

Laboratorio di Calcolo

Prontuario per la configurazione di un ambiente per la programmazione

Aggiornato 2024-09-17



SAPIENZA
UNIVERSITÀ DI ROMA

Simone Di Micco, Numa Zorzi

Contents

1	Configurare un PC Windows	3
1.1	Windows Subsystem for Linux	5
1.2	Macchina virtuale	10
1.2.1	La via più semplice	15
1.3	Macchina Virtuale <i>Custom</i>	16
1.3.1	Ubuntu	20
1.3.2	Ho un computer troppo datato! Che cosa faccio?	25
1.3.3	Architettura ARM	27
1.3.4	Troubleshooting per VM	27
1.4	Dual boot	29
1.4.1	Architettura x64	29
1.4.2	Altre architetture	38
2	MacOS	39
2.1	Approccio 'Nativo' - le Command Line Tools	40
2.1.1	Homebrew	40
2.1.2	GCC	43
2.1.3	Emacs	43
2.1.4	Python3	44
2.2	Virtualizzazione - UTM	46

Abstract

Questo documento è un prontuario pensato per essere messo a disposizione degli studenti di fisica che si approcciano per la prima volta al mondo della programmazione. Si partirà dalle basi, dandovi gli strumenti per poter iniziare a programmare, iniziando quindi dalla configurazione del PC. Configurare un PC per programmare è in realtà un'operazione molto semplice, perciò se siete dei neofiti non abbiate timore poiché l'operazione non richiederà particolari conoscenze tecniche e non più di una mezz'ora di tempo (escludendo i tempi necessari a scaricare da internet i vari programmi, cosa che dipende dalla velocità della connessione a disposizione). Questa guida tenderà ad essere aggiornata anno dopo anno, aggiungendo ogni volta materiale che pensiamo possa esservi utile, raccolto tra tutti i canali del corso, in modo da fornire un prontuario sempre più completo a cui potrete far riferimento idealmente anche negli anni successivi della vostra carriera. Perciò assicuratevi di avere sempre la versione più aggiornata, cioè quella dell'anno accademico in corso.

1 Configurare un PC Windows

Come vedrete (o avete già visto), in laboratorio si usa Linux. Potreste quindi pensare che se avete un Pc Windows siete nei pasticci. Non temete! Il Pc può essere utilizzato, ma dovrete perderci un po' di tempo :-)



Figure 1: Il caratteristico logo di linux è il pinguino

Curiosità: Alcuni di voi potrebbero non sapere cosa sia un sistema operativo. Per farla breve, il vostro computer in sé non è che un sasso molto elaborato, in cui grazie agli impulsi elettrici che scorrono nei circuiti da l'impressione di essere un sasso molto fancy. Il sistema operativo è il programma più basilare che gira sul vostro computer, quello che fa in modo che gli impulsi elettrici vadano dove devono andare. Si dice che trasforma il linguaggio macchina (cioè gli impulsi elettrici), in linguaggio umano (cioè le immagini e i testi con cui possiamo interagire noi). Di sistemi operativi (per pc) ne esistono 3 grandi famiglie: Windows (di proprietà della Microsoft), MacOS (della Apple) e Linux (progetto open source, cioè ad accesso libero e gratuito).

Ci sono tre possibili approcci che possiamo seguire: utilizzare WSL, un sistema sviluppato piuttosto recentemente per mettere a disposizione degli utenti Windows un sottosistema Linux, utilizzare una macchina virtuale, oppure installare Linux in dual boot. Vediamo che significano queste ultime due cose.

Una **macchina virtuale** è un programma che fa finta di essere un computer fisico. Questo significa che sul vostro PC Windows andate a scaricare un programma. All'interno di questo programma voi fate finta di avere un altro computer, su cui installate un sistema

operativo diverso. Questo approccio offre alcuni vantaggi e alcuni svantaggi.

Tra i pro abbiamo che è di certo l'approccio più facile, veloce e sicuro. Configurare una macchina virtuale richiede pochissima capacità tecnica e, anche nel caso si sbagliasse qualcosa, si può in ogni momento dire al programma di cancellare tutto e ricominciare da capo. Dall'altra parte della medaglia abbiamo, però, che con una macchina virtuale si possono fare solo operazioni che non richiedono molte risorse computazionali. Infatti, sebbene il programma simula un secondo computer, in realtà il vostro computer resta comunque uno solo. Perciò possiamo dire che è costretto a fare la fatica di due computer contemporaneamente. In ogni caso per gli scopi dell'intero corso, a meno che non abbiate un computer molto economico o molto vecchio, avrete tutta la potenza sufficiente per poter fare quanto richiesto. L'approccio alla macchina virtuale è quindi consigliata se avete particolare timore di fare danni (ma vi avverto che in ogni caso anche con l'altro approccio non si possono fare grandi danni, al massimo si perde tempo).

Il **dual boot** significa invece installare sul vostro computer fisico un altro sistema operativo al fianco di Windows. Il computer può eseguire un solo sistema operativo per volta, perciò all'accensione vi sarà chiesto ogni volta quale dei due volete usare.

Tra i pro abbiamo che il computer potrà sfruttare il massimo delle risorse, col risultato che avrete un sistema più fluido e molto più performante. Se avete intenzione di proseguire sulla strada della fisica presto o tardi avrete bisogno di quella potenza di calcolo, quindi rassegnatevi al fatto che prima o poi dovete leggere questo capitolo della guida. Tra i contro abbiamo una difficoltà tecnica appena superiore, che nel peggior dei casi potrebbe costarvi la perdita di tutti i dati sul pc (è sempre bene fare una copia quindi dei file importanti). Ma non temete, a meno che non siate particolarmente bravi e fantasiosi a fare casino, non c'è possibilità di rompere il computer.

Prima di iniziare solo un'ultima precisazione. In realtà è impreciso dire che Linux sia un sistema operativo. Linux in realtà è un kernel, cioè solo la componente di base (dall'inglese *kernel=nucleo*). Da solo non sarebbe sufficiente per far funzionare un computer. Intorno a

questo nucleo diverse comunità hanno creato diversi sistemi operativi. Tutti i sistemi operativi creati a partire dal kernel Linux sono dette *distribuzioni di Linux* (o solo distribuzioni per brevità, o ancora meglio, solo distro). Di distribuzioni ne esistono moltissime. Un motivo per cui ne esistono così tante è che ogni comunità si è focalizzata su aspetti diversi di personalizzazioni e utilità. O almeno questo in teoria. In pratica è probabilmente che alla gente piace creare nuove distribuzioni per diletto. In ogni caso questo viene a nostro vantaggio poiché grazie a questa sovrabbondanza è possibile trovare una distribuzione di Linux adatta a qualsiasi necessità. Si dice spesso infatti che "Linux gira anche sui sassi".

Ovviamente non starò qui a fare un tutorial per ogni distribuzione, se siete curiosi su internet potete trovare un'infinità di materiale e supporto dalle community, a patto che mastichiate un po' d'inglese, sebbene le distro principali abbiano una vasta comunità anche in lingua italiana. Invece io mi concentrerò su una o due distro che vanno più che bene per iniziare (ma che potrebbero anche accompagnarvi per sempre).

Bene dopo avervi ammorbato per bene, iniziamo.

1.1 Windows Subsystem for Linux

Il Windows Subsystem for Linux (d'ora in avanti WSL) è una feature di Windows che ci permette, nella pratica, di utilizzare un terminale Windows come fosse un terminale Unix-based (i.e. il terminale di un sistema Linux o MacOS). Senza perdersi in tecnicismi, si tratta di un interprete che automaticamente trasformerà i nostri comandi Bash (il linguaggio predefinito del terminale Linux) in comandi Windows. Se siete in possesso di un PC che supporta WSL, il mio consiglio è di farne uso. È di gran lunga il modo più intuitivo e meglio integrato per lavorare su Linux tramite un sistema Windows. WSL è una funzione piuttosto sofisticata e recente di Windows, pertanto è disponibile solo dalla versione 10 in poi. Per verificare la versione installata sul vostro PC, cliccate su Start > Impostazioni > Sistema > Informazioni. Scorrendo in basso troverete la versione attualmente installata sul vostro dispositivo. È importante verificare che il vostro dispositivo supporti la virtualizzazione. Il sistema WSL infatti, nonostante sia completamente integrato all'interno dell'interfaccia Windows, utilizza lo stesso sistema di una virtual machine per tradurre le istruzioni. Per

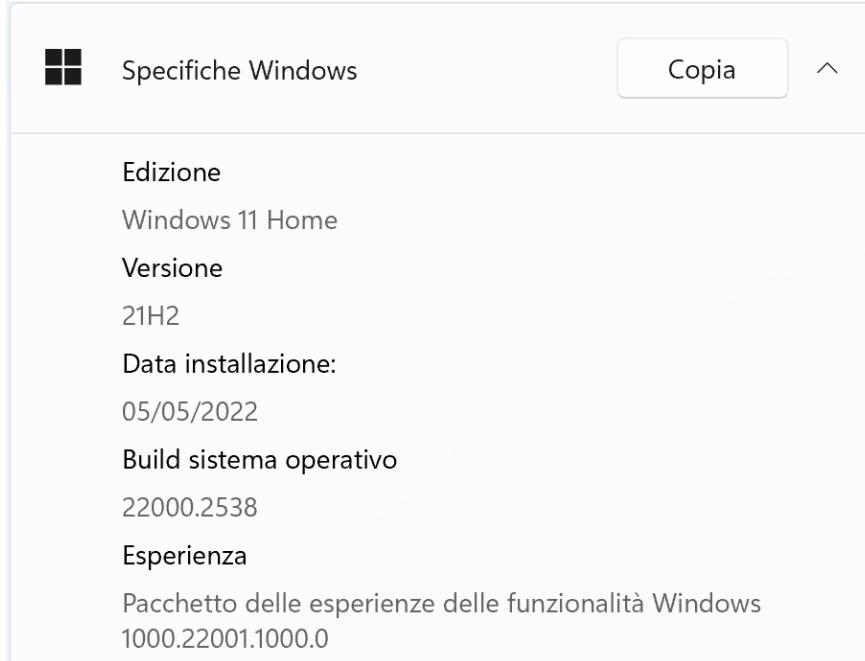


Figure 2: Affinché sia possibile installare WSL, dobbiamo disporre come di Windows 10 Build 19041 o successive. Qualsiasi installazione di Windows 11 supporta nativamente WSL.

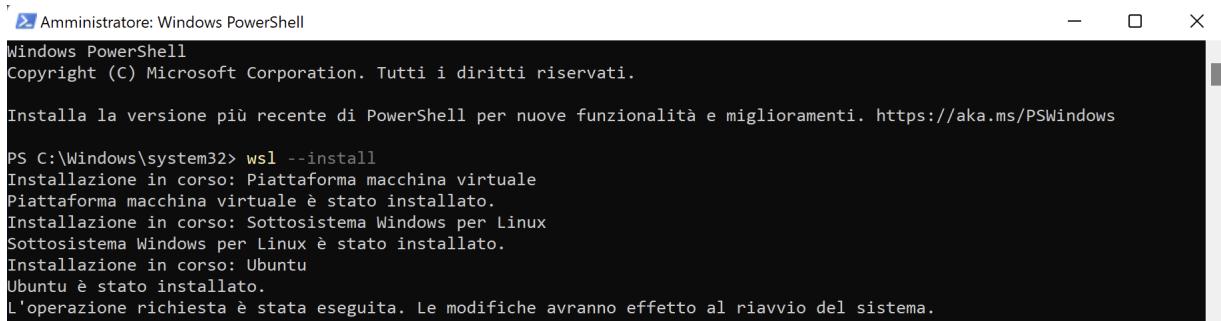
questa verifica è sufficiente seguire lo stesso procedimento della sezione 1.2. Una volta verificata la compatibilità, l'installazione è semplicissima. Dal menù Start, cerchiamo "Windows PowerShell", clicchiamo con il tasto destro sull'icona corrispondente, selezioniamo "Esegui come amministratore" e autorizziamo il programma a modificare il sistema operativo. Si aprirà la schermata di un terminale (i più precisi diranno che si tratta più propriamente di un simulatore di terminale ma la differenza è al momento trascurabile). Da qua sarà sufficiente eseguire il comando

```
wsl --install
```

per installare WSL sul dispositivo. Il comando installa la distribuzione Linux di default, cioè Ubuntu. Qualora si volesse installare¹ una distribuzione differente, si deve usare il comando:

```
wsl --install -d <Nome Distribuzione>
```

¹A meno di casi particolari il consiglio rimane di installare la distribuzione predefinita.



```
[+] Amministratore: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tutti i diritti riservati.

Installa la versione più recente di PowerShell per nuove funzionalità e miglioramenti. https://aka.ms/PSWindows

PS C:\Windows\system32> wsl --install
Installazione in corso: Piattaforma macchina virtuale
Piattaforma macchina virtuale è stato installato.
Installazione in corso: Sottosistema Windows per Linux
Sottosistema Windows per Linux è stato installato.
Installazione in corso: Ubuntu
Ubuntu è stato installato.
L'operazione richiesta è stata eseguita. Le modifiche avranno effetto al riavvio del sistema.
```

Figure 3: Il tipico output di una corretta installazione di WSL. Sarà sufficiente riavviare il sistema per iniziare a usarlo.

Per vedere le distribuzioni disponibili all’installazione, possiamo usare il comando:

```
wsl --list --online
```

Una volta completata l’installazione, riavviamo il sistema per iniziare a utilizzare le nuove funzioni. Dobbiamo comunque verificare che WSL sia attivo. Per farlo, digitare il comando

```
wsl --status
```

Se la risposta è negativa, dobbiamo attivare WSL manualmente. Per farlo, eseguite una PowerShell come amministratore con il comando

```
start-process PowerShell -verb runas
```

e all’interno di questa PowerShell eseguite il comando

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

È il momento di creare il nostro account Ubuntu. Per fare ciò, cerchiamo ’Ubuntu’ dal menù Start e lo apriamo come un programma qualsiasi (vedi fig. 4). Il sistema ci chiederà di inserire uno username e una password, che non devono necessariamente essere uguali a quelli dell’account Windows. Attenzione! Quando inseriamo la password ci sembrerà di non scrivere nulla. Il sistema in realtà registra i caratteri che stiamo inserendo, ma non produce alcun output visivo. Semplicemente scriviamo la password scelta con attenzione e premiamo Invio alla fine. Una volta scelti username e password, il sistema finalizzerà l’installazione di

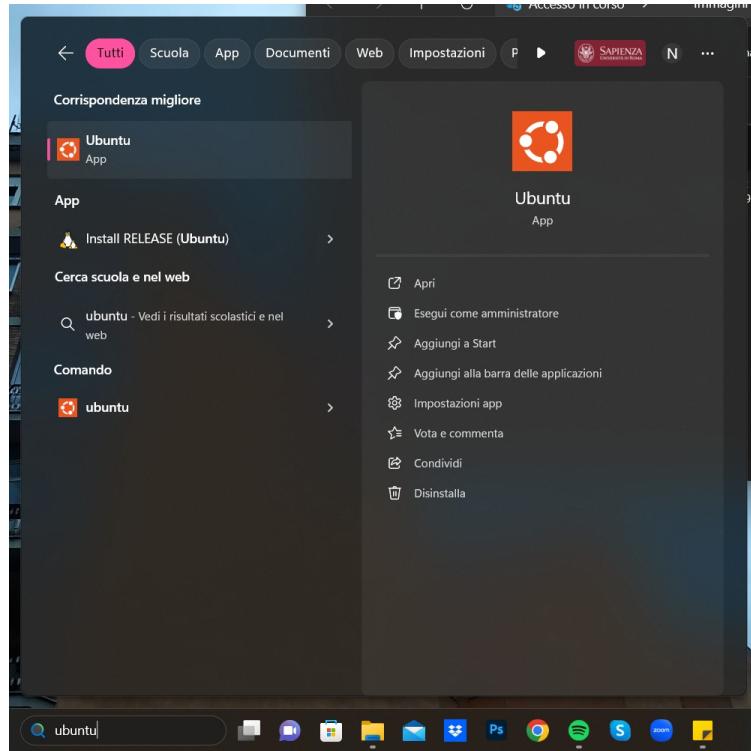


Figure 4: Una volta completata l'installazione e riavviato il sistema, dovremmo trovare Ubuntu direttamente dal menù Start.

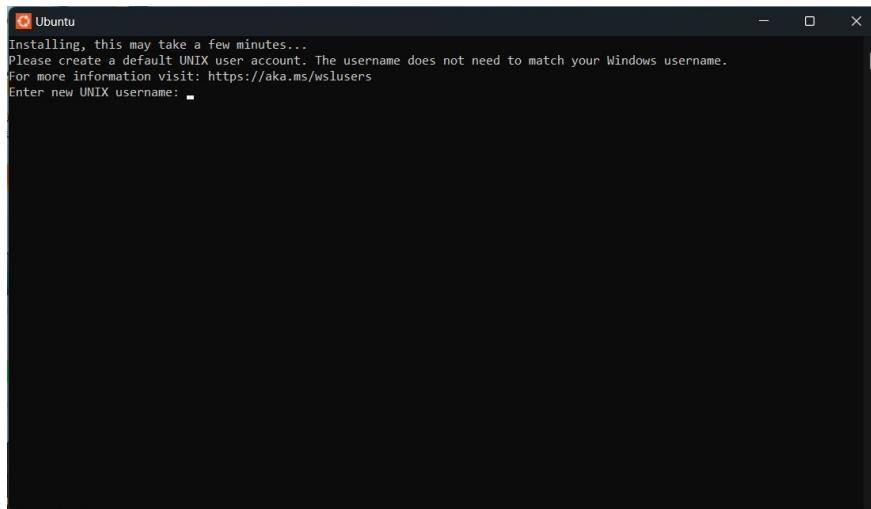
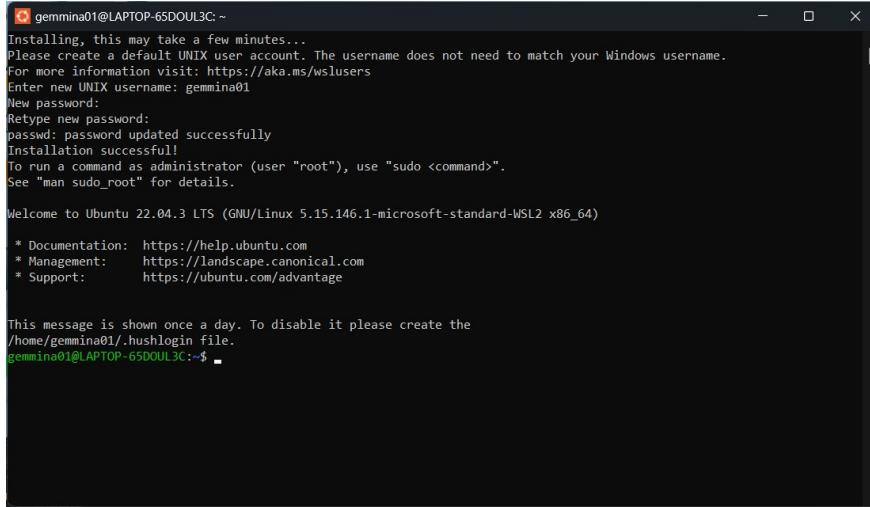


Figure 5: La schermata iniziale che visualizzeremo al primo avvio di Ubuntu.



The screenshot shows a terminal window titled "gemmina01@LAPTOP-65DOUL3C: ~". The output of the installation process is displayed:

```
gemmina01@LAPTOP-65DOUL3C: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: gemmina01
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/gemmina01/.hushlogin file.
gemmina01@LAPTOP-65DOUL3C:~$
```

Figure 6: L'output che visualizzeremo una volta completata l'installazione di Ubuntu.

Ubuntu. Ultimato questo procedimento, si avvierà in automatico il sottosistema Ubuntu. In generale, riconosciamo di essere all'interno di WSL perché ogni riga del terminale inizia con username@nomedispositivo. Per chiudere Ubuntu, apriamo una scheda di Windows PowerShell e digitiamo il comando

```
wsl --shutdown
```

D'ora in poi potremo avviare WSL direttamente all'interno di una scheda PowerShell, semplicemente facendo uso del comando

```
wsl
```

All'interno di una scheda WSL, il sistema si comporta esattamente come un sistema Linux, sia per la navigazione del file system che per chiamare i vari comandi e programmi che installeremo.

Dal terminale WSL potete accedere alle cartelle di Windows con il percorso /mnt/c/ Ad esempio, con il comando

```
ls /mnt/
```

vi verranno mostrate le cartelle in C:\di Windows. Da Windows invece potete vedere il sottosistema WSL tramite la cartella "Linux" presente nel file browser oppure da PowerShell tramite il percorso \\wsl\$\Ubuntu\, ovvero impartendo il seguente comando da Powershell:

```
dir \\wsl$\Ubuntu
```

1.2 Macchina virtuale

Innanzitutto, bisogna essere certi che sul vostro pc sia abilitata la tecnologia di virtualizzazione. Non su tutti i pc è abilitata di default. Per abilitarla bisogna accedere al bios del vostro computer. Ecco come:

- Per prima cosa dovete spegnere il computer.
- Una volta spento bisogna premere il tasto accensione e subito dopo premere ripetutamente il tasto per accedere alle impostazioni di bios. Questo tasto è diverso per ogni produttore di pc. Ad esempio io ho un pc "HP" e sul mio computer è il tasto "F10". Per sapere qual'è il vostro basta fare una breve ricerca sul web del tipo "Accedere al bios su pc [marca]". Il bios ha l'aspetto in fig 7
- Ora non tutti i bios hanno la stessa organizzazione e le stesse schede. Bene o male dovrete cercare una scheda di "configurazione" o di "impostazioni" (settings in inglese) o qualcosa del genere. Spostatevi in quella scheda usando le frecce e cercate l'opzione "virtualization technology" e assicuratevi che sia abilitata (vedi fig. 8). Se non trovate l'opzione potete di nuovo cercare su internet dove trovarla. Se ancora non la trovate ignorate questo passaggio e provate ad andare avanti, magari sul vostro pc è abilitata di default.
- Fatto ciò, salvate le modifiche e uscite (di solito c'è una legenda che vi indica come fare).

Ora possiamo passare a scaricare un programma per creare macchine virtuali. Sono certo che ne esistano diversi, ma uno molto popolare e molto facile da usare è "Oracle VM

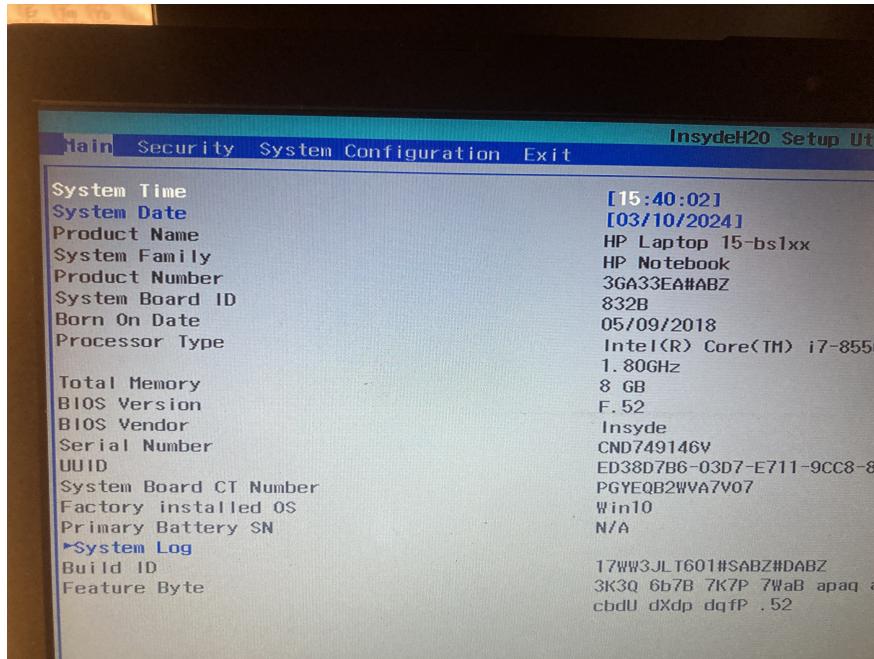


Figure 7

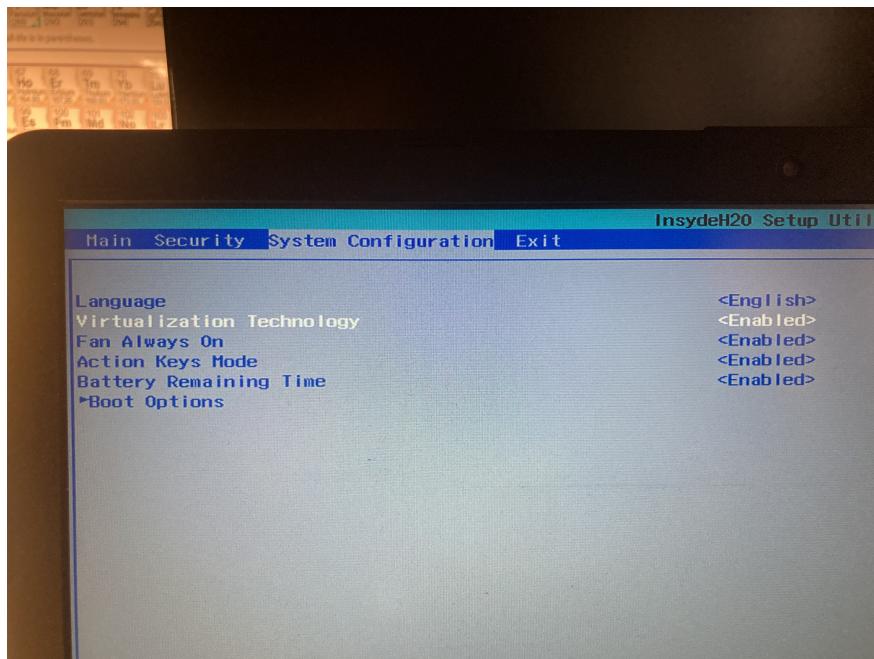


Figure 8

"Virtualbox". Per scaricarlo basta andare cercare su Google "Oracle VM Virtualbox" (o anche solo "VM Virtualbox") (vedi fig. 9)

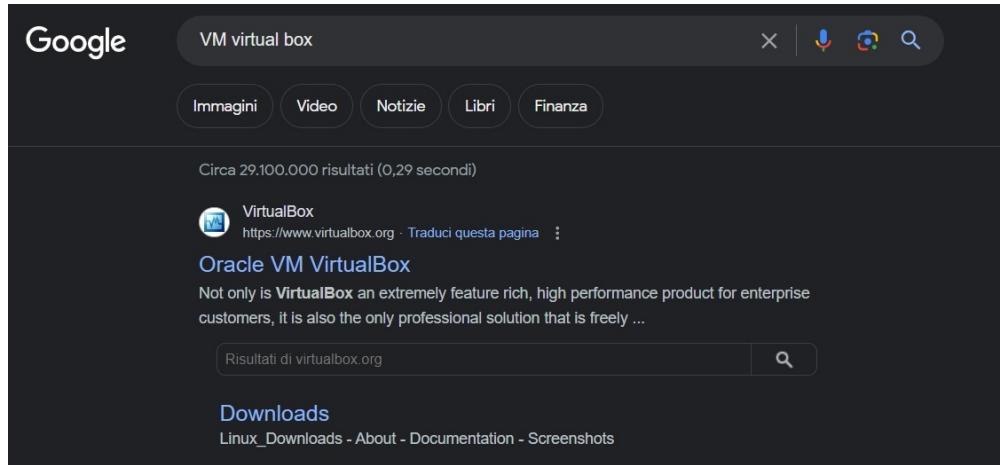


Figure 9: Il primo risultato è quello che ci interessa

Cliccate quindi su "Downloads" del primo risultato. Quindi cliccate su "Windows host" (vedi fig. 10)

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

VirtualBox 7.0.14 platform packages

- [Windows hosts](#)
- [macOS / Intel hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

Figure 10: cliccare su windows host

A questo punto, nella cartella "Download" del vostro computer dovreste avere il file come in figura 11

Fate doppio click per aprirlo e proseguite con l'installazione guidata. Qui non ho messo le immagini ma dovete solo seguire le istruzioni del menu e cliccare avanti. (Se vi appare l'avviso di Windows che il programma vuole apportare modifiche al sistema non vi impanicate, è normale, dategli il permesso. Se invece vi chiede i codici della carta di credito, quello è un

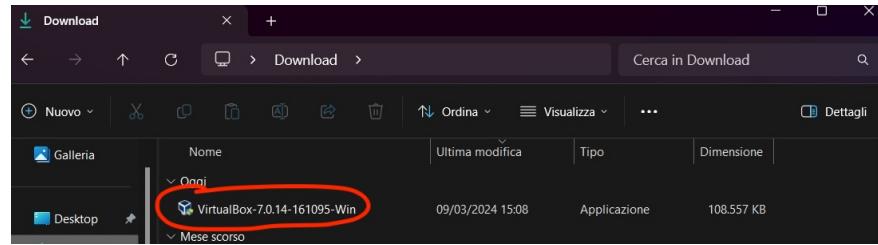


Figure 11: congratulazioni hai scaricato il programma giusto!

po' meno normale, magari non fatelo).

Giunti al termine dell'installazione sul desktop vi trovate questa bella iconcina 12

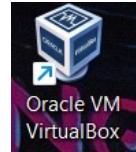


Figure 12: Contrariamente a come si potrebbe pensare dal nome, non predice il futuro

Ora fate doppio clic sull'icona per aprire il programma. Vi accoglierà questa finestra 13

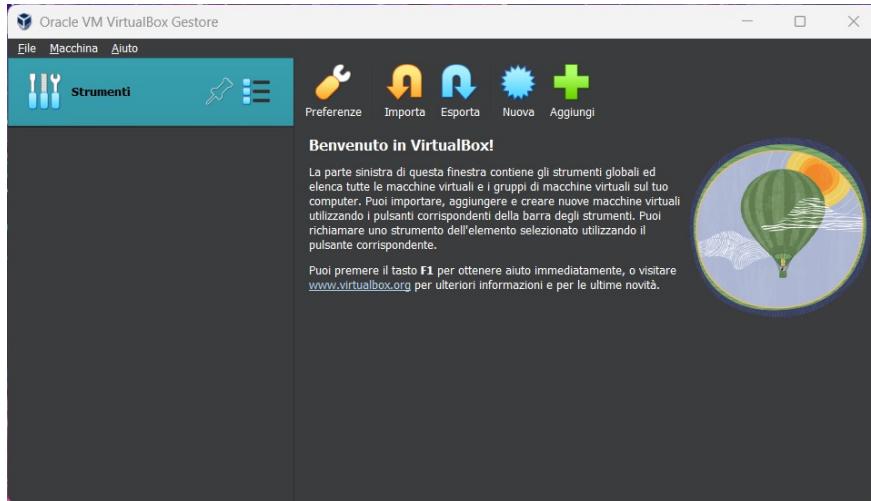


Figure 13

Prima di andare avanti facciamo una cosa che ci sarà utile più avanti. Da questa schermata, cliccate in alto a sinistra sul menù "file", poi su "preferenze", andate sulla scheda "Immissione" e sulla sottoscheda "macchina virtuale". Qui cercate la voce "combinazione tasto host" e verificate quale tasto della tastiera è impostato come host (vedi fig. 14). Assicuratevi di sapere quale sia e se non lo sapete potete cambiarlo semplicemente cliccandoci sopra, selezionate la piccola icona a forma di gomma da cancellare arancione che appare sulla destra e premete il nuovo pulsante che volete assegnare come host. Assicuratevi che sia un tasto di cui potete fare a meno nella tastiera, io ad esempio ho scelto il tasto "maiusc" di destra. Quindi cliccate su ok.

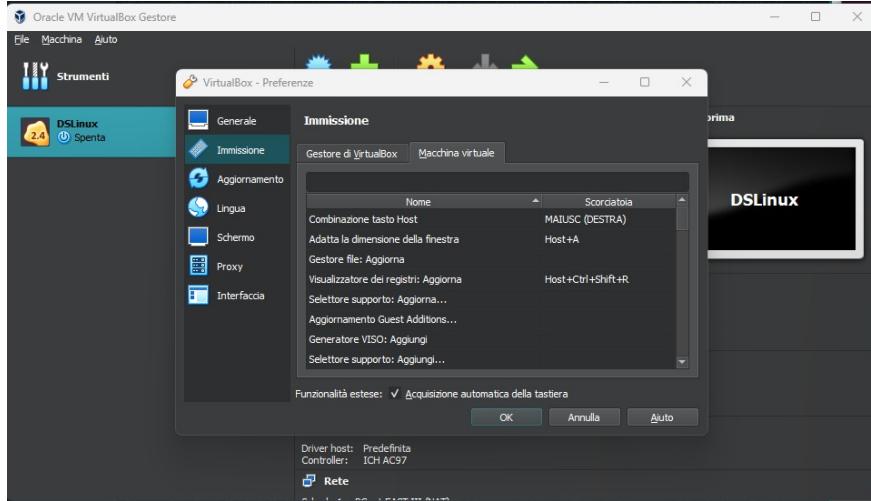


Figure 14: Sulla tastiera di solito ci sono due tasti maiusc, uno a destre e uno a sinistra. Quello a destro non lo usa mai nessuno, probabilmente non lo avete mai usato, questo sarà il giorno più glorioso della sua vita

Bene, il software per la virtualizzazione, dobbiamo passare alla configurazione della macchina virtuale vera e propria. Le possibilità sono molte ed illustrate nelle prossime sezioni. La via più semplice è quella di installare la macchina virtuale di laboratorio, che riprodurrà fedelmente i computer presenti in laboratorio. Alternativamente sarà possibile scegliere un sistema operativo di vostro gusto, come Ubuntu o altri software specifici per dispositivi *poco performanti*.

1.2.1 La via più semplice

Se avete letto fino a questo punto possiamo finalmente illustrarvi il modo più semplice di installare una *Virtual Machine* sul vostro computer. Si tratta infatti di installare una macchina virtuale *già fatta*, senza bisogno di configurarla da zero. Potete scaricare il file a questo link, salvatelo dove saprete ritrovarlo! Una volta ultimato il download, aprite Oracle VM. A differenza di prima, selezionate il simbolo del '+'. Nel menù che si apre, cercate il file appena scaricato e selezionatelo. Cliccate su continua. Nella casella RAM che si aprirà potete aumentare (ma non diminuirlo!) il valore della RAM impostandolo ad un valore pari ad un quarto della ram di cui disponete nel vostro computer. Ad es. se il vostro computer ha

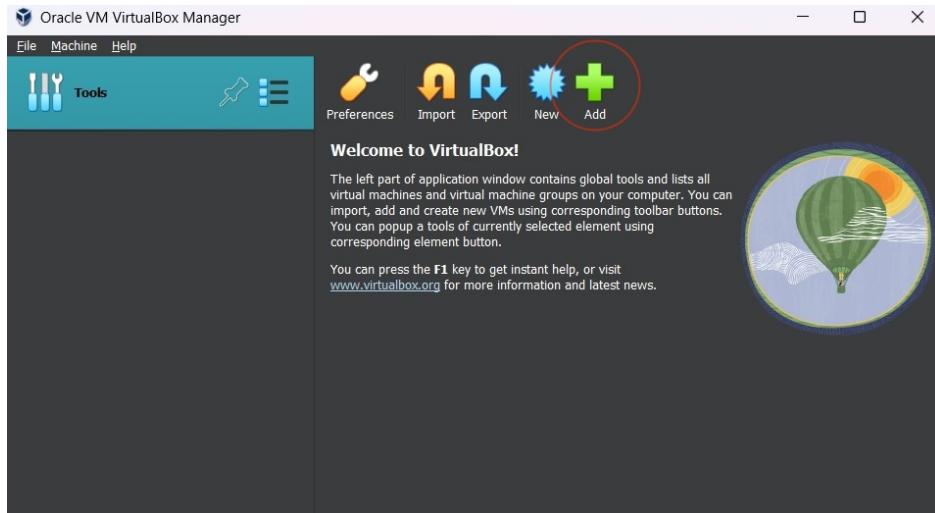


Figure 15: Per aggiungere una macchina virtuale senza configurarla *from scratch*, selezionate il simbolo del '+'.

8 Gb potete aumentare il valore a 2048 Mb, se ha 16 Gb potete inserire 4096 Mb. Cliccate su 'importa' e potrete avviare la vostra macchina virtuale. Questa macchina virtuale è una copia quasi esatta dei computer disponibili in laboratorio, pertanto potrete accedere con lo user 'studente' e la password 'informatica'².

1.3 Macchina Virtuale *Custom*

Qualora vogliate utilizzare una macchina virtuale con un software più completo ed aggiornato di quelle di laboratorio, è possibile procedere con un installazione personalizzata. Innanzitutto dobbiamo scegliere quindi quale distro utilizzare. Per scegliere la distro abbiamo bisogno di capire che tipo di computer hai. Nella stra grande maggioranza dei casi Ubuntu è una distribuzione che va bene. Ma come capirlo? Tutto dipende se avete un computer abbastanza recente e soprattutto performante. Le caratteristiche consigliate sono di avere un computer con almeno 8GB di ram (forse anche 4GB vanno bene, ma più avanti proporrò una soluzione più adatta), una CPU con almeno 4 core e almeno 25 GB di spazio di archiviazione libero. Una caratteristica fondamentale invece è che il vostro computer abbia un'architettura x64

²Qualora ne abbiate bisogno, la password di root è 'labcalc'.

(o "amd64", sono due diciture equivalenti), caratteristica che ormai anno tutti i computer da almeno 10-15 anni, tranne poche eccezioni. So che cosa ti stai chiedendo:

è presto detto. Dalla schermata di fig. 13 clicca su "nuovo" con l'icona a forma di nuvola appuntita. Nella finestra di dialogo che appare mettete un nome a caso e poi andate avanti (vedi fig. 16).

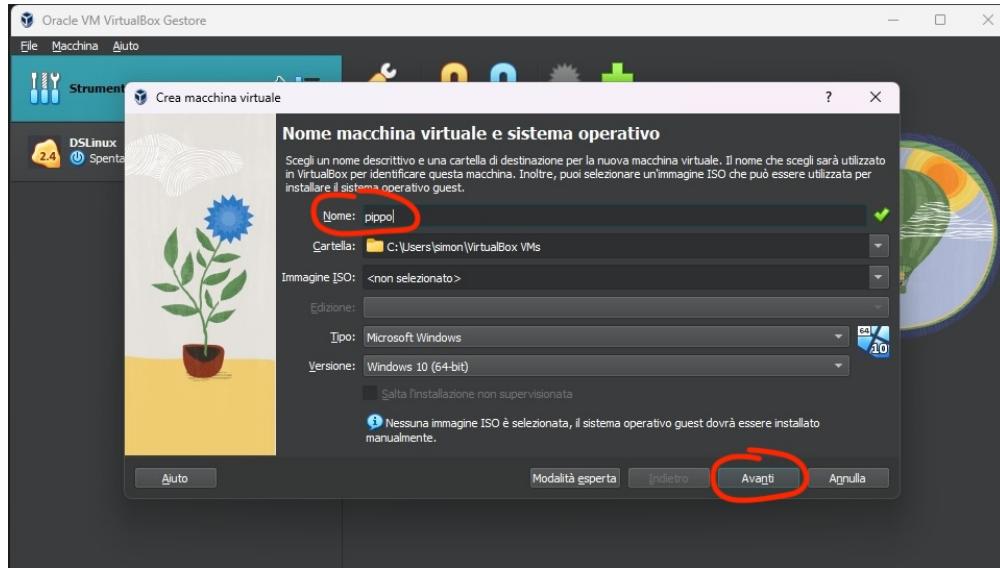


Figure 16: Pippo è il nome fittizio più in voga tra i fisici italiani

nella schermata successiva potrete vedere quanti processori e quanta ram possiede il vostro PC (vedi fig. 17).

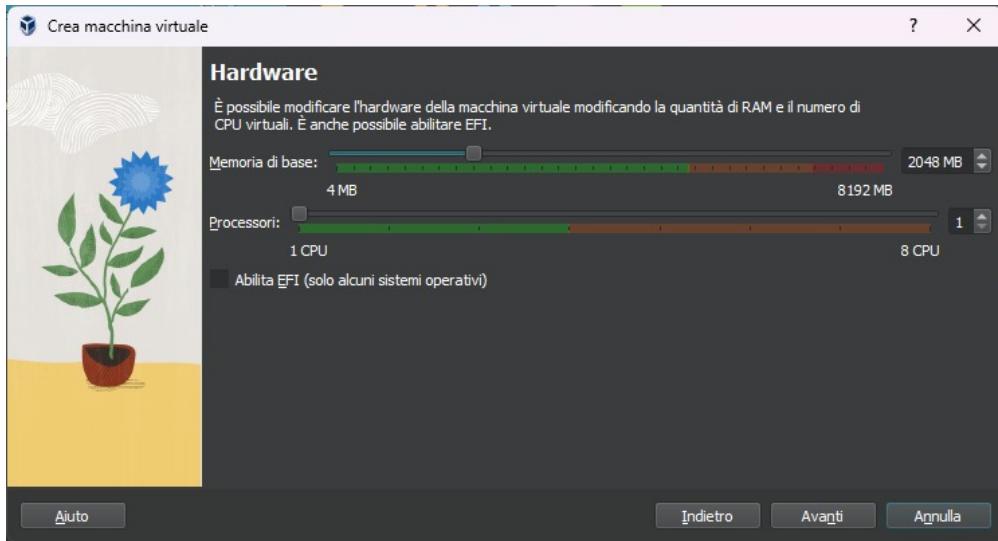


Figure 17: questo è un pc del 2018, ci sono buone possibilità che il vostro sia così o anche migliore

Per quanto riguarda lo spazio disponibile, basta che aprate una cartella qualunque. Nella colonna laterale cliccate su "questo PC", a questo punto vi uscirà l'elenco dei dischi presenti nel vostro PC (in un portatile di solito ce n'è uno solo), e sotto ciascuno c'è scritto quanto spazio c'è disponibile. (fig.18)

Per sapere se il vostro computer è a x64, infine, basta andare nelle impostazioni di Windows, nella barra laterale selezionare "sistema" e scorrere in fondo fino al menù "informazioni di sistema". Da qui potete leggere che tipo di sistema avete. (fig 19)

Se il vostro PC non soddisfa una qualsiasi di queste caratteristiche non temete. Nei capitoli successivi farò in modo di trovare una distribuzione per venirvi incontro. Fatte queste verifiche iniziamo con l'installazione di Ubuntu.

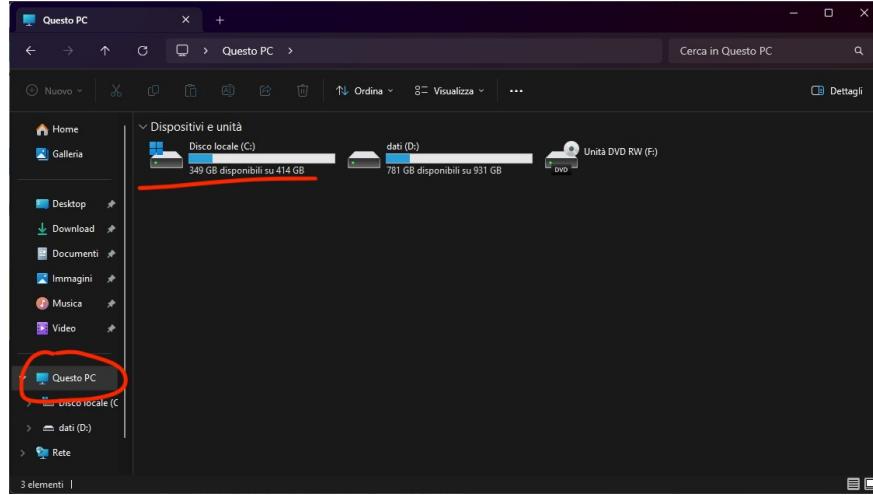


Figure 18: quello con il simbolo di Windows è il disco su cui è installato Windows. Eventuali altri dischi di solito sono destinati all'archiviazione dei dati.

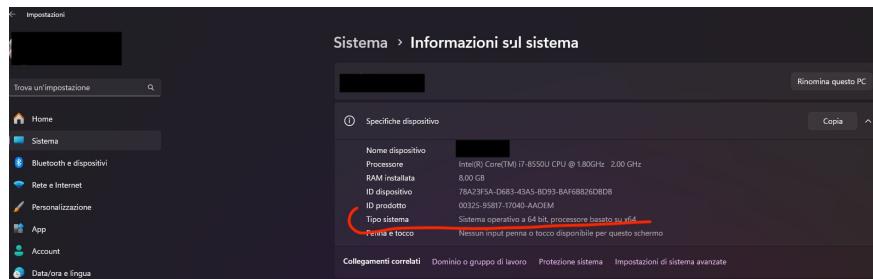


Figure 19

1.3.1 Ubuntu

Per prima cosa bisogna scaricarlo. Per farlo basta cercare Ubuntu su google e cliccate sulla voce "download" del primo risultato. (fig. 20)

Selezionate la versione da scaricare poi premete su avvia il download. Di solito è già selezionata di default la versione più recente per i sistemi desktop (che si distingue dalla versione server, che non è adatta a noi). Potete comunque scegliere dal menù a tendina tra la versione "standard" e quella "LTS" che sta per "long term support". Non ci sono grandi differenze di funzionalità tra le due. La versione standard viene aggiornata per 6 mesi e poi viene sostituita da una nuova versione del sistema operativo. Mentre la LTS viene portata

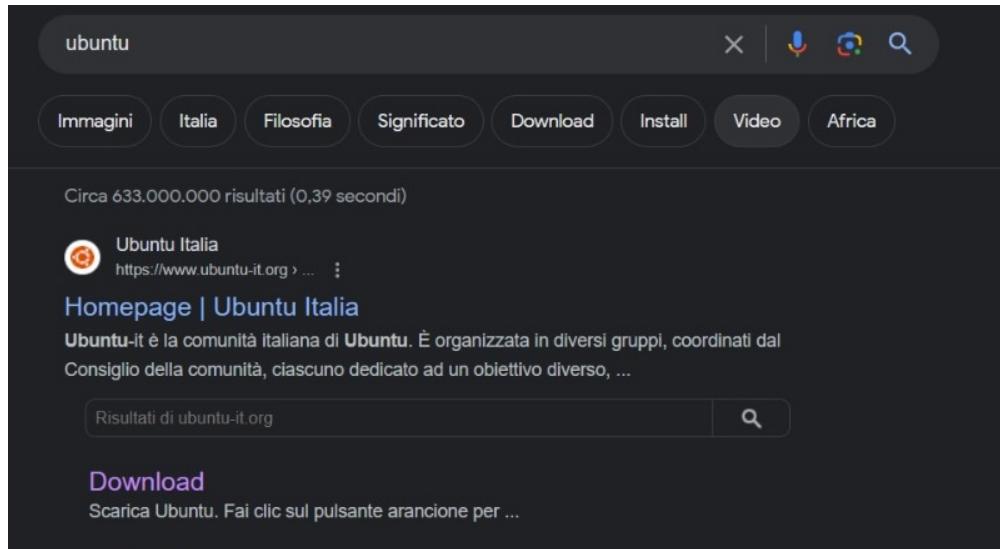


Figure 20: Sono l'unico che ha sempre notato una sospetta somiglianza tra il logo di ubuntu e la foto del buco nero al centro della via lattea?

avanti per 5 anni prima di essere sostituita. Dipende quindi da quanto vi va di essere sempre aggiornati a patto di dover fare l'upgrade del sistema operativo ogni 6 mesi, oppure se preferiate avere un sistema più stabile nel tempo.

Quale che sia la vostra scelta, ci ritroverete nella cartella download del vostro pc un file "immagine", cioè il cui nome termina con ".iso".

Torniamo ora alla schermata di fig. 16, ora clicchiamo ancora su "nuovo", questa volta per creare davvero una macchina virtuale.

Nella finestra di dialogo date un nome alla macchina virtuale. Scegliete come nome semplicemente Ubuntu, così il sistema riconoscerà in automatico alcune impostazioni per la macchina virtuale. Nello stesso menu ora alla voce "immagine iso" cliccate sulla freccetta per aprire il menù a tendina e selezionate "altro" per aprire una finestra di esplorazione dei file. (fig. 21)

Quindi selezionate il file di Ubuntu che abbiamo scaricato poc'anzi e cliccate su apri. (fig. 22)

Fatto ciò andate avanti. Nella prossima schermata dovete solo scegliere un nome utente e la password. Io sceglierò come nome "studente" e come password "password". Nella schermata successiva dovete scegliere quanta RAM e processori destinare alla macchina

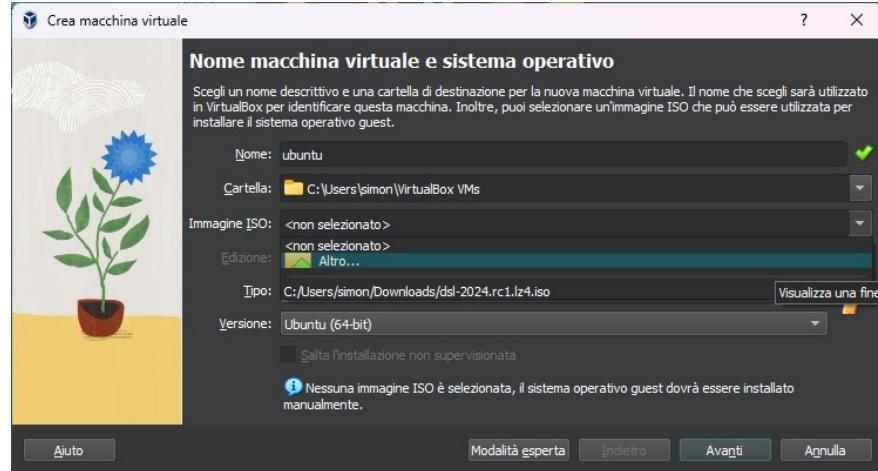


Figure 21

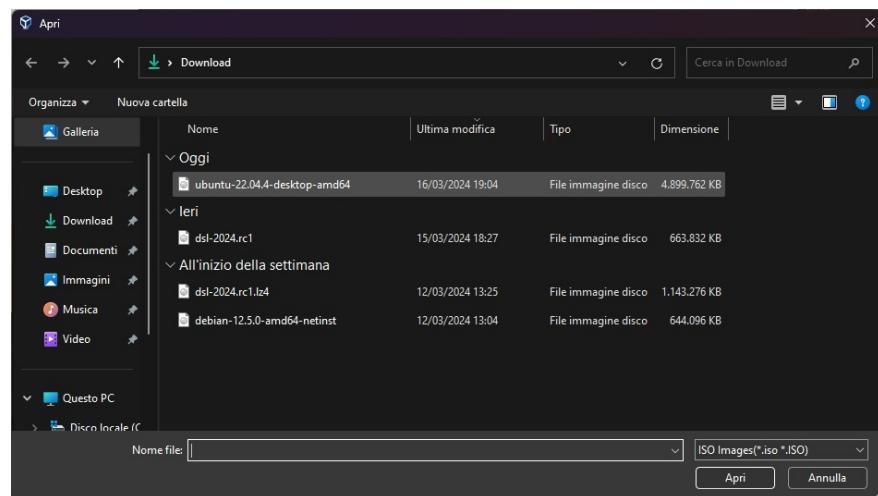


Figure 22

virtuale. Il mio consiglio è di mettere un minimo di 4GB di RAM e la metà dei processori totali. Infine mettete la spunta alla casellina "abilita EFI" ed andate avanti. (fig. 23)

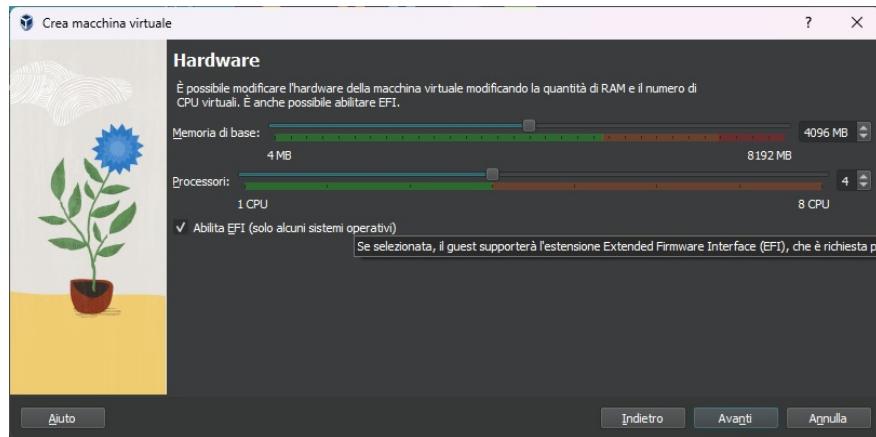


Figure 23: La RAM è richiesta in MB. 1GB=1024MB. Si lo so, ma lo hanno inventato degli ingegneri che volete farci?

Nella prossima schermata vi si chiederà quanto spazio di memoria riservare per la macchina. Vi consiglio 25GB come minimo. Se potete anche 50GB. Andate ancora avanti e cliccate su "fine". Bene ora abbiamo impostato i parametri della macchina virtuale, che quindi si avvierà per iniziare il processo di installazione.

Potrebbe apparirvi un errore che dice "impossibile avviare la macchina virtuale...". In tal caso aprite il menù a tendina e selezionate di nuovo l'immagine disco di Ubuntu, che ora dovrebbe anche suggerirvi tra i recenti. (fig. 24)

Quindi clicca su "monta e riprova l'avvio". Ora dovrebbe partire il programma di installazione. Fate invio su "try or install Ubuntu".

A questo punto è praticamente fatta. L'unica cosa che dovrete fare è seguire le istruzioni a schermo e completare l'installazione. Ubuntu, in sé, ha un sistema di installazione molto facile e intuitivo, per questo vi consiglio di provare a installarlo direttamente in dual boot. Comunque, da qui in avanti, scegliete la lingua e il layout della tastiera quando vi viene richiesto e per il resto lasciate tutte le impostazioni come sono di default. Finita l'installazione la vostra macchina è pronta.

Pro tip: Ricordate il tasto host che abbiamo impostato prima? Se ora premete il tast

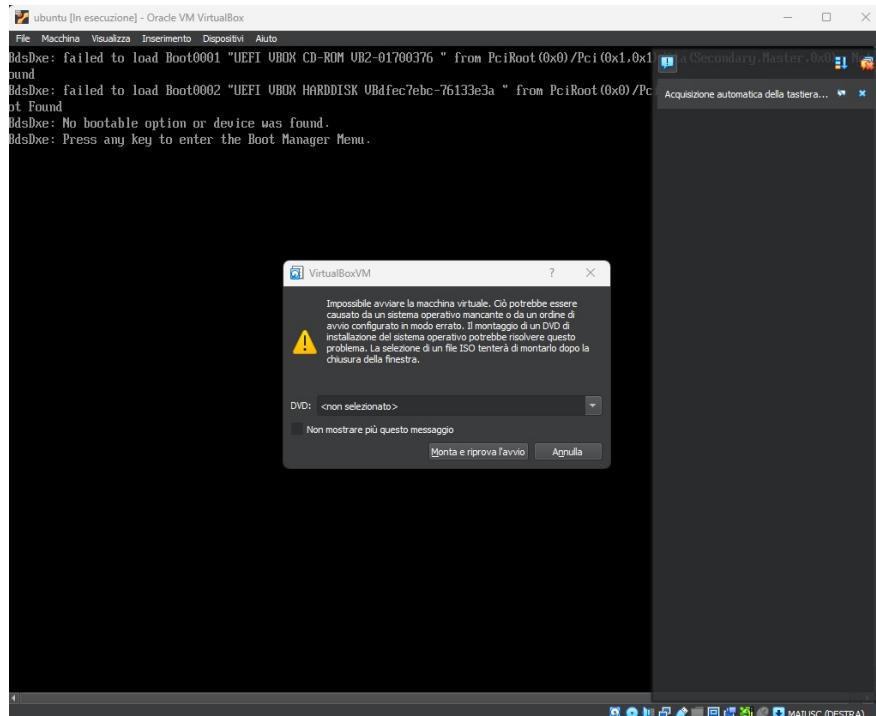


Figure 24

host+f la macchina virtuale passerà alla visualizzazione schermo intero. Per tornare alla visualizzazione in finestra basta premere host+c.

Nella visualizzazione a schermo intero la finestra non occuperà tutto lo spazio. per fare in modo che sia così dovete semplicemente cambiare la risoluzione dello schermo dalle impostazioni di ubuntu. (dopo che si sarà installato ovviamente).

1.3.2 Ho un computer troppo datato! Che cosa faccio?

Beh come primo suggerimento ti dico di valutare di optare per il dual boot. Ubuntu infatti non è un OS (che sta per "operative system") molto dispendioso, al contrario di Windows. Perciò se il vostro computer va bene con Windows va sicuramente meglio con Ubuntu.

Se proprio questa idea non vi piace, allora andiamo ad valutare altre possibilità, sostanzialmente dobbiamo scegliere delle distribuzioni (distro) diverse. Se avete verificato di avere un sistema x64 allora quali alternative puoi provare:

LUbuntu: che sta per Light-Ubuntu. Si tratta cioè di Ubuntu sotto mentite spoglie. Che cosa lo rende più leggero? La risposta facile è che utilizza un diverso ambiente grafico. Dovete sapere che tutte le finestre e in generale tutta la grafica del vostro computer è gestito da un programma che non è in sé legato al sistema operativo. In base a quale programma si sceglie per gestire l'interfaccia grafica si potrebbe appesantire o alleggerire il sistema operativo. Sono sicuro che ci sono anche altre ottimizzazioni che rendono Lubuntu più leggero, ma francamente non le conosco. In ogni caso se riscontrate dei forti rallentamenti con Ubuntu, usando Lubuntu dovreste migliorare considerevolmente la fluidità della macchina.

Per installare Lubuntu potete seguire la stessa identica procedura seguita per Ubuntu. Ovviamente l'unica accortezza è di scaricare il file immagine di Lubuntu dal sito ufficiale "<https://lubuntu.me/downloads/>" (se non sai di cosa sto parlando, leggi la guida per Ubuntu, li lo spiego). Per quanto riguarda le risorse da destinare al sistema a Lubuntu dovrebbero bastare 1 GB di RAM e forse ve la potreste cavare con un solo core (2 sarebbe meglio). A parte questo i passaggi sono gli stessi che per Ubuntu.

Debian: se Lubuntu era Ubuntu sotto mentite spoglie, possiamo anche dire che Ubuntu in realtà sia Debian sotto mentite spoglie. Ubuntu infatti deriva da Debian e ne costituisce una diramazione che nel tempo ha assunto vita propria. Ubuntu ha aggiunto molti abbellimenti grafici e alcune funzionalità al sistema Debian che in sé risulta piuttosto spartano. Proprio questa semplicità però lo rendono particolarmente leggero. Il processo di installazione è simile a quello di Ubuntu. Di nuovo dovete scaricare il file immagine del sistema dal sito ufficiale "<https://www.debian.org/download>" cliccando sul file ".iso". vi segnalo alcune accortezze:

- ho verificato che l'installazione di Debian potrebbe non funzionare abilitando l' "EFI".
Se così fosse semplicemente cancellate la macchina virtuale che non si avvia e quando arrivate alla finestra in figura 23 **NON** spuntate la casella. Per il resto l'installazione proseguirà in modo abbastanza intuitivo, sebbene molto più spartano di Ubuntu.
- Non potrete selezionare la lingua italiana, quindi inizialmente il sistema sarà in inglese. Potrete però selezionare il layout italiano della tastiera. Credetemi abbiate accortezza di farlo perché poi impazzite a trovare i tasti giusti (sebbene potrete comunque modificarlo in seguito).
- Infine, in Debian potreste avere dei problemi ad installare software da terminale, nel senso che il vostro utente non godrà dei privilegi necessari. Quando proverete a dare il comando "`sudo apt install /cose/`" (le parentesi quadre qui indicano un valore variabile, sostituire il nome del programma che volete installare senza mettere le parentesi), dopo aver inserito la password, vi darà un errore dicendo che l'utente non è inserito nel file "sudoerr". Per ovviare basta cambiare utente nel terminale e loggare come utente "root", cioè l'utente con il massimo grado di autorità nel sistema. Per farlo basta che da terminale digitiate "`su root`" dove "su" è un comando che significa "sobstitute user". Dopo aver inserito la password potrete scaricare i programmi. Per tornare al vostro utente potete o chiudere il terminale e aprirne uno nuovo, oppure digitare "`su /nomeutente/`".

Infine vi propongo una distro molto recente (derivata da una distro chiamata "antix"

a sua volta derivata da Debian) che è la più leggera ma anche funzionale che ho trovato. Questa distro va bene anche per gli utenti **NON** hanno un sistema x64. Se avete un computer molto molto vecchio, su cui probabilmente c'è ancora Windows 7, oltre al fatto che è fortemente consigliabile cambiare computer, potreste avere un sistema con architettura "i386" detta anche "a 32 bit". Francamente se avete un computer a 32 bit non sono sicuro che riusciate neppure ad installare il programma di Virtual box. In quel caso vi consiglio di installare Linux come unico sistema operativo (seguendo in passaggi che darò più avanti per il dual boot). Comunque, questa distro si chiama DSLinux, che sta per "Damn Small Linux". Potete scaricare DSLinux dal sito ufficiale "<https://www.damnsmalllinux.org/2024-download.html>". Anche in questo caso, come per debian, potreste dover disabilitare l'opzione dell'EFI.

1.3.3 Architettura ARM

I sistemi ARM sono il futuro dei personal computer. Questa architettura è già stata adottata su tutti i pc Apple dal 2020 in poi. Per i pc Windows invece non è ancora molto diffusa, ma potreste avere tra le mani uno dei pochi pc windows dotati di un processore con architettura ARM. In questo caso, non sono certo che Virtual box funzioni sul vostro pc. In caso affermativo, non dovreste avere problemi ad eseguire macchine virtuali anche per architetture x64, perché il programma penserà a fingere di avere l'architettura giusta. Se così non fosse potete provare con la versione ARM di Debian "<https://www.debian.org/ports/arm/index.it.html>". Se invece virtual box non funziona potreste dover installare necessariamente in dual boot. In tal caso vi consiglio di cercare dei tutorial su Youtube. La procedura dovrebbe infatti essere la medesima che per qualsiasi altro pc, ma non potendolo verificare non ve lo so garantire.

1.3.4 Troubleshooting per VM

- **Non funziona Internet all'interno della macchina virtuale:**

Aprire **Impostazioni>Network**. Cambiare la voce **NAT** con **Bridged o Bridged Adapter** e alla voce **Avanzate** sostituire **Nega** con **Permetti Tutto**. (vedi fig 25)

- **Cambiare le dimensioni del testo su schermo:**

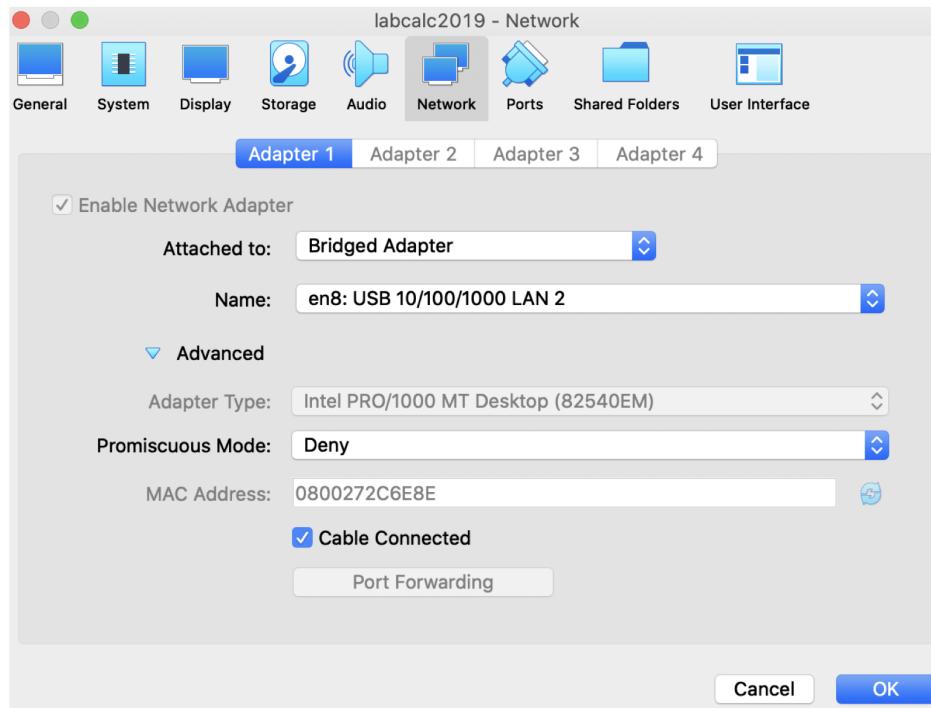


Figure 25: Impostazioni Network.

Aprire Impostazioni>Schermo. Cambiare la voce Fattore di scala a piacimento.

- **end Kernel panic - attempted to kill idle task:**

Per risolvere questo messaggio di errore dobbiamo installare la versione precedente della macchina virtuale di laboratorio, disponibile a questo link. Installatela seguendo le istruzioni della sezione 1.2.1. Andate su Impostazioni>Sistema>Processore. Qua cambiate il numero di processori da 1 (che è il default) a 2/3. Cliccate OK e avviate la macchina virtuale.

1.4 Dual boot

ATTENZIONE: Per fare questo procedimento hai bisogno di una chiave USB da almeno 8GB di memoria. Questa chiavetta verrà formattata (cioè saranno cancellati tutti i dati che ci sono), quindi assicurati che non ci siano dati importanti

ATTENZIONE 2: Assicuratevi di sapere quanto spazio libero avete sul vostro disco. In futuro vi sarà chiesto quanto spazio destinare a Ubuntu (o qualsiasi distro avete scelto). Se selezionate più spazio di quello libero disponibile potrete perdere dei dati.

Anche in questo caso bisogna scegliere quale distribuzione usare. Questo dipenderà essenzialmente dall'architettura del vostro computer. Per conoscere quale architettura ha il vostro pc (quasi certamente x64), potete seguire il procedimento che ho spiegato nel paragrafo della macchina virtuale.

1.4.1 Architettura x64

Diciamo che qui la questione è molto più semplice poiché Ubuntu già di suo ha bisogno di poche risorse e andrà bene per il 99% di voi. Coloro che avessero un pc più datatato potranno facilmente virare su LUbuntu (per avere chiarimenti su Ubuntu/Lubuntu fate riferimento ai rispettivi paragrafi nella sezione "macchina virtuale").

In questo tutorial vediamo come installare Ubuntu in dual boot (per Lubuntu il procedimento è comunque analogo, essendo sostanzialmente la stessa cosa). Per prima cosa scarichiamo il file immagine di Ubuntu (vedi paragrafo Ubuntu in macchine virtuali)

Ora abbiamo bisogno di scaricare un programma per creare una chiave di avvio, cioè per installare sulla chiavetta il programma di installazione di Ubuntu (sembra confusionario, ma è più semplice di quel che sembra).

Di questi programmi ne esistono di certo molti. Uno molto popolare si chiama "Rufus", scaricabile dal sito ufficiale "<https://rufus.ie/it/>" (fig. 26)

Bene, ora nella cartella download dovreste avere il programma rufus e il file immagine

Ultime uscite:					
Link	Tipo	Piattaforma	Dimensione	Data	
rufus-4.4.exe	Standard	Windows x64	1.4 MB	2024.01.17	
rufus-4.4p.exe	Portatile	Windows x64	1.4 MB	2024.01.17	
rufus-4.4_x86.exe	Standard	Windows x86	1.4 MB	2024.01.17	
rufus-4.4_arm64.exe	Standard	Windows ARM64	4.6 MB	2024.01.17	

Figure 26: scaricare la versione standard per x64.

di Ubuntu (di nuovo, fate riferimento alla sezione della macchina virtuale per info sul file immagine).

A questo punto inserite la chiavetta nel pc e avviate rufus. In figura 27 è mostrata la schermata di rufus e sono indicati i passaggi da effettuare per creare la chiavetta avviabile.

Una volta che rufus ha completato l'operazione, abbiamo pronta la chiavetta avviabile. Quello che dovete fare ora è spegnere il computer. Quando si sarà spento, premete di nuovo il tasto accensione per avviare il pc ma subito dopo dovete premere ripetutamente il tasto per entrare nel boot manager. Questo tasto non è definito univocamente, ma dipende dal produttore del vostro pc. Io, ad esempio, possiedo un pc di HP e il tasto per il boot manager è F9. Potete trovare quello del vostro pc semplicemente cercando su internet.

Che cos'è il boot manager? Quando il vostro computer si avvia la prima cosa che fa è far partire il bios, che è un programma integrato nel vostro pc che consente di eseguire le operazioni più basiliari per far funzionare il processore e controllare gli input e gli output. Il bios come prima cosa ricerca tra tutti i dispositivi di input in quale siano presenti programmi di avvio (cioè di boot). Il bios (se non diversamente istruito) ha una priorità per quale programma di avvio scegliere se ce n'è più di uno. Di solito l'unico presente è quello che sta nel vostro disco di sistema ed è il programma di avvio di Windows. Ora però c'è un programma di avvio disponibile anche nella chiavetta. Tuttavia il bios darà comunque priorità a Windows, dobbiamo perciò dirgli manualmente di usare invece la chiavetta.

entrati nel boot manager vi troverete dinanzi una schermata simile a quella in figura 28, da questa schermata selezionate la vostra chiavetta (di solito segnalata dal fatto che sarà l'unico dispositivo USB). Partirà quindi il programma di installazione di Ubuntu. Vedrete che da qui sarà tutto facile. Nella schermata successiva date invio su "try or install ubuntu". Una volta avviato, selezionate la lingua italiana e cliccate su "Installa Ubuntu" (vedi fig. 29)

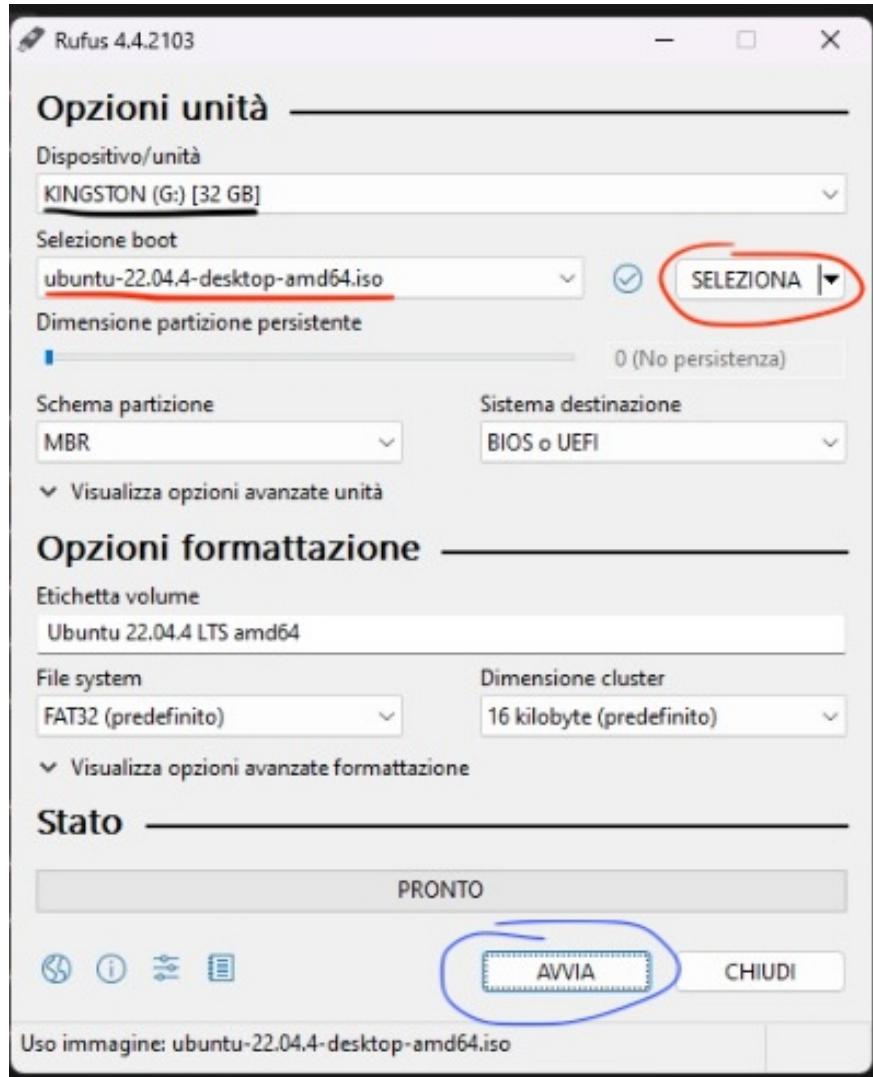


Figure 27: Cerchiato in nero dovrebbe apparirvi in automatico il nome della chiavetta usb che avete inserito. Se così non fosse selezionatela manualmente dal menù a tendina. Cliccando su "selezione" in rosso dovrete andare a selezionare il file immagine di Ubuntu, che una volta selezionato correttamente vi apparirà come quello sottolineato in rosso. Lasciate tutte le altre impostazioni come sono e premete "avvia" cerchiato in blu. Quando ve lo chiede lasciate le impostazioni suggerite. Potrebbe chiedervi anche di installare dei componenti aggiuntivi, permetteteglielo. Aspettate che il programma finisca prima di rimuovere la chiavetta (in caso contrario potreste anche renderla inutilizzabile, fidatevi, ci ho provato (;-;)).

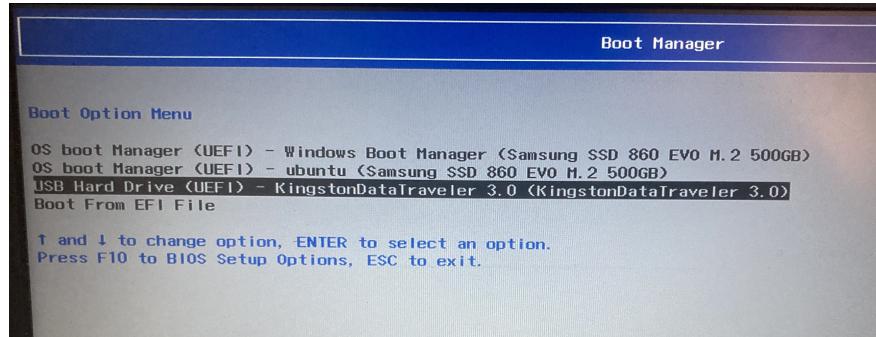


Figure 28: Spostatevi con le frecce sul dispositivo USB e premete invio

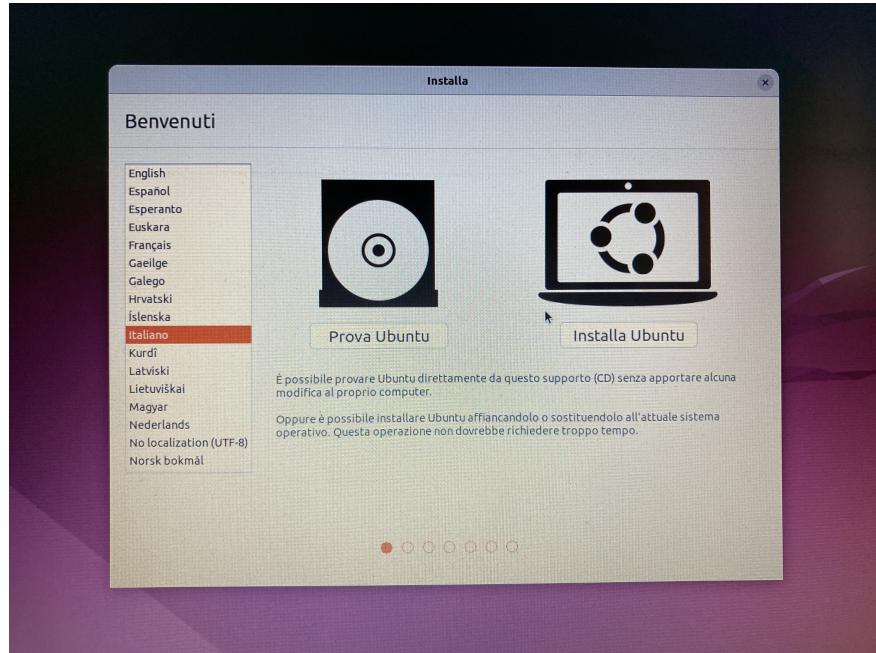


Figure 29

A questo punto potete scegliere se connettervi ad internet (scelta consigliata) oppure no. Connettendovi ad internet il programma scaricherà in automatico gli ultimi aggiornamenti durante l'installazione. Questo potrebbe però rendere più lenta l'installazione a seconda della velocità della vostra connessione. Se vi siete connessi ad internet, nella schermata successiva vi chiederà se fare un'installazione normale o minimale. Qui scegliete voi, se avete intenzione di provare ad usare Ubuntu come OS generico per la vita quotidiana allora scegliete un'installazione normale. Se non avete alcuna intenzione di usarlo fuor che per programmare e volere risparmiare spazio allora fate un'installazione minimale. Vi consiglio vivamente invece di mettere la spunta per scaricare gli aggiornamenti e per installare i driver di terze parti (senza i quali alcuni dispositivi del vostro pc potrebbero non funzionare adeguatamente, come schede video o Bluetooth).

Si giunge ora a una parte cruciale. Prestate **ATTENZIONE** perché fare errori a questo step vi potrebbe costare tutti i dati che avete salvati sul pc. La schermata che vi si presenta davanti è quella di figura 30.

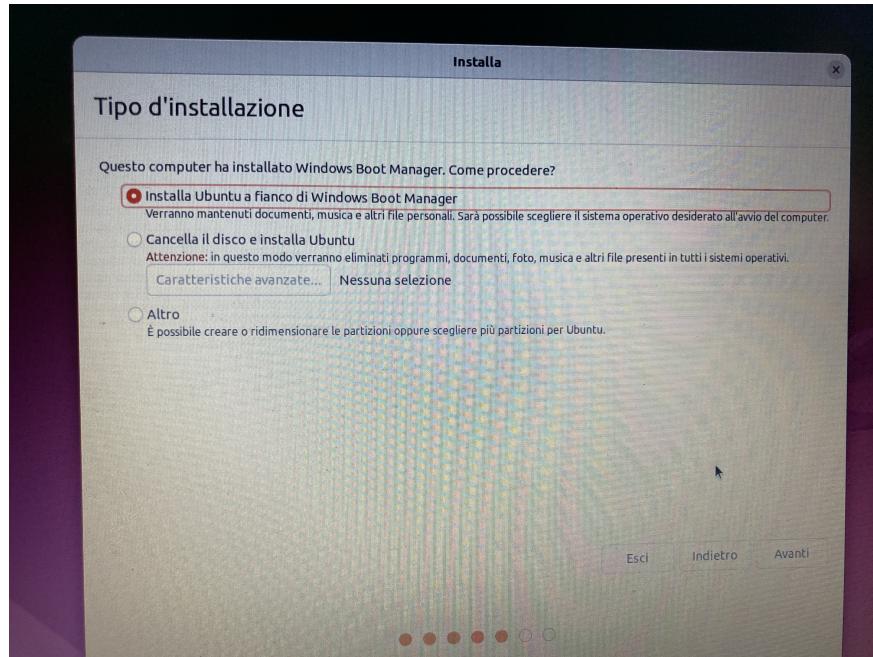


Figure 30: Leggi attentamente cosa selezioni.

Qui puoi dare due scelte:

1. Cancellare tutto il disco e installare **solo** ubuntu. **ATTENZIONE:** così facendo perderai tutti i tuoi dati, pensaci bene! Potresti optare per questa scelta se hai un pc un po' datato che da comunque problemi a funzionare su Windows. Oppure se scopri che Linux è proprio l'OS per te. In questo caso seleziona "cancella il disco e installa Ubuntu".
2. La seconda scelta, quella che interessa i più, è di installare Ubuntu a fianco di Windows. Come dice abbastanza intuitivamente la schermata, selezionerete quindi "Installa Ubuntu a fianco di Windows Boot Manager".

Supponendo che abbiate scelto per affiancare Ubuntu a Windows, nella schermata successiva vi chiederà su quale disco installare Ubuntu (se avete un portatile quasi certamente ne avete uno solo, in caso contrario scegliete lo stesso su cui c'è Windows) e quando spazio destinargli. Per regolare lo spazio si fa in modo molto semplice dalla schermata in figura 31 cliccando e trascinando la barra di divisione tra Ubuntu e il resto. Al contrario della macchina virtuale dove suggerivo almeno 25GB, in questo caso ve ne consiglio un minimo di 50, poiché potreste aver bisogno di più software da installare.

Ora potete fare clicca su installa e il gioco è fatto, dovete solo attendere che lui finisca (potrebbe volerci tra i 10 minuti a delle ore a seconda delle performance del pc e della connessione). **IMPORTANTE:** assicuratevi che il pc sia carico o attaccato alla corrente fino alla fine del processo.

Quando l'installazione è completata, premete su riavvia e ora potete (e dovete, ve lo chiederà lui stesso) rimuovere la chiavetta usb usata per l'installazione.

Eccetto che per questo primo riavvio, d'ora in avanti quando accenderete il pc, vi si chiederà di selezionare il sistema operativo con cui avviare il computer (scegliendo in automatico Ubuntu se non scegliete in 10 secondi).

Alcuni suggerimenti:

- Potreste notare che avviando Windows dopo aver avviato Ubuntu, l'orologio di Windows si trovi un'ora avanti e dovrete reimpostarlo. Per evitare questo bisogna semplicemente cambiare delle impostazioni dell'orologio di Ubuntu. Avviate quindi Ubuntu,

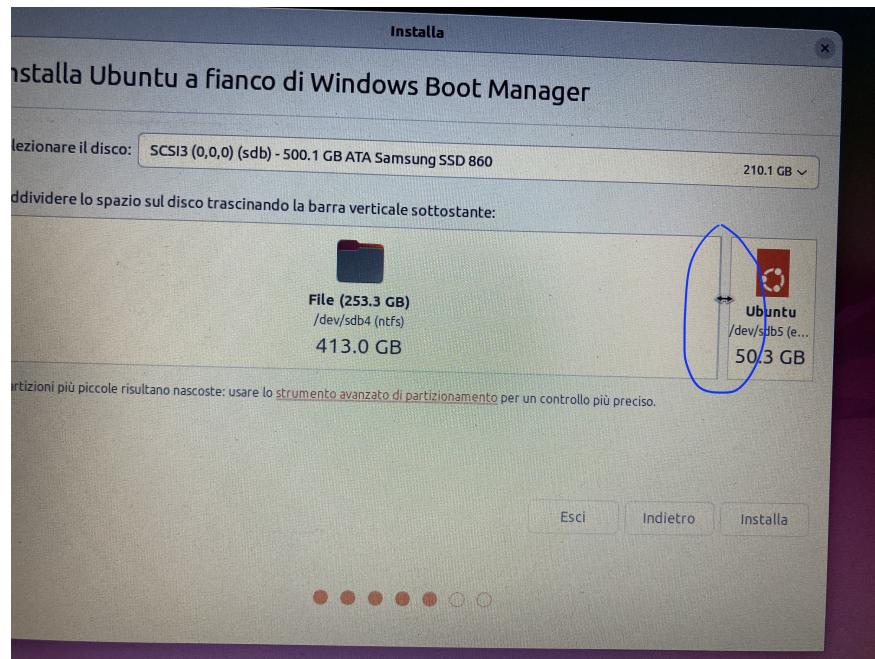


Figure 31: La barra cerchiata di blu è quella da spostare per regolare le dimensioni del disco da riservare a Ubuntu. Ovviamente siate certi di destinare uno spazio minore a quello che avevate libero su Windows.

aprite un terminale e digitate

```
gtimedatectl set-local-rtc 1 --adjust-system-clock
```

Questo dovrebbe risolvere il problema (forse non dal successivo avvio di windows ma dal secondo in poi si).

- Un altro suggerimento è che come me potreste trovare scomodo dover scegliere ogni volta il sistema operativo, soprattutto se siete soliti usare Windows, mentre Ubuntu solo per programmare. Nel modo che vi dirò potrete fare in modo che il pc scelga in automatico Windows a meno che non gli dicate espressamente di prendere Ubuntu. (Nota: questo metodo è testato sul bios del mio pc HP, dovrebbe funzionare per tutti ma non lo assicuro, in ogni caso non succede nulla di brutto, al più non riuscirete a fare questa cosa).

Spegnete il pc. Ora riaccendetelo e entrate nel bios. Per entrare nel bios bisogna premere ripetutamente l'apposito tasto che cambia in base al produttore di pc (per il mio HP è F10, per le altre marche basterà cercarlo su internet). Vi troverete quindi nel bios simile alla figura 7. Il bios anche può cambiare da produttore a produttore e da modello a modello. Ripeto i bios sono organizzati in maniera diversa, provate ad arrivare in fondo alla lettura per capire il senso di quello che faccio e probabilmente riuscirete a trovare l'opzione giusta anche nel vostro bios.

Scorrete con le frecce fino a un menù di impostazioni (nel mio caso è quello della figura 8). Qui cercate una voce "Boot option" e selezionatela (usando le frecce per spostarvi e invio per selezionarla). Aperta questa opzione vi si aprirà una nuova scheda, qui c'è una sezione "Boot order" in cui vengono elencati in ordine in quali posizioni il bios va a cercare dei programmi di avvio. Al primo posto dovrebbe esserci "OS boot manager" (vedi fig. 32). Andate sopra questa opzione e selezionatela. Vi si aprirà un menù in cui sono elencati gli OS booter disponibili. Dovreste ora averne due, quello di windows e quello di ubuntu (vedi fig. 33). I due sono nell'ordine di priorità dall'alto verso il basso. Volete mettere per primo quello di windows. Per farlo, nel mio caso dovevo spostarli in alto o in basso usando F5/F6. Nel vostro bios ci dovrebbe essere una legenda che vi

indica come fare.

Una volta terminato potrete uscire salvando le modifiche. Ora quando avviate il

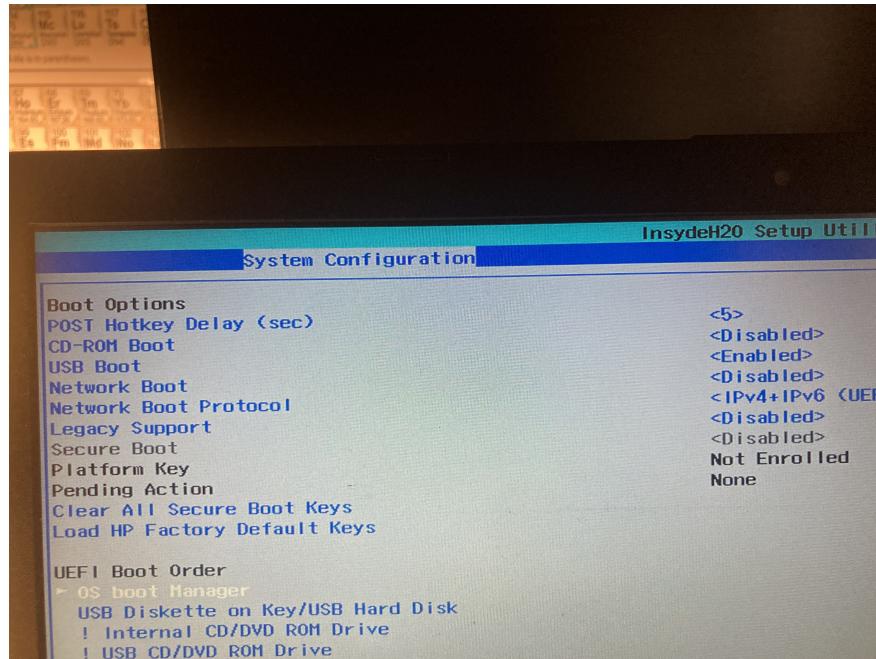


Figure 32



Figure 33

computer lui andrà direttamente su Windows senza chiedervi nulla. Se volete usare Ubuntu all'accensione del computer dovete premere il tasto del boot manager (in modo analogo a come avete fatto per selezionare la chiavetta USB come disco di avvio) e selezionare il boot loader di Ubuntu.

1.4.2 Altre architetture

Il procedimento è del tutto analogo a quello per i sistemi x64. Quello che dovete cambiare è: selezionare la versione di rufus adatta al vostro sistema (vedi istallazione per x64, fig. 26), se avete un sistema con architettura i386 dovete scaricare la versione a 32 bit, mentre se avete un processore ARM dovete scaricare la versione ARM; e come seconda cosa, trovare una distribuzione adatta alla vostra architettura. Per i sistemi i386 una distribuzione valida molto recente è DSLinux, scaricabile a "<https://damnsmalllinux.org/2024-download.html>". Per i sistemi ARM esiste una versione di ubuntu on ARM al link "<https://cdimage.ubuntu.com/jammy/daily-live/current/>" (Nota bene la differenza tra AMD64 e ARM, non sono la stessa cosa, AMD64 indica i sistemi x64, tu vuoi quella con la sigla ARM).

2 MacOS

Il sistema operativo MacOS è, come Linux, basato su un *kernel* UNIX. Senza addentrarci troppo nella storia del software, questo garantisce al sistema un incredibile compatibilità con tutti i più comuni tool usati per la programmazione.

Nel caso siate in possesso di un computer con sistema operativo MacOS, pertanto, vi trovate di fronte a due possibilità:

- Come illustrato per Windows, è possibile installare il software necessario alla virtualizzazione del sistema Linux. Questo ci permette di creare una 'scatola' contenente tutto ciò che utilizzerete nel corso, senza mischiarlo con le normali installazioni presenti nel vostro computer. Inoltre, virtualizzando Linux, il sistema si comporterà esattamente come i computer del laboratorio che dovrete utilizzare il giorno dell'esame.
- Utilizzare direttamente MacOS ai fini della produzione di codice. Data la sua struttura, infatti, sarà molto semplice configurare il sistema per svolgere tutte le attività necessarie al corso e anche ben oltre. L'unico svantaggio di questo approccio è che, per quanto molto simile, il sistema operativo MacOS non è *esattamente* uguale a Linux e pertanto quando vi troverete a usare un computer Linux (come il giorno dell'esame) alcune piccole cose potrebbero cambiare rispetto alla vostra abitudine.

Il nostro consiglio è quello di provare il secondo approccio. Nonostante porti con sé qualche difficoltà aggiuntiva, i vantaggi sono veramente moltissimi. Innanzitutto le performance dei vostri programmi saranno molto migliori, non essendoci necessità di usare potenza di calcolo per la virtualizzazione. Inoltre avrete a disposizione tutti i tool di cui avete bisogno direttamente nel file system del vostro computer, senza necessità di avviare programmi separati per accedervi. Con l'aumentare della complessità del codice e dei programmi che utilizzerete, la via della virtualizzazione diventerà sempre meno ottimale, per cui il consiglio è di imparare quanto prima come farne a meno. Da grandi poteri derivano però grandi responsabilità (e anche alcuni rischi): se non utilizzate la macchina virtuale, c'è il rischio che eventuali errori o malfunzionamenti possano compromettere l'intero sistema operativo principale. Al

contrario, utilizzando la virtualizzazione, qualsiasi problema rimarrà confinato alla macchina virtuale. In questo modo, nella peggiore delle ipotesi, sarà sufficiente reinstallare la macchina virtuale senza danneggiare il sistema principale. Per quanto con un utilizzo cauto del calcolatore il rischio di fare danni permanenti sia basso, è un rischio da tenere in considerazione. Per questo motivo, se volete essere molto cauti o se volete esercitarvi con una riproduzione fedele dei computer di laboratorio, la virtualizzazione è la vostra via.

A differenza che su computer Windows, invece, la via del Dual Boot che abbiamo illustrato nella sezione 1.4, non è nella sostanza percorribile.

2.1 Approccio 'Nativo' - le Command Line Tools

Con pochi passi è possibile configurare un sistema MacOS come un efficiente sistema per la programmazione in C e Python, senza la necessità di macchine virtuali o partizioni separate. Il primo passo necessario per sfruttare le funzionalità *developer* di MacOS è installare le *command line tools*. Per farlo, apriamo il terminale. Lo trovate nella cartella 'Applicazioni'. Qua sarà sufficiente provare a eseguire il comando

```
gcc
```

affinché il sistema ci chieda se vogliamo installare gli strumenti per sviluppatori da riga di comando. Accettando le condizioni del servizio inizierà l'installazione. Queste tools contengono gran parte del software che ci sarà necessario, come il compilatore GCC e un'installazione di Python (su cui torneremo in seguito). Per verificare la bontà dell'installazione, eseguiamo il comando

```
gcc --version
```

se visualizziamo la versione del compilatore senza messaggi di errore (vedi fig. 36), l'installazione è andata a buon fine.

2.1.1 Homebrew

Homebrew è un gestore di pacchetti per MacOS che ci permette di installare e aggiornare software aggiuntivi in maniera estremamente semplice. Una volta installato, infatti, sarà in

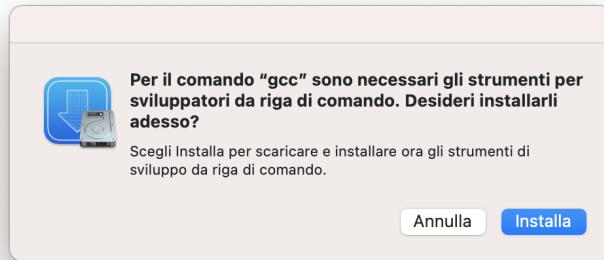


Figure 34: Quando digitiamo qualunque comando sviluppatore sul terminale il sistema in automatico ci chiederà di installare i tool necessari.

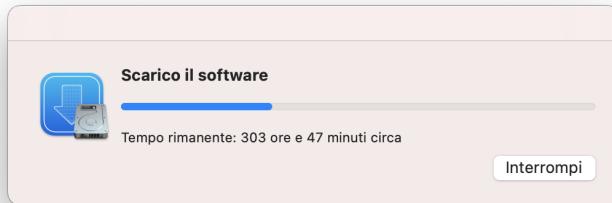


Figure 35: Per qualche motivo le stime del tempo residuo nell'installazione delle command line tools sono spesso completamente fuori scala. Scrivendo questa guida l'installazione è terminata in meno di 15 minuti nonostante la stima di oltre 300 ore.

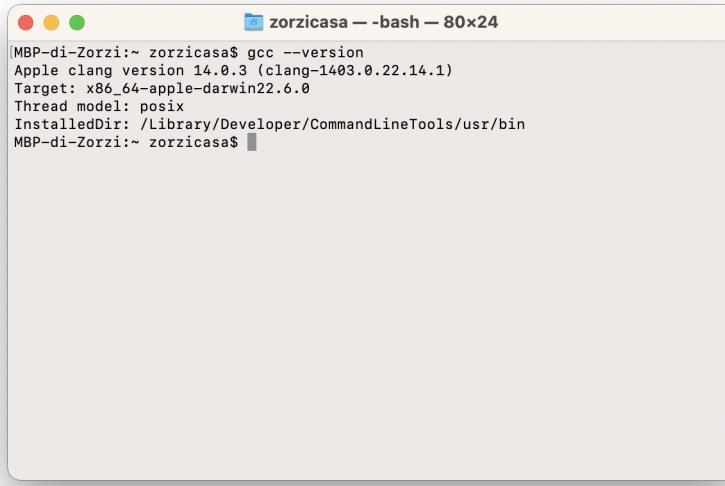


Figure 36: Se l'installazione delle command line tools è andata a buon fine, visualizzeremo le proprietà del compilatore gcc senza messaggi di errore.

grado di installare programmi appartenenti a una vastissima selezione eseguendo un singolo comando e di risolverne automaticamente tutte le dipendenze. Anche il processo di aggiornamento sarà estremamente semplificato. Per installarlo è sufficiente scaricare l'installer presente a questo³ link. Sarà sufficiente eseguire l'installer e seguire le semplici istruzioni per installare Homebrew sul vostro dispositivo. Come per gcc, possiamo verificare la bontà dell'installazione eseguendo

```
brew --version
```

e verificando che l'output non presenti errori. D'ora in avanti potremo installare la gran parte dei programmi di nostro interesse con il comando

```
brew install NomeSoftware
```

mentre con il comando

³Il link è funzionante a marzo '24. Qualora il link non dovesse funzionare mentre consultate questa guida, potete trovare l'installer sul sito ufficiale di Homebrew, brew.sh.

```
brew upgrade
```

aggioreremo tutti i programmi installati tramite Homebrew all'ultima versione disponibile.
Possiamo aggiornare Homebrew stesso con il comando

```
brew update
```

2.1.2 GCC

La procedura di installazione delle command line tools installerà nel vostro computer il compilatore CLang. Siccome i computer di laboratorio dispongono del compilatore GCC, può essere di vostro interesse installare lo stesso compilatore sul vostro computer. Pur essendo sostanzialmente analoghi nelle performance, i due compilatori producono messaggi di errori differenti e può pertanto essere comodo essere abituati agli stessi messaggi di errore che troverete in sede di esame. Aprire il terminale e digitare il comando:

```
brew install gcc
```

Una nota importante: per ragioni di compatibilità, il sistema operativo MacOS sovrascrive il comando 'gcc' con la chiamata al sopracitato CLang. Per usare *veramente* gcc, dovete chiamare esplicitamente la versione specifica installata sul vostro dispositivo. Per vedere le versioni disponibili, usiamo il comando:

```
brew list gcc
```

e avremo come output tutte le versioni disponibili. Il consiglio è di usare la più recente. Se, ad esempio, volessimo usare gcc-11, useremo:

```
gcc-11 xxxx
```

avendo cura di sostituire le x con delle keyword funzionanti.

2.1.3 Emacs

Una volta completata l'installazione di Homebrew, possiamo usarlo per installare alcuni programmi utili per la programmazione in C. Per installare l'editor di testo Emacs, digitiamo nel terminale il comando

```
brew install --cask emacs
```

Homebrew procederà all'installazione e potremo verificare il successo con il comando

```
emacs --version
```

Per iniziare a modificare un file testuale con emacs è sufficiente digitare

```
emacs NOMEFILE
```

Nel caso il vostro Pc monti una tastiera italiana, emacs potrebbe dare dei problemi nella digitazione di caratteri speciali come { o [. Per ovviare a questo problema, apriamo il file contenente le preferenze di emacs digitando:

```
emacs $HOME/.emacs
```

Se abbiamo appena installato il programma questo file dovrebbe essere vuoto. All'interno copiamo i seguenti comandi:

```
(setq mac-option-modifier 'none)
(setq mac-command-modifier 'meta)
(setq inhibit-startup-message t)
(setq initial-scratch-message nil)
```

Salviamo il file e chiudiamo tutte le finestre di emacs. Riavviando il software ora le combinazioni di tasti funzioneranno come di default.

2.1.4 Python3

MacOS contiene *out-of-the-box* un'installazione di Pyhton, ma si tratta della versione 2.7, piuttosto datata. Nelle ultime versioni delle command line tools, è stata inserita una versione più nuova, la 3.9.x, ma mancano la maggior parte dei pacchetti utili al calcolo numerico, come Numpy. Invece di installarli manualmente, il mio consiglio è di utilizzare una distribuzione di Python3 già predisposta al calcolo scientifico, come *Anaconda*. Per scaricarla andiamo al seguente link. Cliccando su *Download*, ci verrà data la scelta tra Apple Silicon



Figure 37: Per verificare il tipo di processore che monta il nostro dispositivo, clicchiamo sulla mela in alto a sinistra nello schermo e selezioniamo "Informazioni su questo Mac". Alla voce *Processore* (o *Chip*), troviamo l'architettura della CPU.

e Intel. Scegiamo il pacchetto adatto al sistema su cui stiamo lavorando (vedi fig. 37). Una volta scaricato il pacchetto, apriamo il file .pkg e accettiamo le condizioni di installazione. Riavviando il terminale, se l'installazione è andata a buon fine, dovremmo vedere comparire la scritta '(base)' all'inizio di ogni riga del terminale. Ciò significa che l'ambiente legato ad Anaconda è attivo e avremo accesso alle librerie per il calcolo scientifico. Possiamo verificare la loro effettiva presenza aprendo Python con il comando

```
python3
```

e inserendo i comandi

```
import numpy  
numpy.__version__
```

o

```
import matplotlib  
matplotlib.__version__
```

Se riusciamo ad eseguire i comandi senza ottenere messaggi d'errore, abbiamo installato correttamente Anaconda e abbiamo accesso alle librerie che ci serviranno.

2.2 Virtualizzazione - UTM

Qualora preferiate lavorare in ambiente Linux *puro* anche sul vostro Mac, è necessario installare un software per la virtualizzazione/emulazione di altri sistemi operativi. Se disponete di un Mac con processore x86 (Intel), Virtual Box è disponibile a questo link e potete seguire le indicazioni presentate alla sezione 1.2. Nel caso siate in possesso di un computer con CPU Apple Silicon (vedi fig. 37), il nostro consiglio è di usare UTM, un software per la virtualizzazione/emulazione esclusivo per MacOS, con performance migliori e un'interfaccia migliorata. Potete scaricare l'installer a questo link o direttamente dal Mac App Store. Una volta scaricato il file, eseguitelo e trascinate il logo di UTM all'interno della cartella 'Applicazioni'. Ora eseguite l'app e selezionate "Create a new virtual machine" (vedi fig. 38). Nel

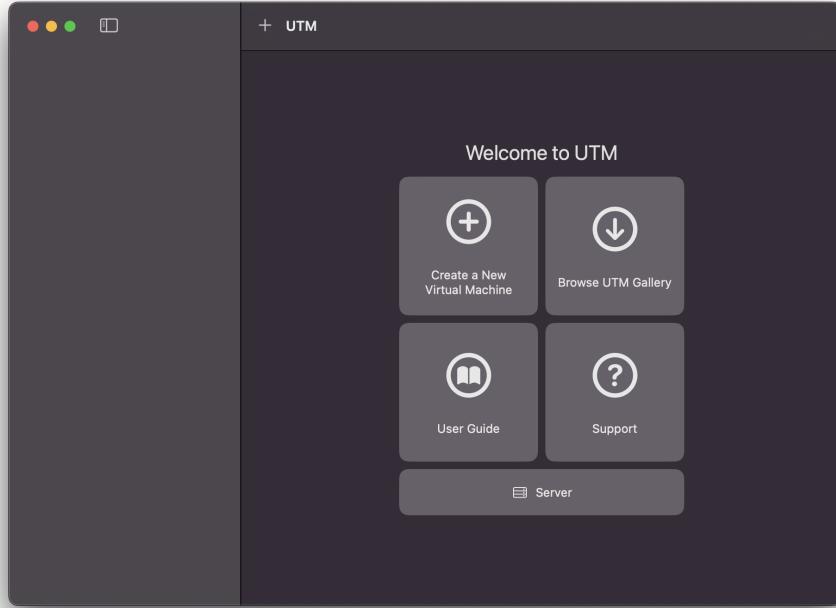


Figure 38: Schermata di avvio di UTM.

menù che si apre è possibile aprire una macchina virtuale già esistente cliccando su "Open...". Potete scaricare la macchina virtuale di laboratorio a questo link e selezionarla dal menù che si aprirà in UTM. Sarà possibile avviare immediatamente la macchina virtuale la quale si comporterà esattamente come i computer del laboratorio di calcolo.



Figure 39: Desktop della macchina virtuale di laboratorio. Per accedere utilizzare lo user-name 'studente' e la password 'informatica'.