



Preparatevi a entrare nel **mondo di Linux**:  
dove **il pinguino regna** e potrete dire **addio ai**  
**riavvii forzati!**

## COS'È LINUX?

Linux è stato **creato da Linus Torvalds** nel 1991.  
Esso è il **kernel** dei sistemi operativi GNU/Linux.

Il kernel non è altro che **il cuore** del sistema operativo, infatti esso **gestisce le risorse hardware** (CPU, RAM, ecc.).

Ma un cuore ha anche bisogno di un corpo, ed è qui che entra in gioco il **progetto GNU**.

GNU è un **insieme di strumenti base** (terminale, compilatori, ecc.) per creare un **sistema utilizzabile da utenti finali**.





**Linus Benedict Torvalds**



Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus ([torv...@kruuna.helsinki.fi](mailto:torv...@kruuna.helsinki.fi))

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

## OPEN SOURCE



Il **movimento open source** promuove un modello di sviluppo in diretto **contrasto con i modelli di sviluppo proprietari** in cui poche aziende controllano l'intero ciclo di vita del software.

Il modello open source ha dimostrato che per sviluppare **codice di qualità** non è necessario un grande budget, ma una **comunità impegnata ed appassionata**.

Grazie a questo modello, **Linux è riuscito a prosperare**, diventando una base solida sulla quale si reggono **infrastrutture critiche**.

## LE DISTRO LINUX

Le **distribuzioni Linux** (abbreviate in distro), sono ciò che lo rendono un sistema operativo **versatile ed adattabile**.

Ogni distro **combina il kernel** Linux con **software selezionato** e configurato per scopi specifici.

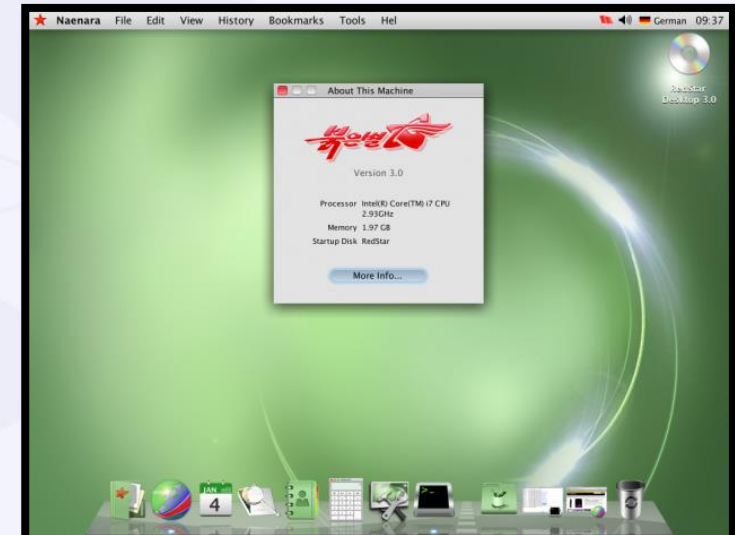
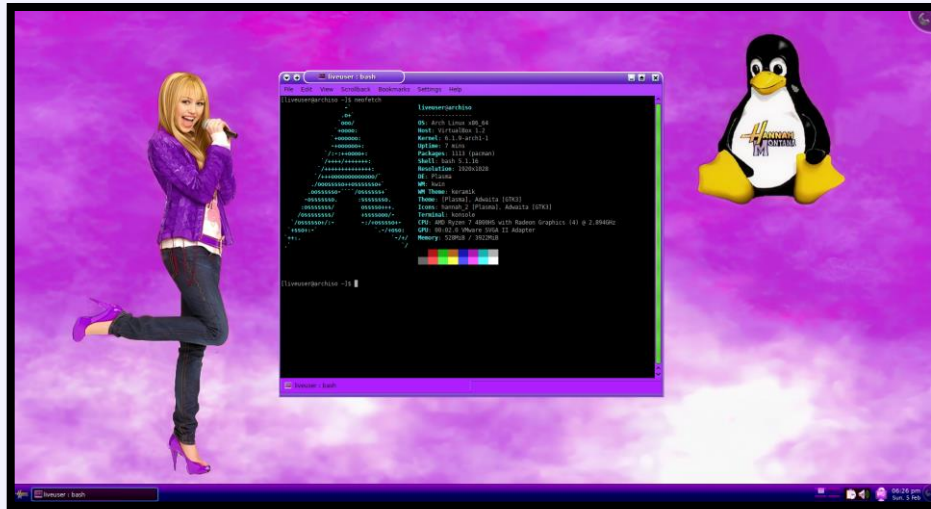
Ogni distro ha un **sistema di installazione di pacchetti diverso**. Questi contribuiscono a mantenere il sistema sicuro poiché scaricano il software da **versioni ufficiali**.

La vasta gamma di distribuzioni rende Linux **estremamente flessibile** in base alle esigenze.









## PERCHÉ È IMPORTANTE



L'importanza di Linux risiede, come abbiamo già detto, nella sua **capacità di adattarsi a contesti diversi**, dalla gestione di **server** a quella di **dispositivi embedded** (RaspberryPi, etc.).

È la spina dorsale dell'infrastruttura tecnologica globale, essendo **utilizzato in oltre il 90% dei server** che gestiscono Internet.

Inoltre, la sua natura open source permette di **scovare vulnerabilità in tempo zero**, rendendolo di fatto uno dei **sistemi operativi più sicuri al mondo**.



## MA QUINDI PERCHÉ UN PINGUINO?

La **mascotte ufficiale** di Linux è il **pinguino Tux**.

Il suo nome potrebbe derivare dal fatto che **il suo aspetto ricorda uno smoking** (tuxedo in inglese).

Tux è diventato la mascotte ufficiale perché incarna lo **spirito amichevole e non troppo serio** di Linus, il quale decise di trasmettere questi valori anche alla sua creatura.





## D'ACCORDO, E COME LO INSTALLO?

Per ognuna delle seguenti opzioni avrete delle **guide linkate nella repository GIT** del corso.

**Installare Ubuntu su una macchina vergine** o formattarne una non vergine.

**Installare WSL** (Windows Subsystem for Linux), questo vi permetterà di avere **dentro windows** una shell Linux.

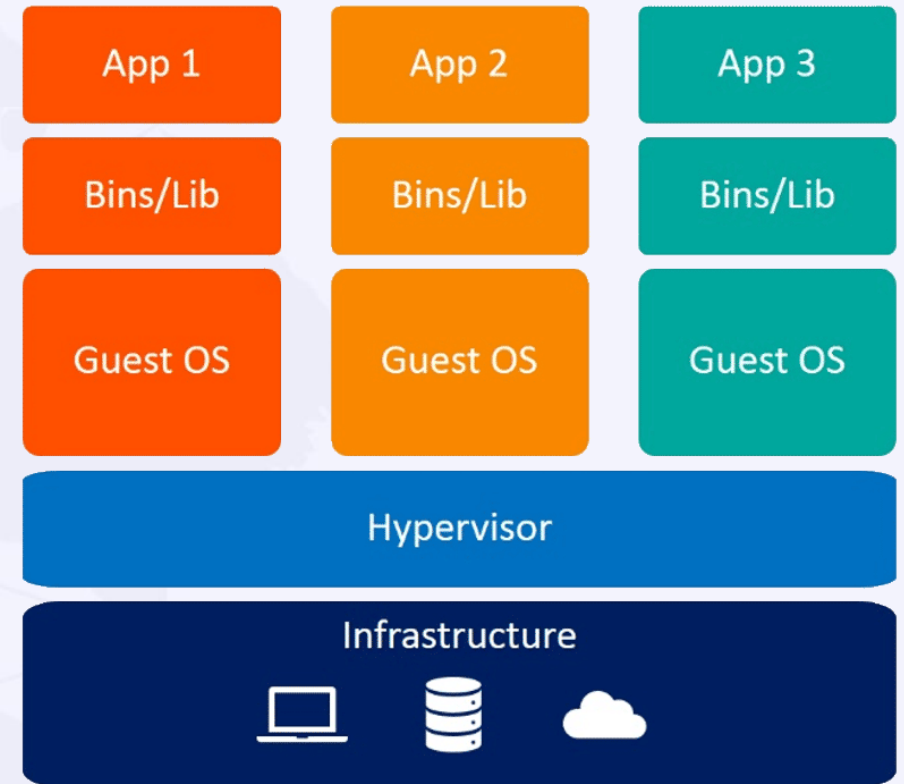
**Installare un software di virtualizzazione** come «Oracle Virtual Box».



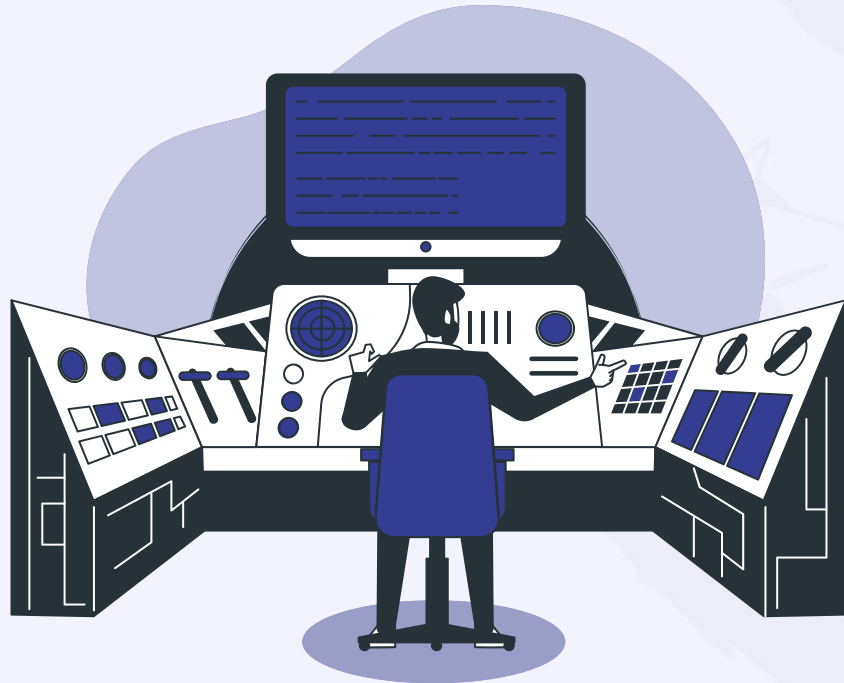
## MACCHINE VIRTUALI

Le **macchine virtuali** (VM) sono ambienti software che **emulano l'hardware di un computer reale**, permettendo di eseguire più sistemi operativi in parallelo su un'unica macchina fisica.

Attraverso una **hypervisor** (come VirtualBox), la VM **utilizza le risorse del sistema** host (come CPU, RAM e disco) per **creare uno spazio virtuale** dove un sistema operativo guest può funzionare indipendentemente.




## WINDOW'S SUBSYSTEM FOR LINUX




Il **Windows Subsystem for Linux (WSL)** è una funzionalità di Windows che **permette di eseguire un ambiente Linux direttamente su Windows**, senza la necessità di una macchina virtuale o del dual boot.

WSL fornisce **un'implementazione leggera del kernel Linux**, consentendo agli utenti di **eseguire la riga di comando** e strumenti nativi di Linux **direttamente su Windows**.





64-bit CPU  
4 GB RAM  
64 GB of storage  
DirectX 12 graphics  
UEFI, Secure Boot, TPM  
Required internet



The device is connected to electricity

Windows minimum requirements

Linux minimum requirements

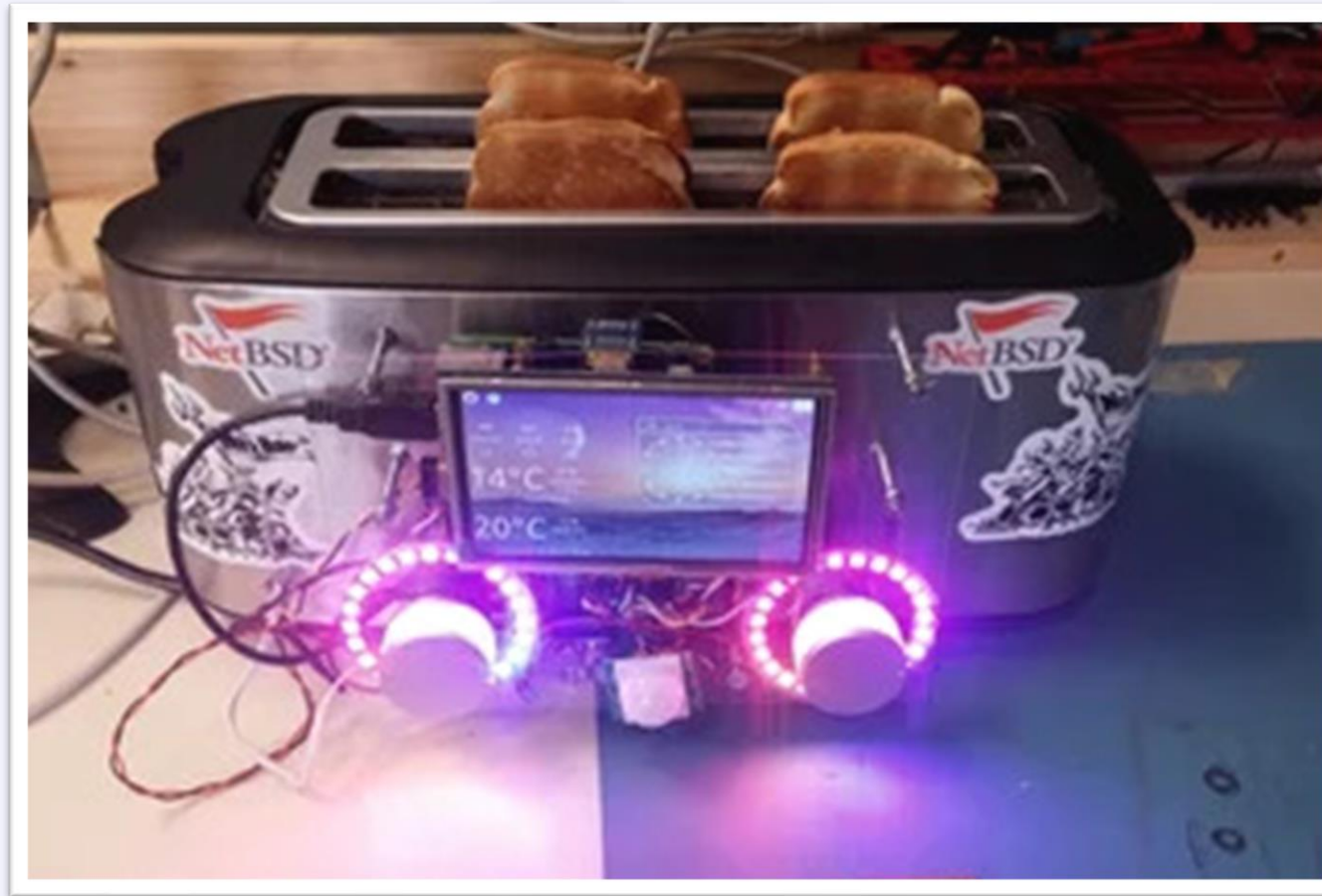


Requires Intel 8k/Ryzen 3k or better  
4GB RAM  
Must have TPM 2.0



CPU (optional)

Non avete scuse, **gira pure su un tostapane...**



Letteralmente...

## FINE INTRODUZIONE

**Grazie per l'attenzione**, spero che vi abbia intrattenuto abbastanza.

Domani spero di **vedervi tutti con una macchina Linux** per **sporcarvi le mani**.

Se c'è qualcosa che **non avete capito**, venite e **chiedete**, altrimenti mi potete **contattare su Telegram a @daniele.susino**



## BENTORNATI A CODING ZERO



Pronti a **trasformare codice in magia**?  
Nel **regno di Tux**, i **compilatori** e gli **interpreti** sono i tuoi  
**incantesimi**, e la **programmazione** è solo **l'inizio**  
**dell'avventura!**





**SIATE SINCERI...**

## AVETE AVUTO PROBLEMI?



Questo è il **momento giusto** di chiedere per **risolvere le problematiche**, non fatevi **intimorire da Tux**.

# COS'È IL FILE SYSTEM?

Il **filesystem** è come il sistema di **organizzazione dei file e delle cartelle** sul tuo computer.

Immagina di avere un **enorme armadio con tanti cassetti**, ognuno dedicato a qualcosa di diverso.

Il **filesystem** fa lo stesso: **organizza e tiene traccia di dove vengono salvati tutti i file**, come documenti, immagini e programmi, all'interno di una **struttura ad albero**.



## ASPETTA, STRUTTURA AD ALBERO?



Esatto! Il **filesystem** non è solo un metodo per memorizzare i file, ma una **struttura gerarchica complessa** che consente al sistema di **identificare, leggere, scrivere e gestire dati** in modo efficiente.

Ogni sistema operativo può utilizzare **diversi tipi di filesystem**, ed ognuno di essi ha le sue caratteristiche e **vantaggi specifici**.



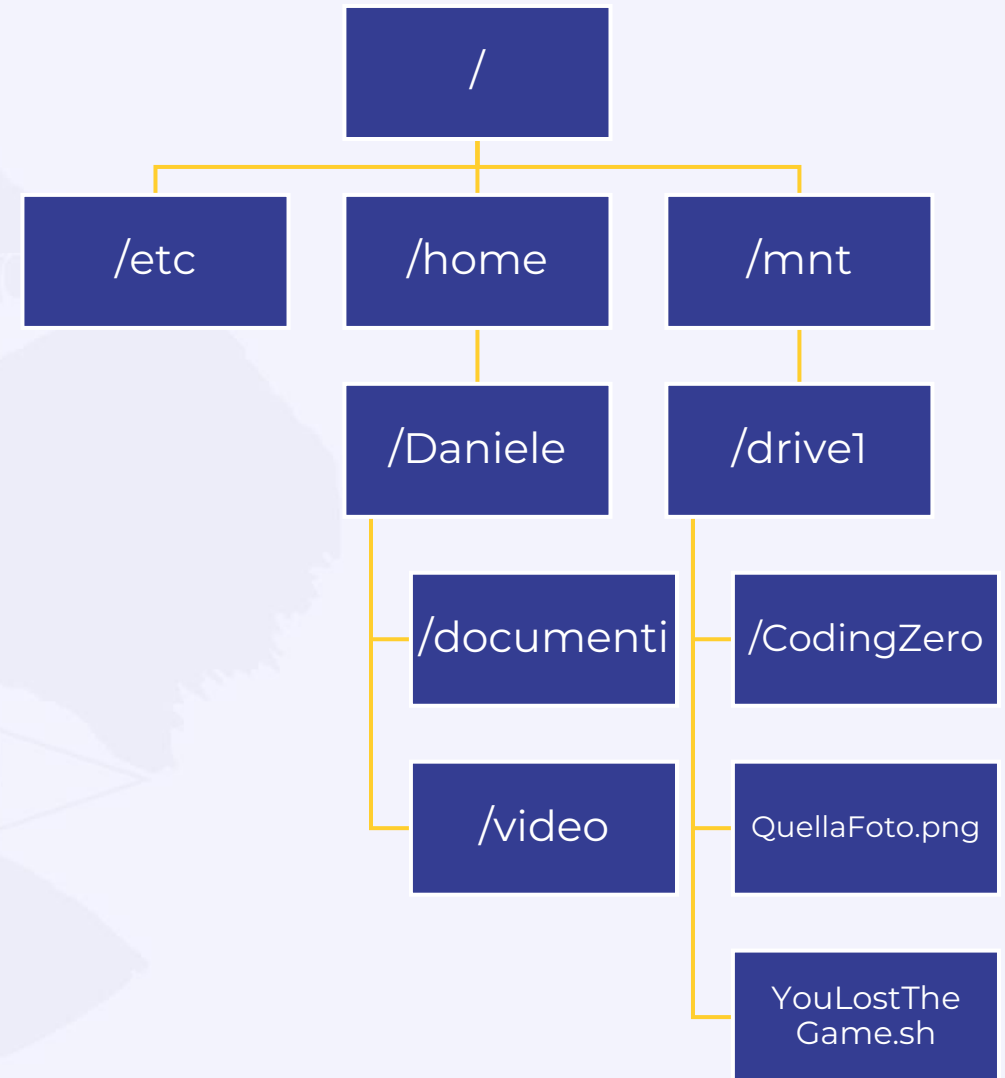
## E COME È STRUTTURATO?

La **directory** (cartella) **principale** è la cosiddetta **directory radice** «/».

Ogni elemento all'interno del sistema è **rappresentato tramite una directory oppure un file**, entrambi con rispettivi permessi e metadati (dati complementari).

Le **directory più importanti** sono:

- **/home**: cartella **dell'utente**.
- **/bin**: contiene i **programmi eseguibili**.
- **/mnt**: contiene i **dispositivi di archiviazione**.
- **/etc**: contiene i **file di configurazione**.



## È IL MOMENTO DI SCALDARE I MOTORI!

Se ancora non l'avete fatto, questo è il **momento giusto** per **accendere** le vostre **macchine Linux**.

Adesso vedremo qualche **comando base**, quindi se poteste **provarli mentre li spieghiamo** sarebbe perfetto.



# NAVIGARE NEL FILE SYSTEM

Per **muoversi nel file system** i **comandi base** sono i seguenti:

- **ls**: permette di **visualizzare tutto** ciò che è presente **nella directory corrente**.
- **cd**: permette di **entrare dentro una directory**.
- **cat**: permette di **visualizzare il contenuto di un file** sul terminale.
- **nano**: applicativo che permette di **modificare file di testo**.
- **touch**: permette di **creare un file di testo**.





## IL COMANDO «MAN»

Il **comando man** è sicuramente uno di quelli che **viene usato di più**.

Man sta per **Manual**, infatti qualora si usasse qualsiasi comando preceduto dal comando man, **vedremmo la sua documentazione**.



# IL PACKAGE MANAGER

Un **gestore di pacchetti** è un programma che semplifica **l'installazione e la gestione del software** su Linux, comprese le sue **dipendenze**.

Ogni **distribuzione** Linux ha il suo **gestore di pacchetti**: ad esempio, Ubuntu usa **APT**.

**I comandi principali di APT sono:**

- **sudo apt update:** Aggiorna le informazioni sulle versioni disponibili dei software.
- **sudo apt upgrade:** Aggiorna ogni pacchetto alla sua versione più recente.
- **sudo apt install <nomePacchetto>:** Installa un pacchetto specifico



## IL SUPER USER

Il **super user**, noto anche come **root**, è un utente speciale che ha il **controllo completo sul sistema**.

Per **proteggere il computer**, gli utenti normali (come il vostro) devono **ottenere solo temporaneamente i privilegi** di amministratore.

Questo viene fatto con il **comando sudo**, che permette di fare un **operazione con i privilegi** del super user in **modo controllato**.



# IL GNU C COMPILER

Il **GCC** (GNU C Compiler) è il **compilatore più utilizzato** per programmare in **C e C++**.

È un progetto **open source**, sviluppato dal **progetto GNU**.

Esso **traduce** codice sorgente **C** in **codice eseguibile**.

Per **compilare con GCC** il comando è il seguente: **gcc nomeFile.c -o nomeFile.out**

Per **eseguire** un file .out basta fare **./nomeFile.out**





## LA SKILL PIU' IMPORTANTE

La **cosa essenziale** per quanto riguarda la vita di un informatico, è **saper cercare le cose**.

USATE GOOGLE, **CERCATE**, INCURIOSITEVI, **SPULCIATE**. ANCORA MEGLIO IN **INGLESE**.

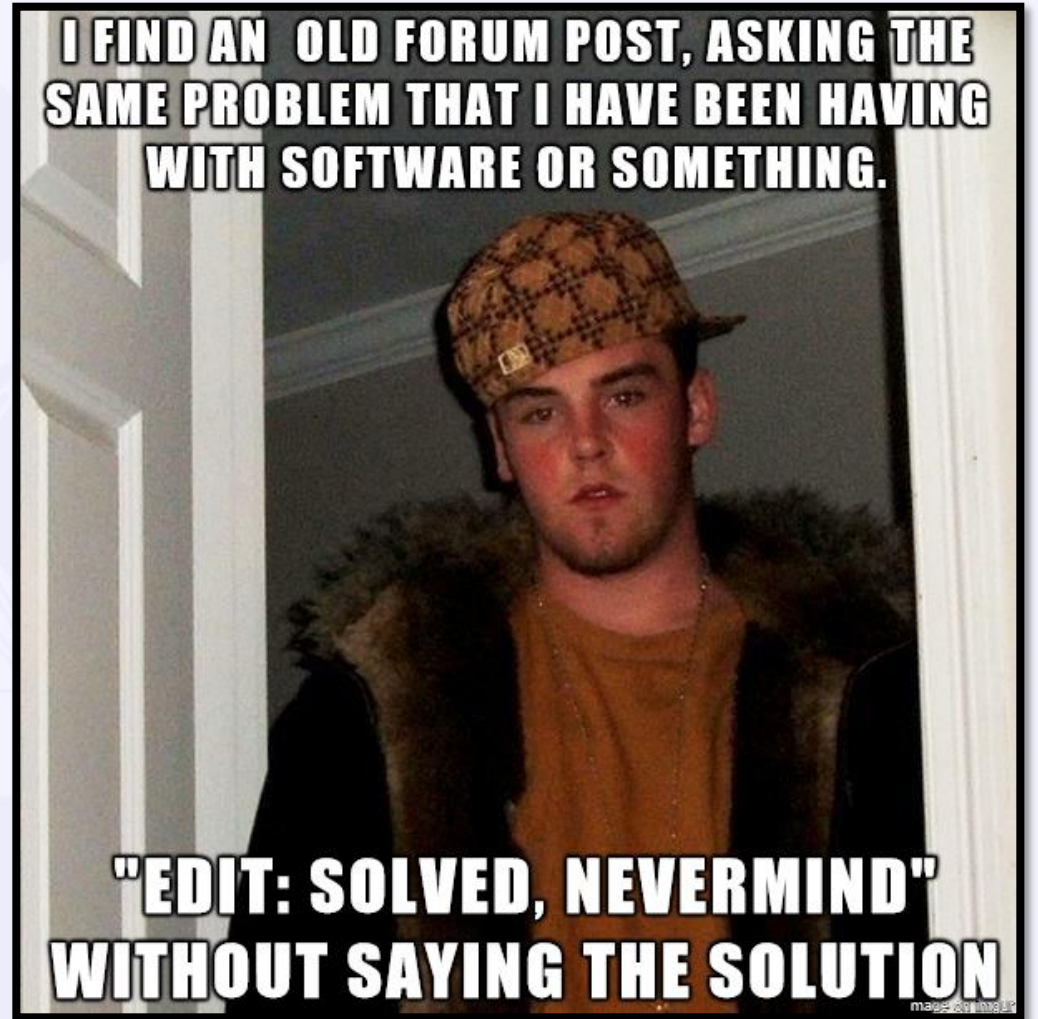


Obscure tech problem



Forum post from 2003

I FIND AN OLD FORUM POST, ASKING THE  
SAME PROBLEM THAT I HAVE BEEN HAVING  
WITH SOFTWARE OR SOMETHING.



"EDIT: SOLVED, NEVERMIND"  
WITHOUT SAYING THE SOLUTION



## TEMPO DI SPORCARCI LE MANI!



Spero che **abbiate capito tutto** fino ad ora.  
Dai **che si inizia a fare sul serio!**

## ALCUNI ESERCIZI

### CREAZIONE DI UN FILE DI TESTO:

- **Muovetevi** nella cartella **/home/nomeUtente**
- **Create** un file **«prova.txt»**
- **Scriveteci** qualcosa usando il **comando nano**
- **Visualizzate** il contenuto **con cat**

### CREAZIONE DI UNA CARTELLA:

- **Muovetevi** nella cartella **/home/nomeUtente**
- **Create** una cartella chiamata **«Test»**
- **Entrate** nella cartella **Test**
- **Create** un file al suo interno, **scriveteci qualcosa**, e **visualizzatene** il contenuto.
- **Eliminate** il **file** e poi la **cartella**

**Non abbiamo detto** appositamente **alcuni comandi**, provate a **cercarli voi!**





## ALCUNI ESERCIZI PARTE 2

### INSTALLAZIONE SL

Provate ad installare il pacchetto sl •  
Cercate di capire come funziona •

### GUERRE STELLARI

Cercare e capire come riprodurre star wars •  
sul terminale  
Capire come terminare un programma dal •  
terminale

## ORA DI PREPARARE IL NOSTRO AMBIENTE DI SVILUPPO!



Daje che abbiamo **quasi finito**

# CONFIGURAZIONE DELL'AMBIENTE DI SVILUPPO

## GNU C COMPILER:

- Sudo apt install gcc

## PYTHON:

- Controllare la presenza con **python3 --version**

## VSCODE:

- Scaricare da app center

## ESTENSIONI VSCODE:

- Ci servirà l'estensione «C/C++ Extension Pack»  
e l'estensione «Python»





## FINE LEZIONI LINUX

**Grazie per l'attenzione,** spero che vi abbia intrattenuto abbastanza.

Domani vi lascio a **Diego** che vi tratterà **con i guanti**, perché vi vorrà **sicuramente dissezionare**.

Se c'è qualcosa che **non avete capito**, venite e **chiedete**, altrimenti mi potete **contattare su Telegram a @daniele.susino**

