racchiusa fra parentesi quadre (salviamo il file come if.sh).

Le condizioni possono essere anche il risultato di espressioni logiche con **and** e **or**.

La clausola **else if (elif)** permette di annidare più costrutti if in un unico blocco.

# if with &&

```
UW PICO 5.09          File: if.sh          Modified
echo "Login:"
read username
echo "Password:"
read password

if [ ( $username == "admin" && $password == "secret" ) ]; then
    echo "Valid user"
else
    echo "invalid user"
fi

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg  ^U UnCut Text  ^T To Spell
```

```
ubuntu@ubuntu-hp12:~$ nano if.sh
ubuntu@ubuntu-hp12:~$ bash if.sh
Login:
admin
Password:
secret
valid user
ubuntu@ubuntu-hp12:~$ bash if.sh
Login:
admin1
Password:
secret1
invalid user
ubuntu@ubuntu-hp12:~$
```

# elif

```
UW PICO 5.09          File: if.sh          Modified
echo "Enter your lucky number"
read n
if [ $n -eq 101 ]; then
    echo "You got 1st prize"
elif [ $n -eq 510 ]; then
    echo "You got 2nd prize"
elif [ $n -eq 999 ]; then
    echo "You got 3rd prize"
else
    echo "Sorry, try for the next time"
fi

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg  ^U UnCut Text  ^T To Spell
```

```
ubuntu@ubuntu-hp12:~$ nano if.sh
ubuntu@ubuntu-hp12:~$ bash if.sh
Enter your lucky number
101
You got 1st prize
ubuntu@ubuntu-hp12:~$ bash if.sh
Enter your lucky number
999
You got 3rd prize
ubuntu@ubuntu-hp12:~$ bash if.sh
Enter your lucky number
100
Sorry, try for the next time
ubuntu@ubuntu-hp12:~$
```

## Costrutto CASE

In alternativa a else if è possibile usare il costrutto **case**, il blocco inizia con la clausola **case** e si chiude con la clausola **esac**, i casi sono evidenziati da parentesi tonda (salviamo il file come case.sh) :

```
UW PICO 5.09          File: case.sh          Modified
echo "Enter your lucky number:"
read n
case $n in
    101)echo "You got 1st prize" ;;
    510)echo "You got 2nd prize" ;;
    999)echo "You got 3rd prize" ;;
    *)echo "Sorry, try for the next time" ;;
esac

ubuntu@ubuntu-hp12:~$ nano case.sh
ubuntu@ubuntu-hp12:~$ bash case.sh
Enter your lucky number:
101
You got 1st prize
ubuntu@ubuntu-hp12:~$ bash case.sh
Enter your lucky number:
999
You got 3rd prize
ubuntu@ubuntu-hp12:~$ bash case.sh
Enter your lucky number:
100
Sorry, try for the next time
ubuntu@ubuntu-hp12:~$ 
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

## Funzioni

La Bash non permette la creazione di funzioni con parametri, ma è possibile utilizzare dei parametri sfruttando altre variabili (salviamo il file come func.sh):

```
UW PICO 5.09          File: func.sh          Modified
Area_Rettangolo() {
    area=$(( $1 * $2 ))
    echo "Area is : $area"
}
# $1 e $2 sono gli input che vengono dati alla funzione :
Area_Rettangolo 10 20

ubuntu@ubuntu-hp12:~$ nano func.sh
ubuntu@ubuntu-hp12:~$ bash func.sh
Area is : 200
ubuntu@ubuntu-hp12:~$ 
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

```
ubuntu@ubuntu-hp12:~$ nano func.sh
ubuntu@ubuntu-hp12:~$ bash func.sh
Area is : 200
ubuntu@ubuntu-hp12:~$ 
```

# Esistenza di file/directory

Tramite il costrutto if e degli switch particolari è possibile testare

l'esistenza di file o directory :

```
| UW PICO 5.09          File: file.sh          Modified
filename=$1
if [ -f "$filename" ]; then
    echo "File exists"
else
    echo "File does not exist"
fi
dirname=$2
if [ -d "$dirname" ]; then
    echo "Dir exists"
else
    echo "Dir does not exist"
fi

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg  ^U UnCut Text^T To Spell
```

```
ubuntu@ubuntu-hp12:~$ nano file.sh
ubuntu@ubuntu-hp12:~$ bash file.sh
File does not exist
Dir does not exist
ubuntu@ubuntu-hp12:~$ bash file.sh func.sh Shell
File exists
Dir exists
ubuntu@ubuntu-hp12:~$ bash file.sh case.sh ShellNot
File exists
Dir does not exist
```

## Altri esempi

Creare uno script che aggiorni data e ora ogni secondo :

```
| UW PICO 5.09          File: date.sh          Modified
while [ 1 = 1 ];
do
    date
    sleep 1
done

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg  ^U UnCut Text^T To Spell
```

```
ubuntu@ubuntu-hp12:~$ nano date.sh
ubuntu@ubuntu-hp12:~$ bash date.sh
Sun May 12 12:16:16 CEST 2024
Sun May 12 12:16:17 CEST 2024
Sun May 12 12:16:18 CEST 2024
Sun May 12 12:16:19 CEST 2024
Sun May 12 12:16:20 CEST 2024
Sun May 12 12:16:21 CEST 2024
Sun May 12 12:16:22 CEST 2024
Sun May 12 12:16:23 CEST 2024
Sun May 12 12:16:24 CEST 2024
Sun May 12 12:16:25 CEST 2024
Sun May 12 12:16:26 CEST 2024
Sun May 12 12:16:27 CEST 2024
```

Realizzare uno script che stampa il contenuto di una directory con il peso di ogni file :

```
| UW PICO 5.09          File: peso.sh           Modified
for i in $(ls -l); do
    du -hs $i;
done;■

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is   ^V Next Pg  ^U UnCut Text^T To Spell
```

```
ubuntu@ubuntu-hp12:~$ bash peso.sh
208K  Documenti/
2.4M  Java-C++
51M   Programmazione/
440K  ReTe/
84K   Shell/
348K  Test/
512K  Threads/
59M   UNI/
10M   WebStormFolder/
ubuntu@ubuntu-hp12:~$ █
```

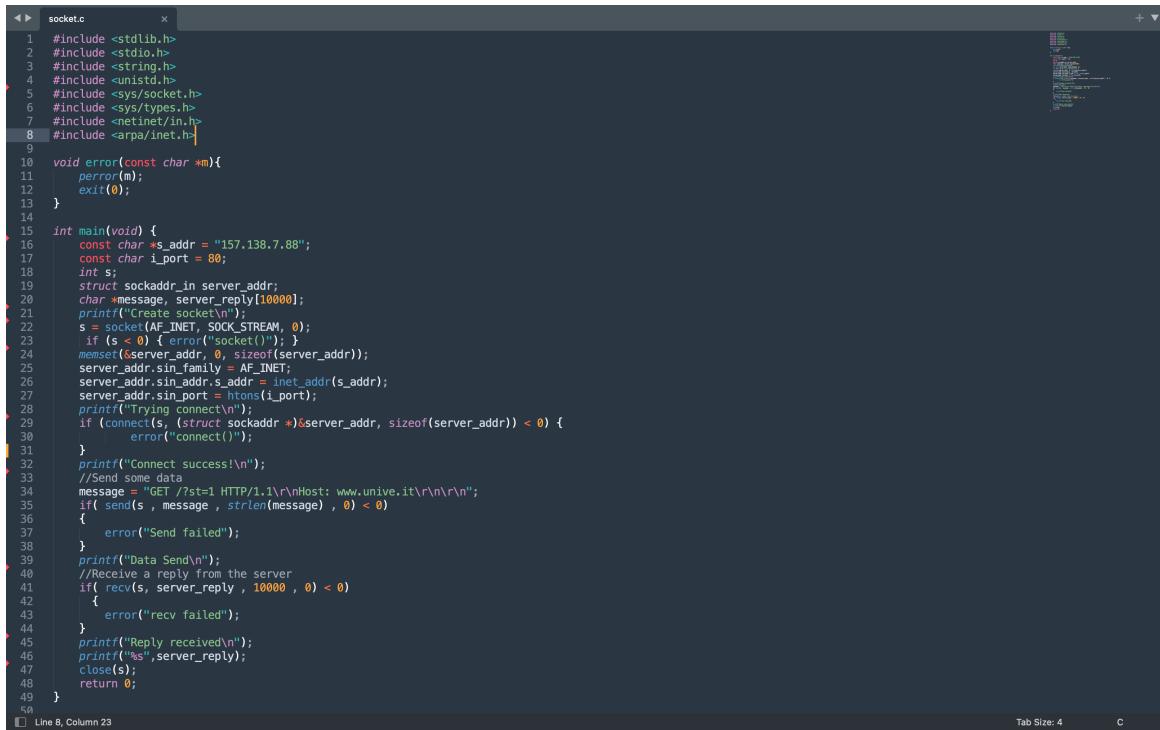
## Idea di progetto

Utilizzando i cicli, le funzioni, il costrutto case e i comandi di sistema Linux, realizzare uno script in bash che permette una semplice gestione di un server linux :

- Gestione degli utenti
- Gestione dei demoni
- Gestione del file system
- Gestione della rete
- Ecc...

# Esercitazione 3 : Servizi di Rete

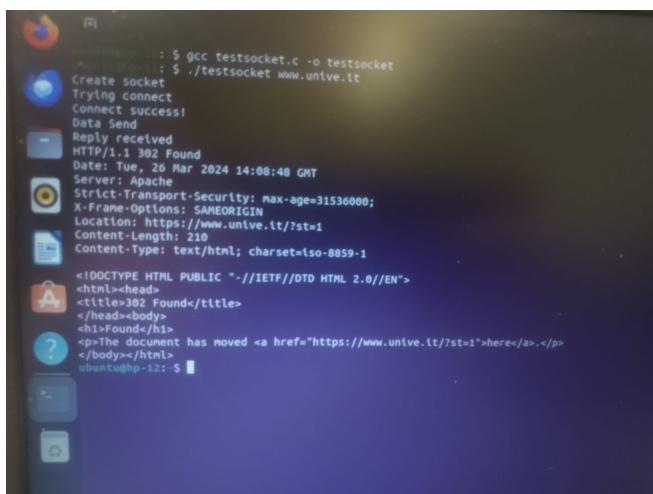
Scarichiamo il seguente file **testsocket.c** (vedi sotto),



```
socket.c  x
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <unistd.h>
5 #include <sys/socket.h>
6 #include <sys/types.h>
7 #include <netinet/in.h>
8 #include <arpa/inet.h>
9
10 void error(const char *m){
11     perror(m);
12     exit(0);
13 }
14
15 int main(void) {
16     const char *s_addr = "157.138.7.88";
17     const char i_port = 80;
18     int s;
19     struct sockaddr_in server_addr;
20     char message, server_reply[1000];
21     printf("Create socket\n");
22     s = socket(AF_INET, SOCK_STREAM, 0);
23     if (s < 0) { error("socket()"); }
24     memset(&server_addr, 0, sizeof(server_addr));
25     server_addr.sin_family = AF_INET;
26     server_addr.sin_addr.s_addr = inet_addr(s_addr);
27     server_addr.sin_port = htons(i_port);
28     printf("Trying connect\n");
29     if (connect(s, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
30         error("connect()");
31     }
32     printf("Connect success!\n");
33     //Send some data
34     message = "GET /?st=1 HTTP/1.1\r\nHost: www.unive.it\r\n\r\n";
35     if (send(s , message , strlen(message) , 0) < 0)
36     {
37         error("Send failed");
38     }
39     printf("Data Send\n");
40     //Receive a reply from the server
41     if (recv(s, server_reply , 10000 , 0) < 0)
42     {
43         error("recv failed");
44     }
45     printf("Reply received\n");
46     printf("%s",server_reply);
47     close(s);
48     return 0;
49 }
```

Lo compiliamo con “**gcc testsocket.c -o testsocket**” e lo eseguiamo con “**./testsocket**” ([www.unive.it](http://www.unive.it)).

Abbiamo anche bisogno del pacchetto *build-essential*, quindi lo scarichiamo con il comando **sudo apt install build-essential**.



```
: $ gcc testsocket.c -o testsocket
: $ ./testsocket www.unive.it
Create socket
Trying connect
Connect success!
Data send
Reply received
HTTP/1.1 302 Found
Date: Tue, 26 Mar 2024 14:08:48 GMT
Server: Apache
Strict-Transport-Security: max-age=31536000;
X-Frame-Options: SAMEORIGIN
Location: https://www.unive.it/?st=1
Content-Length: 210
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="https://www.unive.it/?st=1">here</a>.</p>
</body></html>
ubuntu@hp-112: $
```

Compilazione + esecuzione  
testsocket.c

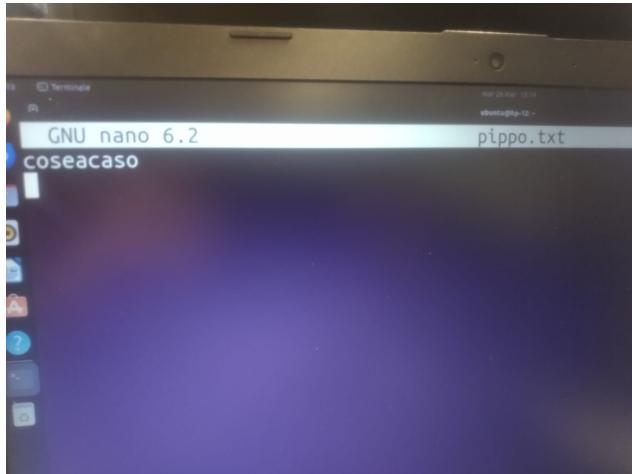
## Operazioni periodiche (at)

Vogliamo provare ad usare il comando at, quindi come prima cosa  
dobbiamo installarlo con “**`sudo apt install at`**”, poi dopo che è finita  
l’installazione lo testiamo attraverso il comando “**`at now`**”. Infine eseguiamo i  
seguenti comandi per creare un file di testo per poi inserirci dentro una  
parola :

- 1) touch /home/ubuntu/pippo.txt
  - 2) echo "coseacaso" > /home/ubuntu/pippo.txt
  - 3) premere CTRL+D per uscire

```
Attività Terminali
```

ubuntu@hp-12: \$ at now  
warning: commands will be executed using /bin/sh  
at Tue Mar 26 15:13:00 2024  
at> touch /home/ubuntu/pippo.txt  
at> echo "seocasao" > /home/ubuntu/pippo.txt  
at> PREMERE CTRL+D PER USCIRE<EOF>  
job 2 at Tue Mar 26 15:13:00 2024,  
ubuntu@hp-12: \$ ls  
LS: comando non trovato  
ubuntu@hp-12: \$ ls  
backup.sh ciao.sh elif.sh funcquadrat.sh  
Scartati script.sh testsocket  
ubuntu@hp-12: \$ .sh  
canc.sh commenti.sh file.sh func.sh  
script2.sh Scriventi testsocket.c hw.sh  
case2.sh download for.sh user\_id  
script3.sh sleep.sh test.txt utenti\_id  
case.sh echo.sh funcquadrat.sh iflog\_in\_number.sh  
script4.sh nano.pippo.txt  
\*\*\*. \$ nano pippo.txt

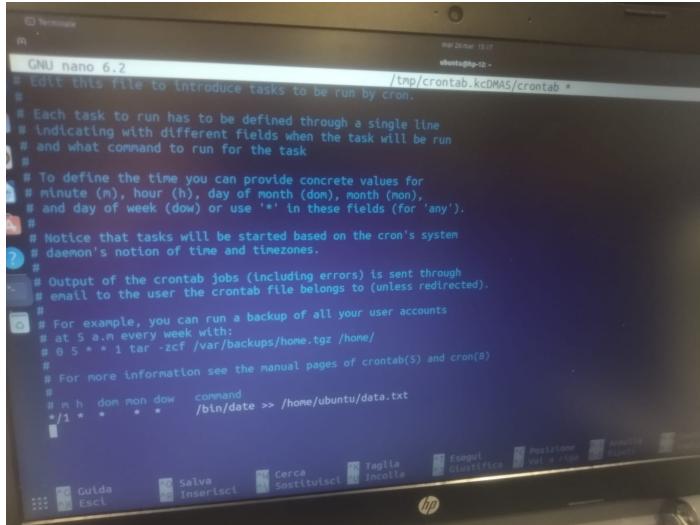


## Creazione file pippo.txt

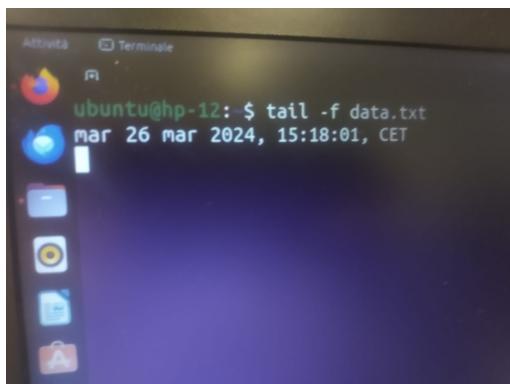
# Operazioni periodiche (crontab)

Editiamo la crontab dell'utente : “**crontab -e**”. Inseriamo la seguente riga in fondo al file : “**\*/1 \* \* \* \* /bin/date >> /home/ubuntu/data.txt**”.

Poi per controllare che tutto funzioni, eseguiamo : “**tail -f data.txt**” :



```
GNU nano 6.2
# Edit this file to introduce tasks to be run by cron.
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
# m h dom mon dow   command
*/1 * * * * /bin/date >> /home/ubuntu/data.txt
```



```
ubuntu@hp-12: ~ tail -f data.txt
mar 26 mar 2024, 15:18:01, CET
```

Legenda per comprendere meglio la riga inserita in fondo al file :

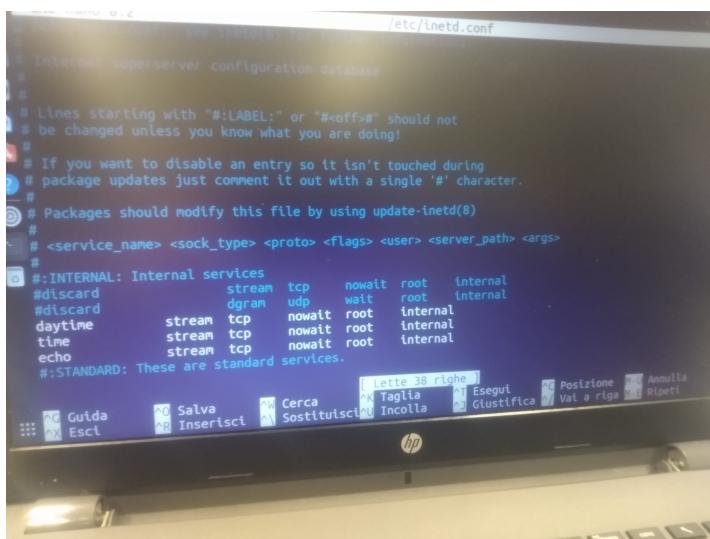
<b>* * * * *</b>	command(s)
<b>^ ^ ^ ^ ^</b>	
	allowed values
	-----
-----	Day of week (0 - 7) (Sunday=0 or 7)
-----	Month (1 - 12)
-----	Day of month (1 - 31)
-----	Hour (0 - 23)
-----	Minute (0 - 59)

## Servizi di base - installazione inetd

NB : per eseguire tutti i seguenti comandi dobbiamo essere root, quindi invece di scrivere “**sudo**” prima di ogni comando, possiamo “settarcì” come root con il comando “**sudo -s**”.

Tra i servizi di base possiamo elencare ftp, tftp, telnet, rsh.

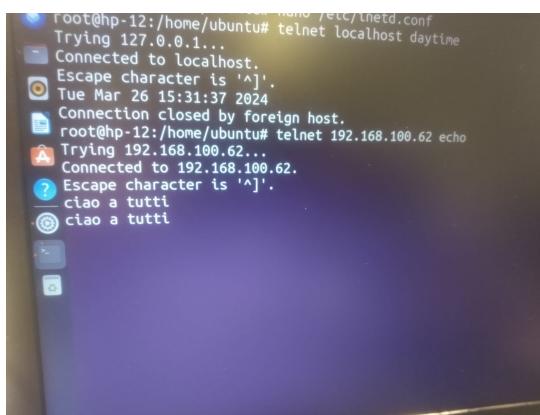
In Debian / Ubuntu per installare il pacchetto basta lanciare il comando : “**apt install openbsd-inetd**”. Il pacchetto è poi configurabile editando il file “**/etc/inetd.conf**”.



```
# Internet superserver configuration database

# Lines starting with "#:LABEL:" or "#<off>" should not
# be changed unless you know what you are doing!
#
# If you want to disable an entry so it isn't touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8)
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard    stream  tcp    nowait  root    internal
#discard    dgram   udp    wailt   root    internal
daytime    stream  tcp    nowait  root    internal
time      stream  tcp    nowait  root    internal
echo      stream  tcp    nowait  root    internal
#:STANDARD: These are standard services.
```

E proviamo ad attivare i servizi daytime, time ed echo (come nella figura sopra). Dopodiché li testiamo con i comandi : “service inetd restart”, “telnet localhost daytime” e “telnet 192.168.100.62”.



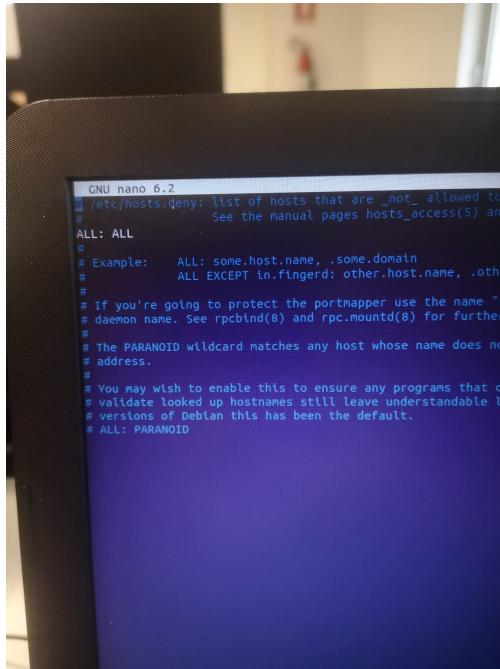
```
root@hp-12:/home/ubuntu# telnet localhost daytime
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^].
Tue Mar 26 15:31:37 2024
Connection closed by foreign host.
root@hp-12:/home/ubuntu# telnet 192.168.100.62 echo
Trying 192.168.100.62...
Connected to 192.168.100.62.
Escape character is '^].
? ciao a tutti
ciao a tutti
? ciao a tutti
```

Installiamo ora anche il servizio di trasferimento file *ftp* con il comando :

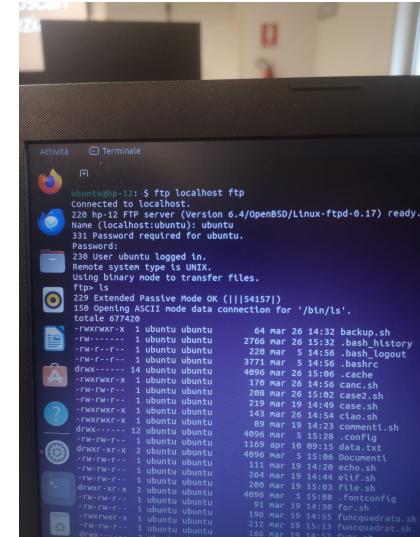
**“apt install ftptd”**, dopodiché li testiamo con i seguenti comandi: “**service inetd restart**”, “**ftp localhost ftp**” e “**ftp 192.168.100.62 ftp**”.

Dopodiché proviamo ad editare il file

**“hosts.deny”** nel seguente modo :



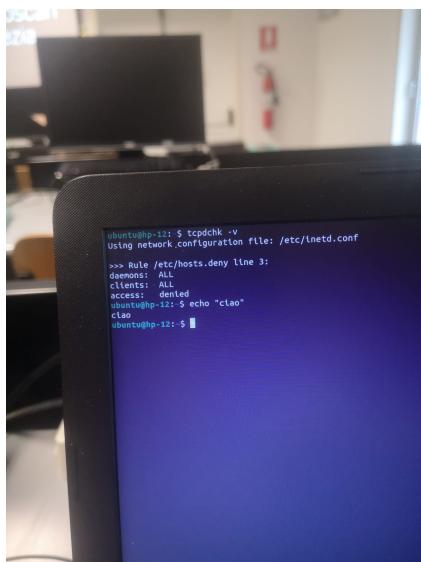
```
GNU nano 6.2
/etc/hosts.deny: list of hosts that are _not_ allowed to
                   See the manual pages hosts_access(5) and
ALL: ALL
#
# Example:    ALL: some.host.name, .some.domain
#             ALL EXCEPT in.fingerd: other.host.name, .other.
#
# If you're going to protect the portmapper use the name "r
# daemon name. See rpcbind(8) and rpc.mountd(8) for further
#
# The PARANOID wildcard matches any host whose name does not
# address.
#
# You may wish to enable this to ensure any programs that do
# validate looked up hostnames still leave understandable log
# versions of Debian this has been the default.
#
# ALL: PARANOID
```



```
ubuntu@hp-12: $ ftp localhost ftp
Connected to localhost.
220 hp-12 FTP server (Version 6.4/OpenBSD/Linux-ftp-0.17) ready.
Name (localhost:ubuntu): ubuntu
331 Password required for ubuntu.
Password:
230 User ubuntu logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Extended Passive Mode OK (|||541571)
150 Opening ASCII mode data connection for '/bin/ls'.
Total: 6770
drwxr-x--x 1 ubuntu ubuntu      64 mar 26 14:32 backup.sh
-rw-r--r-- 1 ubuntu ubuntu   2766 mar 26 15:32 .bash_history
-rw-r--r-- 1 ubuntu ubuntu   220 mar  5 14:56 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu   373 mar 26 15:32 .profile
drwxr--r-- 14 ubuntu ubuntu   4096 mar 26 15:06 .cache
-rw-r--r-- 1 ubuntu ubuntu   170 mar 26 14:56 .canc.sh
-rw-r--r-- 1 ubuntu ubuntu   209 mar 26 15:02 .case2.sh
-rw-r--r-- 1 ubuntu ubuntu   231 mar 26 15:02 .ciao.sh
-rw-r--r-- 1 ubuntu ubuntu   143 mar 26 14:54 .commenti.sh
drwxr-xr-x 12 ubuntu ubuntu   4096 mar 26 15:06 .config
drwxr-xr-x 2 ubuntu ubuntu   4096 mar  5 15:00 .documenti
drwxr-xr-x 1 ubuntu ubuntu   4096 mar 19 14:20 .echo.sh
drwxr-xr-x 1 ubuntu ubuntu   4096 mar 19 15:03 .filesh
drwxr-xr-x 1 ubuntu ubuntu   4096 mar  5 15:08 .fontconfig
drwxr-xr-x 1 ubuntu ubuntu   4096 mar 19 14:38 .for.sh
drwxr-xr-x 1 ubuntu ubuntu   196 mar 19 14:20 .funzionato.sh
drwxr-xr-x 1 ubuntu ubuntu   212 mar 19 15:11 .funzionato2.sh
drwxr-xr-x 1 ubuntu ubuntu   362 mar 19 14:53 .funzionato3.sh
```

Attenzione! Bisogna de-commentare

**“ALL: ALL”**, altrimenti nessuno avrà la possibilità di accedere (è la lista degli host che non hanno il permesso di accesso, quindi lasciando ALL stiamo dicendo che nessuno ha l'accesso).



```
ubuntu@hp-12: $ tcpdchk -v
Using network configuration file: /etc/lnetd.conf
>>> Rule /etc/hosts.deny line 3:
daemons: ALL
ctao: denied
ubuntu@hp-12: $ echo "ctao"
ctao
ubuntu@hp-12: $
```

==> Piccolo test : “**tcpdchk -v**”

# NTP - installazione e configurazione

In debian/ubuntu esiste il pacchetto ntp : “**apt install ntp**”, “**apt install ntpdate**” e “**apt install ntpstat**”.

Per la configurazione è sufficiente editare il file “**/etc/ntp.conf**” inserendo nella direttiva server il server ntp di riferimento :

“**server ntp.dsi.unive.it**” ==>

Altri comandi per la gestione del clock :

**rdate** : permette la sincronizzazione con un server che abbia a disposizione il servizio time :

“**apt install rdate**” , “**rdate -s ntp1.inrim.it**”.

Avendo attivato il servizio time sulla nostra vm possiamo fare : “**rdate -s 192.168.100.62**”.

**hwclock** : gestisce il clock hardware del pc : “**hwclock -w**” (scrive l'ora del SO sul bios).

```
GNU nano 6.2
# Log statistics if you want statistics to be logged.
#statsdir /var/log/ntpstats/
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# Specify one or more NTP servers.
server ntp.dsi.unive.it
# pool.ntp.org is the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP# #104525). See http://www.pool.ntp.org/join.html for
# more information.
pool 0.ubuntu.pool.ntp.org tburst
pool 1.ubuntu.pool.ntp.org tburst
pool 2.ubuntu.pool.ntp.org tburst
pool 3.ubuntu.pool.ntp.org tburst

# Use Ubuntu's ntp server as a fallback.
pool ntp.ubuntu.com

# Access control configuration; see /usr/share/doc/ntp-doc/html/accept.html
# for details. The well-known ports 123 and 13.1 are typically
# allowed through firewalls.

# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# If you want to allow connections from anywhere, don't allow configuration.
restrict -1 default kod notrap nomodify nopeer noquery limited
restrict -1 default kod notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1

# needed for adding pool entries
restrict 192.168.1.0掩码 255.255.255.0 noquery
# Clients from this (example) subnet have unlimited access, but only if
# cryptographically authenticated.
restrict 192.168.1.0掩码 255.255.255.0 noquery

# If you want to provide time to your local subnet, change the mask line
# to something like 192.168.1.0掩码 255.255.255.0
# restrict 192.168.1.0掩码 255.255.255.0 noquery

# If you want to receive time broadcasts or your local subnet, do something
# like this:
# broadcast 192.168.1.1掩码 255.255.255.0
# broadcast 192.168.1.1掩码 255.255.255.0 noquery
```

```
date: Could not connect socket: Network is unreachable
ubuntu@hp-12:~$ rdarte
Comando «rdarte» non trovato, si intendeva forse:
  comando «rdate» da deb rdate (1:1.10.2-1build1)
Provare: sudo apt install <nome deb>
ubuntu@hp-12:~$ 
ubuntu@hp-12:~$ sudo rdate -s 192.168.100.79
ubuntu@hp-12:~$ 

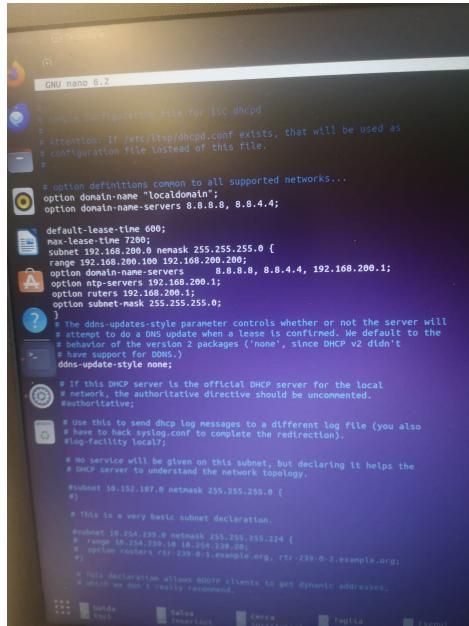
ubuntu@hp-12:~$ sudo rdate -s 192.168.100.79
^Z [5]+  Fermato                  sudo rdate -s 192.168.100.79
ubuntu@hp-12:~$ sudo rdate -s 192.168.100.62
ubuntu@hp-12:~$ rdate -s ntp.dsi.unive.it
ubuntu@hp-12:~$ 
^C
ubuntu@hp-12:~$ nano /etc/ntp.conf
ubuntu@hp-12:~$ sudo hwclock -w
ubuntu@hp-12:~$ 
```

# DHCP - installazione

Affinché la nuova scheda venga vista è necessario riavviare la vm.

Per trovare la scheda di riavvio : “**dmesg | grep ens**”. Dopo l'avvio configurate la scheda con netplan o al volo usando il comando ifconfig : **“ifconfig ens38 192.168.200.1 netmask 255.255.255.0 up”**.

Ora editiamo il file “**/etc/dhcp/dhcpd.conf**” :



```
GNU nano 6.2
# Sample configuration file for ISC dhcpcd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
option domain-name "localdomain";
option domain-name-servers 8.8.8.8, 8.8.4.4;
#
# default lease time 600;
# max-lease-time 7200;
subnet 192.168.200.0 netmask 255.255.255.0 {
    range 192.168.200.100 192.168.200.200;
    option domain-name-servers 8.8.8.8, 8.8.4.4, 192.168.200.1;
    option ntp-servers 192.168.200.1;
    option routers 192.168.200.1;
    option subnet-mask 255.255.255.0;
}
#
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# 'none' style, since most clients (e.g. dhclient) expect 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;
#
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;
#
# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local;
#
# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
#subnet 10.102.187.0 netmask 255.255.255.0 {
#
# This is a very basic subnet declaration.
#subnet 10.254.239.0 netmask 255.255.255.224 {
#    range 10.254.239.10 10.254.239.28;
#    option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#
# This declaration allows hosts clients to get dynamic addresses,
# which we don't really recommend.
#}
```

Riavviamo il servizio : “**service isc-dhcp-server restart**”, “**systemctl restart isc-dhcp-server**”.

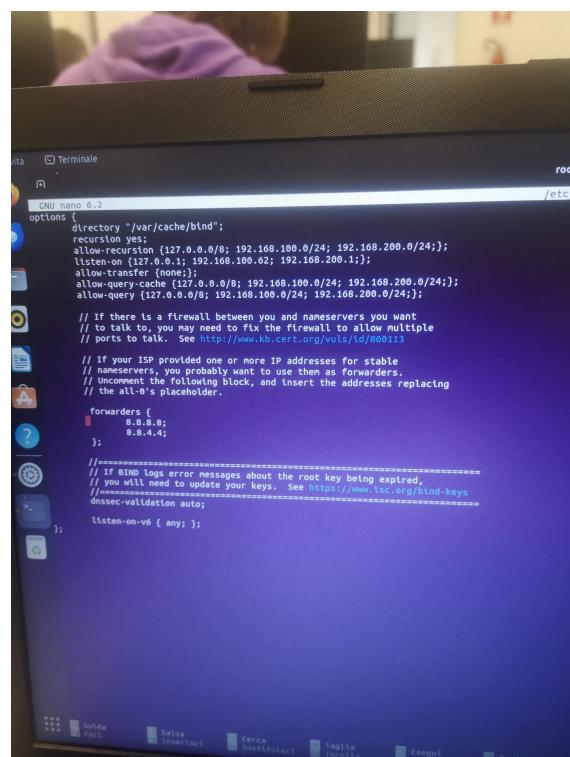
Infine, possiamo controllare l'esito tramite il comando : “**journalctl -xe**”. Per sapere se funziona, dobbiamo creare una seconda vm e agganciarla alla stessa rete del nostro server dhcp (VMnet2).

# Installazione DNS

NB : nelle seguenti operazioni dovremo sostituire 192.168.64.129 con l'ip della nostra macchina virtuale o del portatile hp (192.168.100.62) e la rete 192.168.64.0/24 con la rete di cui fa parte la nostra vm o la rete 192.168.100.0/24.

Collegiamoci via ssh alla VM (nome utente = ubuntu) : “**ssh** ubuntu@192.168.100.62”, poi (sempre da root) installiamo i pacchetti necessari per bind : “**apt install bind9 bind9utils bind9-doc**”. Ora editiamo il file che indica a bind la configurazione di partenza, forzando il solo uso a ipv4 : cambiamo o aggiungiamo la riga “**OPTIONS=-u bind”** in “**OPTIONS=-u bind -4”**”. Infine, riavviamo bind con “**service bind9 restart**” o “**systemctl restart bind9**”.

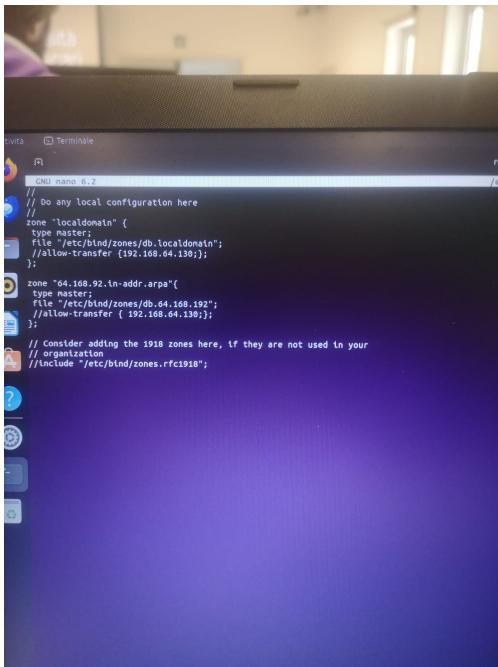
Cominciamo con la configurazione di bind. Prima di tutto dobbiamo specificare che opzioni di bind vogliamo attive. Editiamo quindi il file “**/etc/bind/named.conf.options**” :



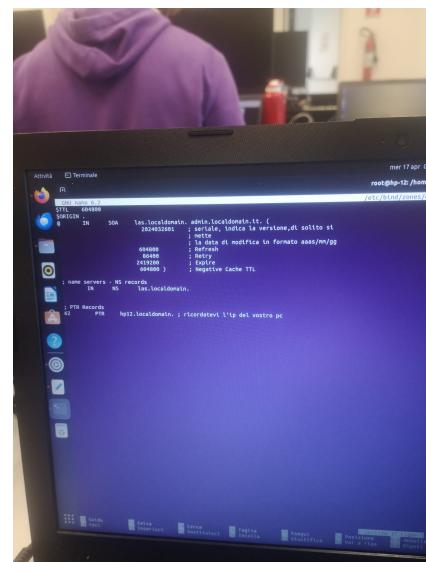
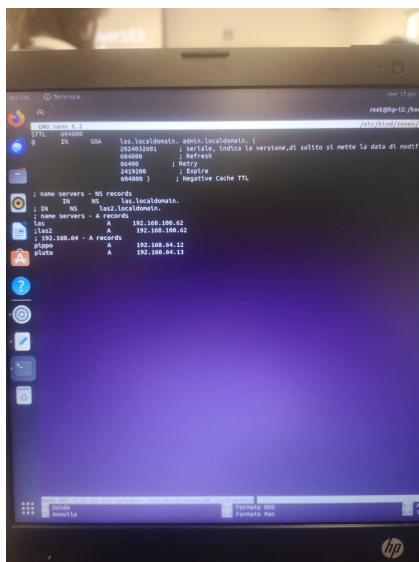
```
GNU nano 6.2
options {
    directory "/var/cache/bind";
    recursion yes;
    allow-recursion { [172.0.0.0/8; 192.168.100.0/24; 192.168.200.0/24];
                      [127.0.0.1; 192.168.100.62; 192.168.200.1]; };
    allow-transfer { [none]; };
    allow-query-cache { [127.0.0.0/8; 192.168.100.0/24; 192.168.200.0/24];
                       [allow-query { [127.0.0.0/8; 192.168.100.0/24; 192.168.200.0/24]; }; ];
    };
    allow-query { [127.0.0.0/8; 192.168.100.0/24; 192.168.200.0/24]; };
    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113
    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.
    forwarders {
        8.8.8.8;
        8.8.4.4;
    };
    //========================================================================
    // If BIND logs error messages about the root key being expired,
    // you'll need to update your keys. See https://www.tsc.org/bind-keys
    dnssec-validation auto;
};

listen-on-v6 { any; };
```

Editiamo ora il file “**/etc/bind/named.conf.local**” :



Procediamo con la creazione dei file che descrivono la definizione della zona. Se non esiste, creiamo la directory delle zone “**mkdir -p /etc/bind/zones/**”. Poi definiamo la zona diretta creando il file “**/etc/bind/zones/db.ubuntu.localdomain**”, definiamo anche la zona inversa creando il file “**/etc/bind/zones/db.64.168.192**” :



Testiamo ora le varie configurazioni fatte. Anzitutto vediamo se bind è configurato correttamente con : “**named-checkconf**” e se il comando non ritorna nulla, la configurazione è ok. Poi controlliamo se la definizione delle zone è corretta. Iniziamo dalla zona diretta : “**named-checkzone**

**localdomain.it /etc/bind/zones/db.localdomain**” , poi controlliamo anche la zona inversa con “**named-checkzone <nostra\_rete> /etc/bind/zones/ db.192.168.64**” , “**named-checkzone <nostra\_rete> /etc/bind/zones/ db.192.168.200**”.

Procediamo anche con l’attivazione del nostro dns : “**systemctl restart bind9**” , “**ufw allow bind9**”. Infine, un po di prove con : “**host hp-12.ubuntu.localdomain. 127.0.0.1**” , “**host hp-12.ubuntu.localdomain. 192.168.100.62**” , “**host www.unive.it 192.168.100.62**”, e “**host www.google.it 192.168.100.62**”.

```

Attività Terminal
ubuntughp-12: ~ $ host www.google.it hp-12.ubuntu.localdomain
Host ubuntu.localdomain not found: 2(SERVFAIL)
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:

Host www.google.it has address 142.251.209.3
www.google.it has IPv6 address 2a00:1450:4002:4102::2003
ubuntughp-12: ~ $ host www.unive.it 192.168.100.62
Using domain server:
Name: 192.168.100.62
Address: 192.168.100.62#53
Aliases:
www.unive.it has address 157.138.7.88
ubuntughp-12: ~

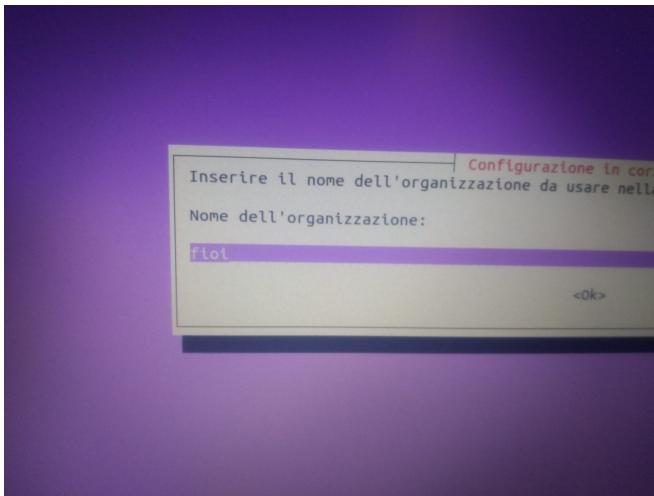
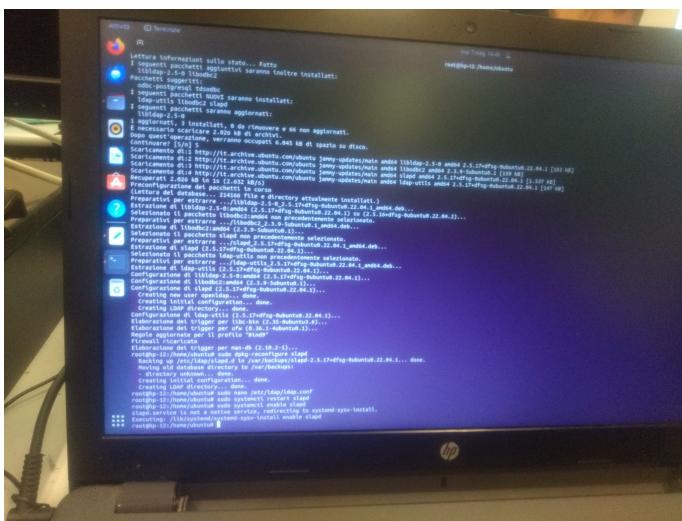
root@hp-12:/home/ubuntu# host hp-12.ubuntu.localdomain. 127.0.0.1
Using domain server:
Name: 127.0.0.1
Address: 127.0.0.1#53
Aliases:
hp-12.ubuntu.localdomain has address 192.168.100.62
root@hp-12:/home/ubuntu# host hp-12.ubuntu.localdomain. 192.168.100.62
Using domain server:
Name: 192.168.100.62
Address: 192.168.100.62#53
Aliases:
hp-12.ubuntu.localdomain has address 192.168.100.62
root@hp-12:/home/ubuntu# host www.google.it 192.168.100.62
Using domain server:
Name: 192.168.100.62
Address: 192.168.100.62#53
Aliases:
www.google.it has address 142.250.180.131
www.google.it has IPv6 address 2a00:1450:4002:4002::2003
root@hp-12:/home/ubuntu#

```

# Esercitazione 4 : Autenticazione

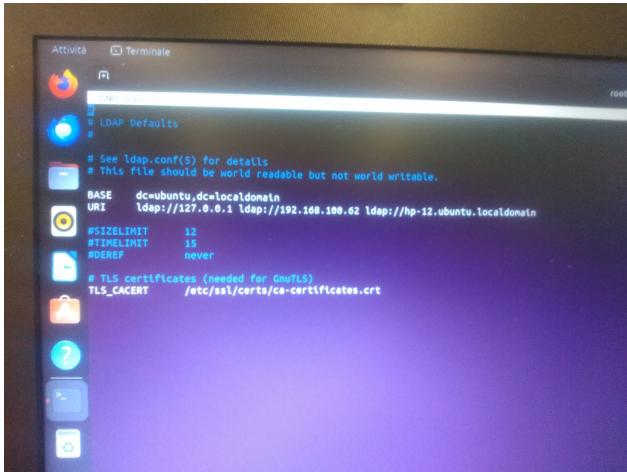
## Installazione LDAP

Installiamo il server ldap con il comando : “**sudo apt install salpo ldap-utils**” ed inseriamo 2 volte la password di amministratore (Ubuntu1234). Configuriamo il server ldap con : “**sudo dpkg.reconfigure slapd**” e poi scegliamo un pò di cose tra le opzioni che ci propongono (NON lasciare vuota la configurazione, scegliere il nome dell’organizzazione [fioi], impostiamo password e formato di memorizzazione [in particolare per il backend scegliamo come consigliato MBD], selezioniamo no alla rimozione del database ed infine spostiamo il vecchio database).



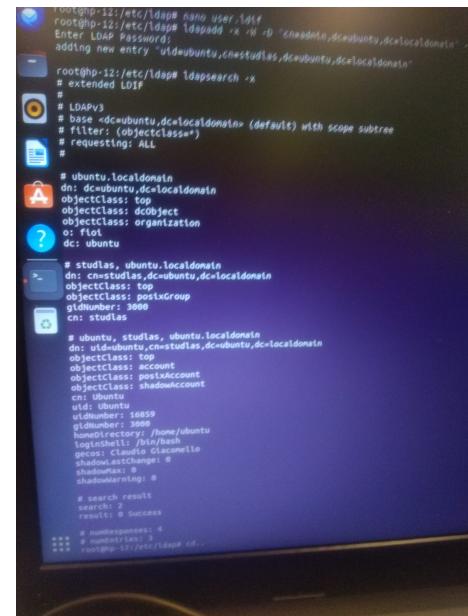
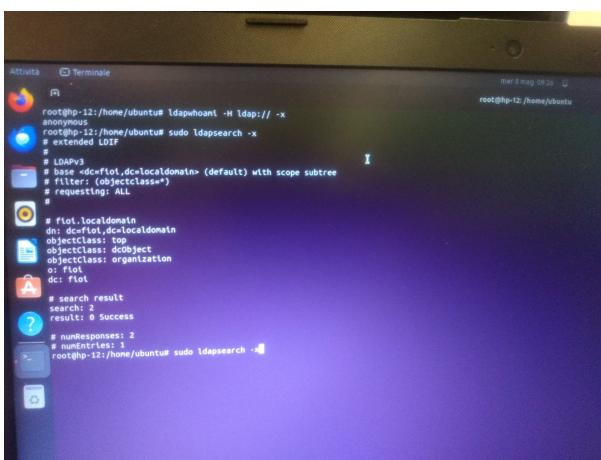
# Lightweight Directory Access Protocol - LDAP - DIT

Modifichiamo il file di configurazione principale ldap.conf aggiungendo il dominio di esempio impostato in precedenza ed inseriamo delle righe :



Poi riavviamo il demone slapd e abilitiamolo ad avviarsi all'avvio con i comandi : “**sudo systemctl restart slapd**” e “**sudo systemctl enable slapd**”.

Per verificare il corretto funzionamento della connessione LDAP dobbiamo eseguire i seguenti comandi : “**sudo ldapwhoami -H ldap:// -x**” e “**sudo ldapsearch -x**”

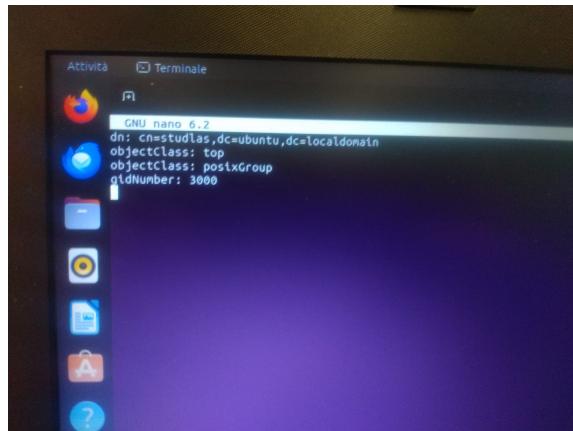
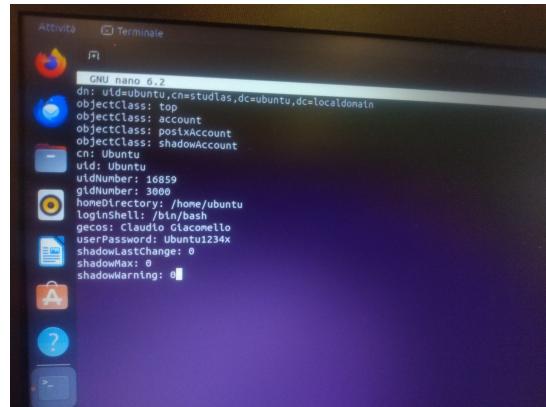


# LDAP - Creazione gruppi e utenti (e verifica)

In ldap, per creare gruppi e utenti è necessario creare un file che definisce l'oggetto da creare come schema ldap. Creiamo il file *group.ldif* per il gruppo "studlas". Stessa cosa per creare l'utente dentro a *user.ldif*.

```
# Ubuntu
# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
root@hp-12:~/home/ubuntu# nano /etc/ldap/group.ldif
root@hp-12:~/home/ubuntu# nano /etc/ldap/ldap.conf
root@hp-12:~/home/ubuntu# ldapadd -x -W -D "cn=admin,dc=ubuntu,dc=localdomain" -f group.ldif
group.ldif: No such file or directory
root@hp-12:~/home/ubuntu# cd ..
root@hp-12:~/home/ubuntu# ls
backup.sh  clao.sh  echo.sh  funcquadrato.sh  hui.sh  log.sh  pipoo.txt  script3.sh  sudo
canc.sh  commenti.sh  elif.sh  funcquadrato.sh  login_number.sh  logoff.sh  script1.sh  test
case2.sh  data.tx  file.sh  func.sh  tflogin.sh  mail.sh  script2.sh  seript.sh
for.sh  hosts.deny  tf.sh  Music.sh  script3.sh  serivewash  test
root@hp-12:~/home/ubuntu# cd ..
root@hp-12:~/# ls
root@hp-12:~/# cd /etc/ldap
root@hp-12:/etc/ldap# ls
group.ldif  ldap.conf  ldif2db  ldif2db.pl  ldapadd  ldapsearch  ldapwhoami  ldapx2db
root@hp-12:/etc/ldap# ldapadd -x -W -D "cn=admin,dc=ubuntu,dc=localdomain" -f group.ldif
Enter LDAP Password:
adding new entry "cn=studlas,dc=ubuntu,dc=localdomain"
root@hp-12:/etc/ldap#
```

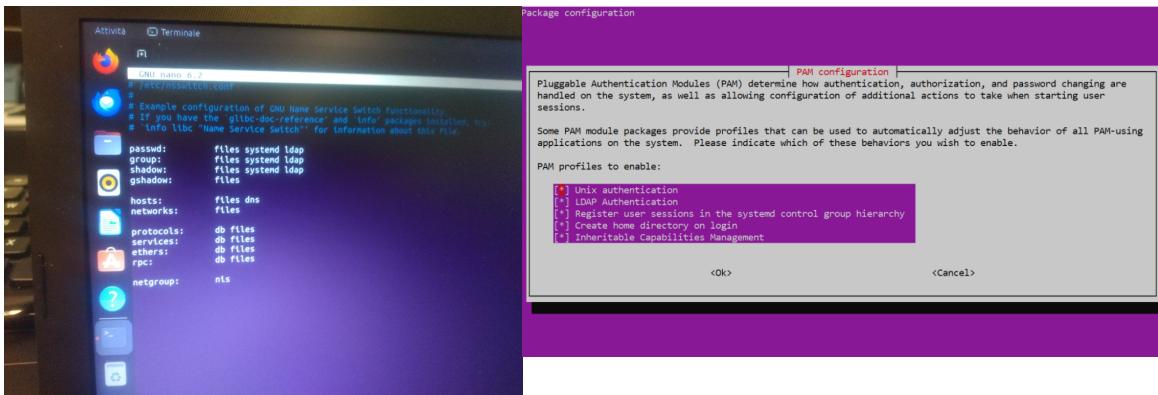


Poi per testare ri-eseguiamo il comando "**ldapsearch -x**".

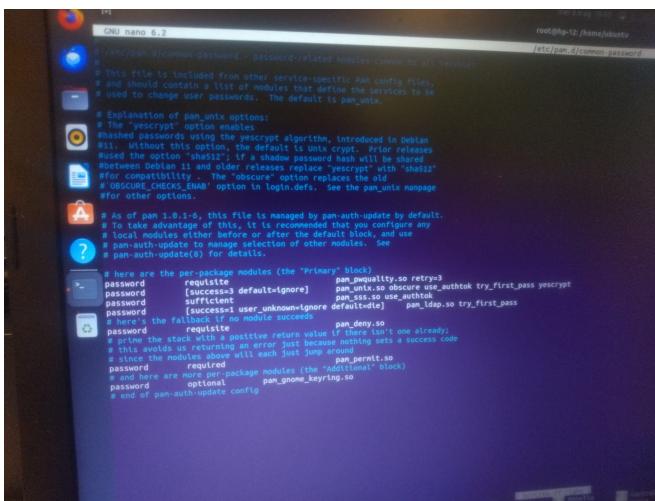
## LDAP - nss e pam

Installiamo le estensioni ldap per nsswitch e pam : "**sudo apt install libnss-ldap libpam-ldap**".

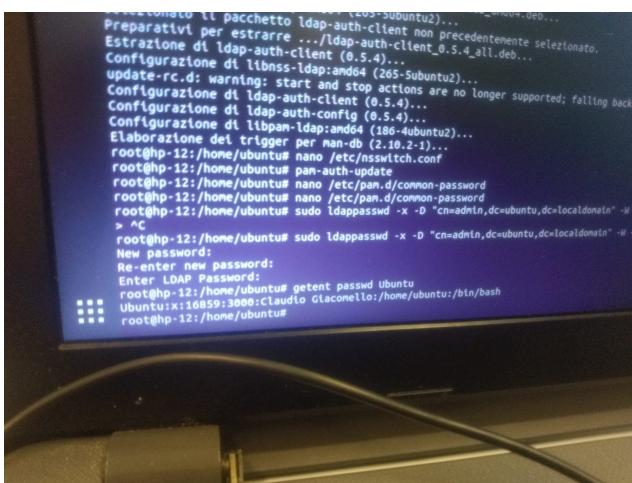
Ora modifichiamo il file "**/etc/nsswitch.conf**" e poi successivamente utilizzare pam-auth-update per sistemare la configurazione di pam con "**pam-auth-update**"



Editiamo il file “**/etc/pam.d/common-password**” in modo da cambiare la riga 26 nella maniera corretta. Poi, cambiamo la password dell’utente.



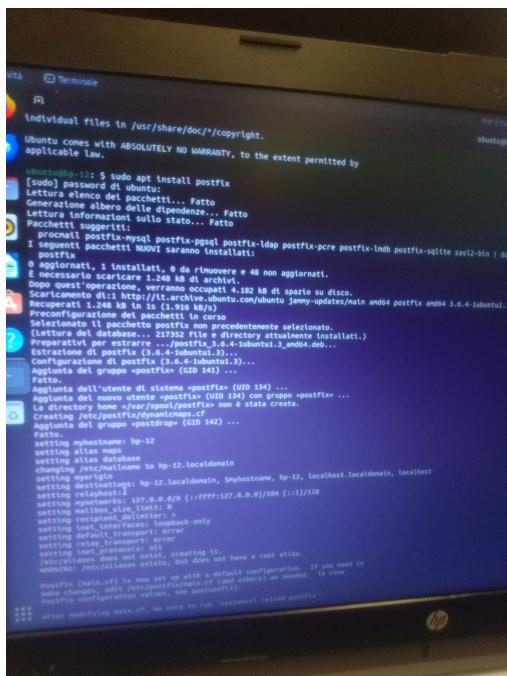
Infine testiamo se tutto funziona con i comandi “getent” :



# Esercitazione 5 : Posta Elettronica

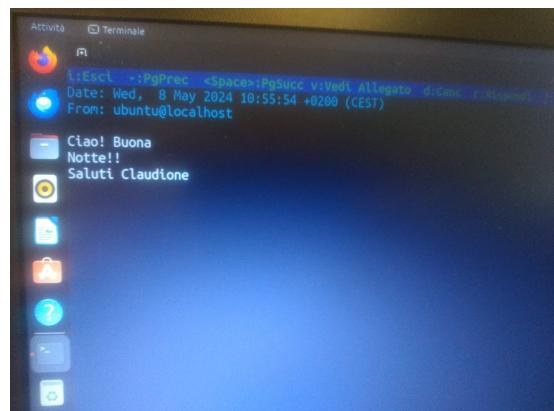
## Postfix : Installazione e test

Comando per installare postfix : “**sudo apt install postfix**” e scegliamo “Local only”. Come System Domain mettiamo “<hostname>.localdomain”.



Testiamo con “**telnet localhost 25**” e poi configuriamo in “**/etc/postfix**” i file “**master.cf**” e “**main.cf**”.

```
: ~ % service postfix restart
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
220 hp-12 ESMTP Postfix (Ubuntu)
ehlo localhost
250-hp-12
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITTIME
250-DSN
250-SMTPUTF8
250 CHUNKING
mail from: ubuntu@localhost
250 2.1.0 Ok
rcpt to: ubuntu@localhost
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Ciao! Buona
Notte!!
Saluti Claudio
.
250 2.0.0 Ok: queued as CB6F180EFA
quit
221 2.0.0 Bye
Connection closed by foreign host.
ubuntu@hp-12:~
```



```

# Sample auto.master file
# This is a 'master' automounter map and it has the following format:
# mount-point [map-type]:[format]:[map [options]]
# For details of the format look at auto.master(5).
#
# /misc  /etc/auto.misc -timeout=60
# /smb  /etc/auto.smb
# /net   /etc/auto.net
# /mnt   /etc/auto.mnt

#
# NOTE: mounts done from a hosts map will be mounted with the
# "suid" and "nosuid" options unless the "suid" and "dev"
# options are explicitly given.
# /net  -hosts

# Include /etc/auto.master.d/*_autofs
# To add an extra map using this mechanism you will need to add
# two configuration items - one /etc/auto.master.d/extra.autofs file
# (using the same line format as the /etc/auto.master file)
# and one entry in /etc/auto.master (e.g., /etc/auto.extra or an auto.extra NIS map)
# that is referred to by the extra.autofs file.
#
+dir:/etc/auto.master.d

# If you have Fdbs set up and the related binaries, either
# built as part of autofs or via the util-linux package,
# uncomment this line to use the Fdbs program map to access
# your Fdbs mounts.
# /nfs4 /usr/sbin/fdbs-map.nfs4 nobind
#

```

```

##@controlli spam mtamail
smtpd_delay_rejects yes
smtpd_helo_required yes
smtpd_helo_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    check_helo_access hash:/etc/postfix/helo_access,
    reject_non_fqdn_hostname,
    reject_invalid_hostname,
    permit
smtpd_sender_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_non_fqdn_sender,
    reject_unknown_sender_domain,
    permit
smtpd_recipients_restrictions =
    reject_unauth_pipeline,
    reject_non_fqdn_recipient,
    reject_unknown_recipient_domain,
    permit_mynetworks,
    permit_sasl_authenticated,
    reject_unauth_destination,
    check_sender_access hash:/etc/postfix/sender_access,
    check_recipient_access hash:/etc/postfix/recipient_access,
    check_helo_access hash:/etc/postfix/secondary_mx_access,
    reject_rbl_client zen.spamhaus.org,
    #postgrey
    #check_policy_service inet:127.0.0.1:10023
    permit
#anavils
#content_filter = smptpanavils:127.0.0.1:10024
#smptpanavils_destination_concurrency_limit = 1

```

Dopo aver editato i file, facciamo un test per vedere che tutto funzioni :

```

^] ^C
quit
Connection closed by foreign host.
: $ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is ']'.
220 hp-12 ESMTP Postfix (Ubuntu)
ehlo localhost
250-hp-12
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250-CHUNKING
mail from: ubuntu@localhost
250 2.1.0 Ok
rcpt to: f1o1@localhost
550 5.1.1 <f1o1@localhost>: Recipient address rejected: User unknown in local recipient table
rcpt to: las@localhost
550 5.1.1 <las@localhost>: Recipient address rejected: User unknown in local recipient table
rcpt to: ubuntu@localhost
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Ciao!
Zio peral
Saluti
Claudio Giacomello
.
250 2.0.0 Ok: queued as ADCE2810A6
*** quit

```