

**Università di Pisa**  
**Corso di Laurea in Ingegneria Informatica**

**Note sull'ambiente di lavoro utilizzato nei  
Laboratori di Fondamenti di Informatica I**

a cura di  
**Marco Cococcioni**

# Un po' di terminologia (1/2)

Processore (anche detto CPU-Central Processing Unit)	Unità di calcolo centrale del calcolatore. Si tratta del cuore di un personal computer, perché è l'unità che effettua tutte le operazioni fondamentali: lettura/scrittura in memoria, operazioni aritmetiche e logiche, esecuzioni di istruzioni, ecc... I processori possono essere significativamente diversi fra loro e generare incompatibilità di software. Alcuni dei principali produttori di processori sono: Intel, AMD, ARM e NVIDIA.
Memoria centrale (anche detta RAM)	Dispositivo che permette la memorizzazione temporanea dell'informazione, in unità elementari delle celle, ciascuna delle quali è dotata di un proprio indirizzo.
Memoria di massa (anche detto disco rigido – hard disk)	Dispositivo che consente di salvare dell'informazione in maniera persistente nel calcolatore, in modo che sia disponibile anche dopo lo spegnimento del calcolatore.
Periferiche di ingresso e uscita	Sono i dispositivi utilizzati per inserire informazioni dentro al calcolatore (tastiera, mouse, ecc...) e per portare verso l'esterno i risultati dell'elaborazione (monitor, casse audio, ecc...)
Sistema operativo	Software che serve a rendere facilmente utilizzabili al programmatore le risorse di calcolo (CPU, memoria centrale, ecc...) e le periferiche di ingresso e uscita disponibili, senza doversi preoccupare dei relativi dettagli hardware. In questo laboratorio si utilizzerà il sistema operativo Linux Debian. Altri sistemi operativi popolari sono: Windows 7/8 (Microsoft), Mac Os X (Apple) (Linux e Mac Os X sono basati sullo stesso predecessore: Unix).
File (anche detto archivio)	In un calcolatore le informazioni (sequenze di bit) vengono memorizzate all'interno di archivi logici detti file. I file si distinguono in <b>file ASCII</b> (sequenza di caratteri) e <b>file binari</b> (sequenza di numeri/codici). Un'altra distinzione possibile è in <b>file eseguibili</b> e <b>file non eseguibili</b> . Scopo della programmazione è scrivere file eseguibili dal calcolatore partendo da file ASCII contenenti istruzioni di un particolare linguaggio di programmazione.
Directory (anche detta cartella o folder)	Contenitore logico di file. Utilizzato per organizzare i file in maniera gerarchica, all'interno della directory principale (directory root: "/")
File System	Meccanismo fornito dal sistema operativo per la creazione di file e di directory, per la gestione della loro organizzazione logica all'interno di una struttura ad albero, per il controllo degli accessi, il controllo dell'integrità, ecc...

# Un po' di terminologia (2/2)

Directory root ('/')	Directory principale del File System (radice, letteralmente). Tutte le altre sono sotto-directory di root. Per spostarsi nella directory di root dare il seguente comando: <b>cd /</b>
Directory home	Per accedere al sistema operativo occorre autenticarsi con nome utente e password. Ad ogni utente corrisponde una directory speciale, detta directory home, dove può creare e cancellare file e directory a piacimento (infatti le directory di sistema non sono modificabili). Tale directory è /home/nomeUtente ( <b>/home/studenti</b> , nel nostro caso).
File Manager (anche detto file browser, o esplora risorse)	Si tratta di un programma dotato di interfaccia grafica (GUI-Graphical User Interface) che permette di gestire i file e le cartelle (creazione, cancellazione, spostamento) comodamente utilizzando il mouse. Linux Debian mette a disposizione thunar come FM ( <b>thunar nomeDir</b> ).
Programma (detto anche file eseguibile o applicazione)	Sequenza di operazioni eseguibili da un certo calcolatore (ossia da uno specifico sistema operativo ed uno specifico processore).
GUI (Graphical User Interface)	I programmi dotati di GUI sono quelli che sono dotati di una interfaccia grafica che facilita l'interazione con il programma attraverso bottoni ed altri elementi grafici.
Console (anche detta finestra terminale, terminal window, finestra di comando, riga di comando)	Si tratta di un particolare programma eseguibile che permette di inviare comandi al sistema operativo (esempi: <b>ls</b> , <b>pwd</b> , <b>cat nomeFile</b> ). Inoltre possono essere posti in esecuzioni dei file eseguibili creati dall'utente, lanciando il comando: ./nomeFileEseguibile Quando si apre una finestra di comando essa ci posiziona in un punto preciso del file system (tipicamente la directory home). La directory corrente può essere visualizzata con il comando <b>pwd</b> e cambiata con il comando: <b>cd nomeNuovaDirectory</b> .
Ambiente Desktop	Si tratta di un insieme di programmi che fornisce una interfaccia grafica al sistema operativo, presentandolo come una sorta di scrivania (Desktop), dotata di un menu start. In laboratorio viene utilizzato <b>Xfce</b> versione 4 (si avvia con il comando <b>startxfce4</b> ).
Editor	Programma che consente di creare, modificare e salvare file ASCII. L'editor di riferimento del corso è <b>gedit</b> . Un editor alternativo è Mousepad, ma ne viene sconsigliato l'uso.
Compilatore e Linker	Sono i due programmi chiave con cui è possibile generare un file eseguibile partendo da un file ASCII. Nel caso di programmi C++ la compilazione ed il collegamento (linking) avvengono mediante la chiamata al programma <b>g++</b> .
Debugger	Programma per la verifica della correttezza di un programma scritto dall'utente. In laboratorio verrà utilizzato il Data Display Debugger.

# L'ambiente di sviluppo utilizzato a laboratorio

Ambiente di sviluppo utilizzato a laboratorio	
Sistema operativo utilizzato	Linux (distribuzione Debian)
Compilatore	g++, versione 4.4.5
Debugger	gdb (gnu debugger)
Debugger Grafico	Data Display Debugger (ddd) - interfaccia grafica per gdb
Ambiente Desktop	Xfce versione 4 (startxfce4)
File Manager Grafico	thunar

L'immagine «iso» di un DVD «bootable», ossia avviabile, può essere scaricata al seguente link:

<http://www.iet.unipi.it/m.cococcioni/FondamentiDiInformatica/debian.iso>

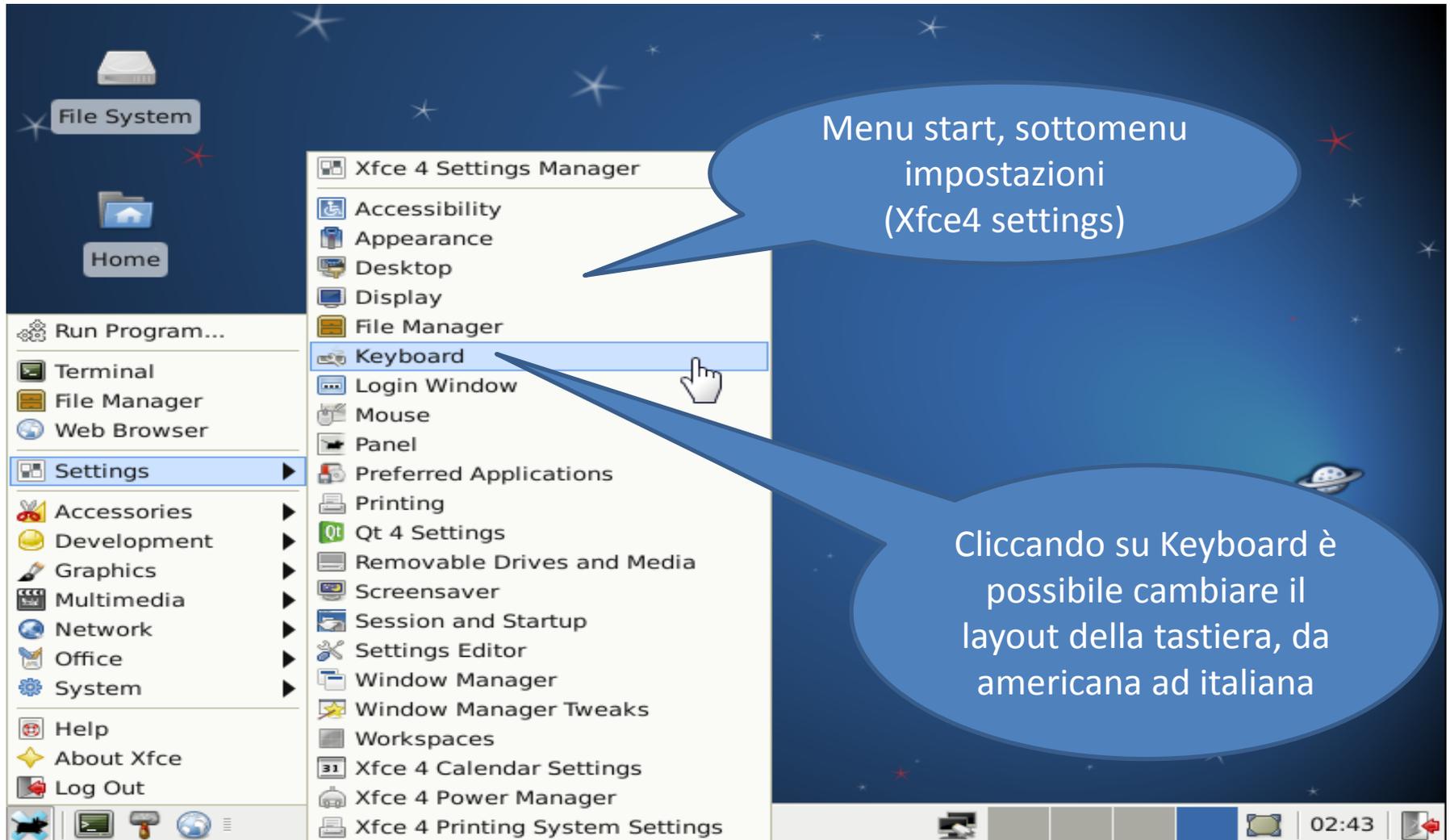
Le istruzioni su come effettuare la masterizzazione dell'immagine affinché risulti «bootable» si possono reperire qui:

<http://www.iet.unipi.it/m.cococcioni/FondamentiDiInformatica>

# Un'occhiata al sistema operativo Linux Debian ed al desktop manager Xfce

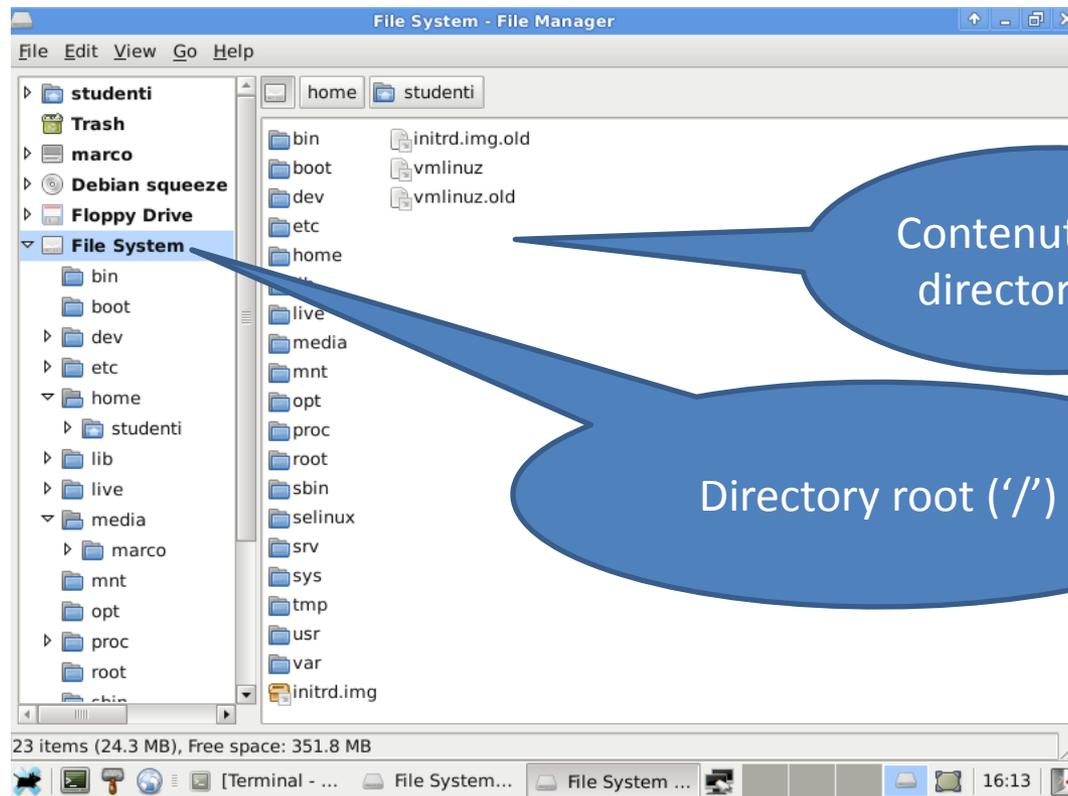


# Il Menu Start (di Xfce)

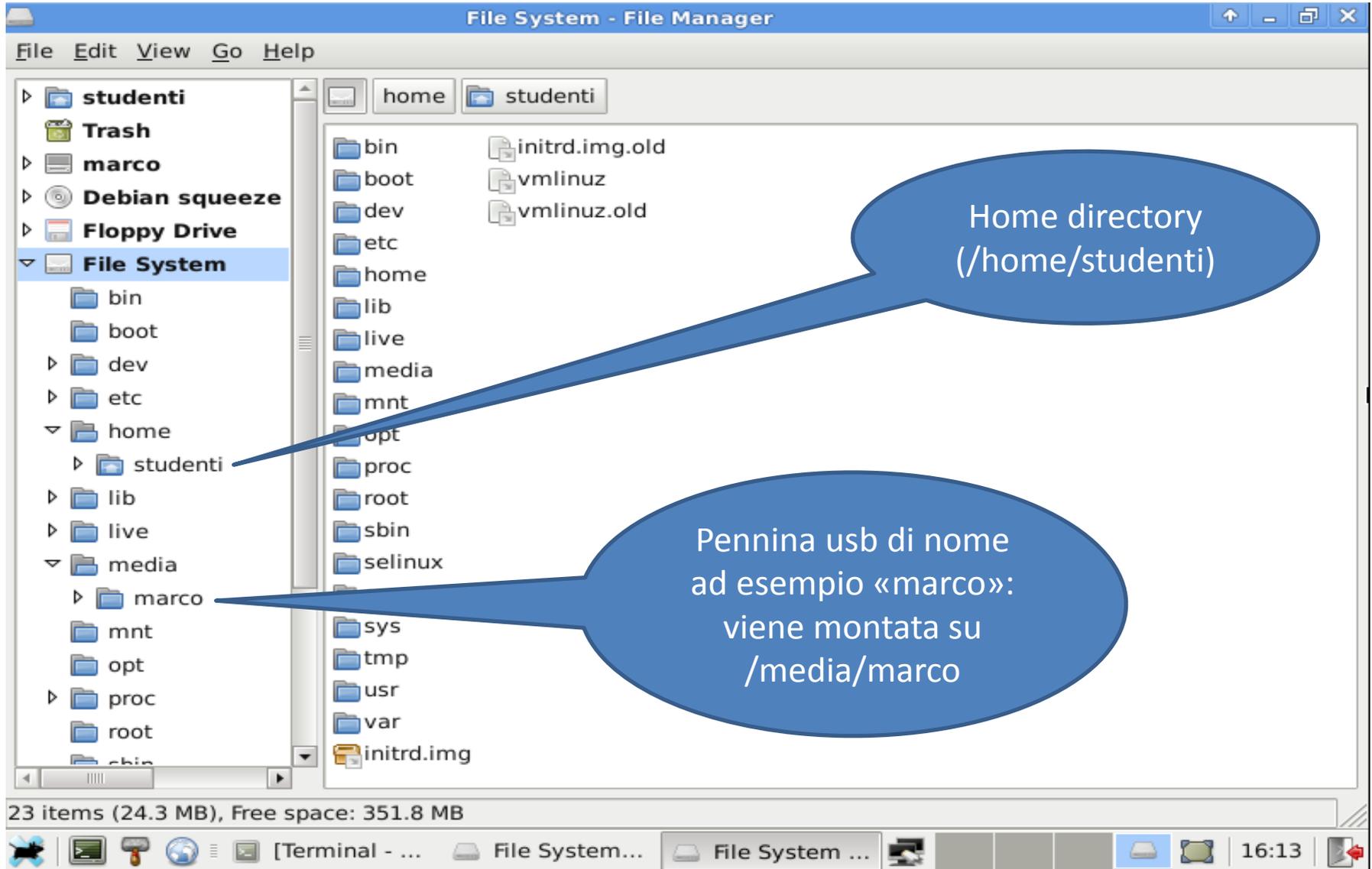


# Organizzazione delle directory (1/4)

Linux Debian implementa molto fedelmente in **File System Standard** dei sistemi unix-like. Questo standard definisce il nome e la posizione delle directory del File System



# Organizzazione delle directory (2/4)



# Organizzazione delle directory (3/4)

File System - File Manager

File Edit View Go Help

home studenti

bin boot dev etc home lib live media mnt opt proc root sbin selinux srv sys tmp usr

bin  
boot  
dev  
etc  
home  
lib  
live  
media  
mnt  
opt  
proc  
root  
sbin  
selinux  
srv  
sys  
tmp  
usr

initrd.img

23 items (24.3 MB), Free space: 351.8 MB

[Terminal - ...] File System... File System ... 16:13

Directory bin  
Contiene i file eseguibili:  
pwd, ls, cd, cat, ecc...

Directory delle periferiche  
(device). Ad esempio:  
/dev/null

Configurazione del sistema  
(etc sta per *etcetera*)

Directory home per  
l'utente root  
(amministratore di sistema)

# Organizzazione delle directory (4/4)

File System - File Manager

File Edit View Go Help

home studenti

- bin
- boot
- dev
- etc
- home
- lib
- live
- media
- mnt
- opt
- proc
- root
- sbin
- selinux
- srv
- sys
- tmp
- usr
- var

initrd.img

initrd.img.old

vmlinuz.old

23 items (24.3 MB), Free space: 351.8 MB

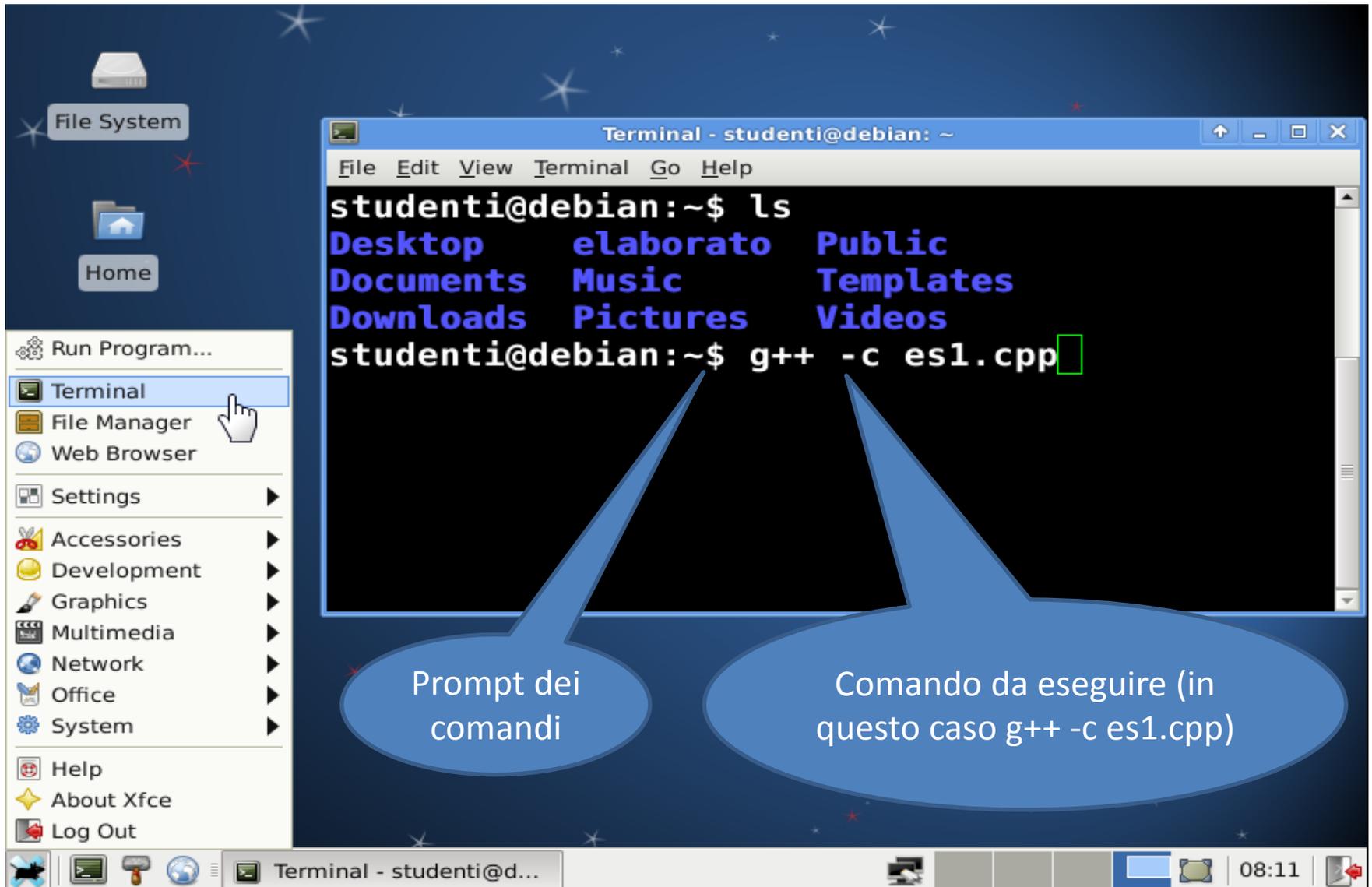
[Terminal - ...] File System... File System ... 16:13

File per il boot loader (kernel, initrd)

Eseguibili per l'amministrazione del sistema (system binary: init, route, ecc...)

File Variabili come log, database, siti web e file temporanei delle e-mail

# La console (finestra terminale)



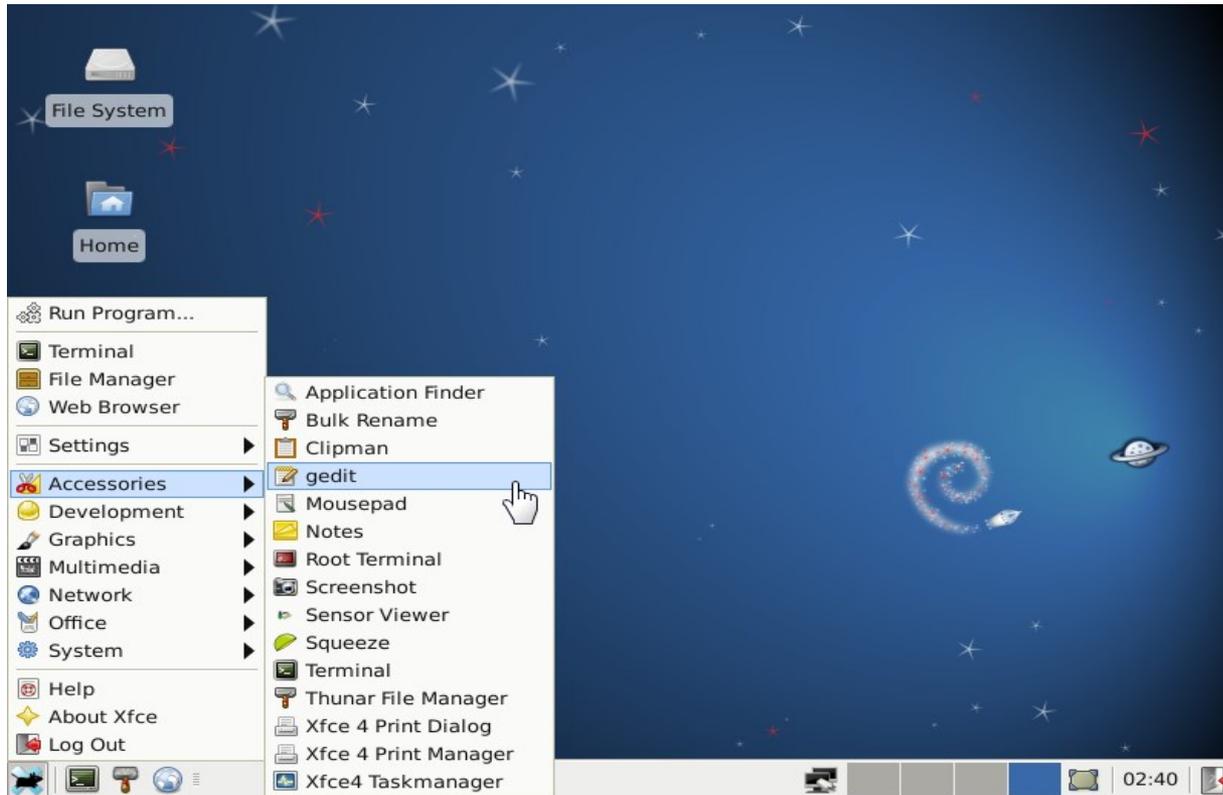
# Comandi base per la console

Comandi base	
<code>pwd</code>	Mostra il percorso ( <i>path</i> ) della directory corrente, ossia della <i>working directory</i> ( <code>pwd</code> sta per <i>path of the working directory</i> )
<code>ls</code>	Mostra i file e le directory presenti nella directory corrente. <b>ls -al</b> mostra anche i file nascosti e altre informazioni sui file/directory (dimensione, data, ecc...)
<code>cd percorso</code>	Fa sì che la nuova directory corrente sia <i>percorso</i> . Esempio: 'cd /media' ci porta su '/media'. Invece 'cd ..' ci sposta nella directory superiore.
<code>cp file1 file2</code>	Copia file1 in file2.
<code>mv file1 file2</code>	Rinomina <i>file1</i> come <i>file2</i> . Vale anche per directory.
<code>rm nomeFile</code>	Cancella il file <i>nomeFile</i> . Il file non può più essere recuperato, in quanto non finisce nel cestino.
<code>cat nomeFile</code>	Mostra a video il file <i>nomeFile</i> .
<code>mkdir nomeDir</code>	Crea la nuova directory <i>nomeDir</i> .
<code>rmdir nomeDir</code>	Cancella la directory <i>nomeDir</i> , purché sia vuota.
<code>rm -r nomeDir</code>	Rimuove la directory dopo averla svuotata.
<code>./nomeFile</code>	Esegue <i>nomeFile</i> , qualora esso sia un file eseguibile presente nella directory corrente (per renderlo eseguibile: <b>chmod +x nomeFile</b> ).
<code>clear</code>	Pulisce lo schermo.
<code>g++ -c esX.cpp</code>	Compila <i>esX.cpp</i> (produce file binario in codice oggetto <i>esX.o</i> ).
<code>g++ esX.o</code>	Collega (linking) il file <i>esX.o</i> e produce l'eseguibile <i>a.out</i> (per eseguirlo: <b>./a.out</b> ).

# Altri comandi per la console

Comandi avanzati	
<code>more nomeFile</code>	Alternativa a cat: mostra a video <i>nomeFile</i> , una pagina per volta. Si passa alla pagina successiva premendo la barra spaziatrice. Si esce con <i>q</i> .
<code>PS1="nuovo prompt"</code>	Comando per cambiare il prompt (es.: <code>PS1="\$ "</code> ).
<code>Contrl + C</code>	Interrompe il processo in esecuzione (utile per terminare programmi errati che vanno in ciclo o quando un programma non risponde)
<code>ps</code>	Visualizza i programmi attualmente in esecuzione (i così detti processi). Ad ogni processo è assegnato un numero univoco X.
<code>ps -A</code>	Visualizza tutti i processi, compresi quelli di sistema.
<code>kill X</code>	Termina il processo avente numero X.
<code>tree</code>	Visualizza il sotto-albero della directory corrente in maniera grafica.
<code>find -name nomeFile</code>	Mostra il percorso del file <i>nomeFile</i> qualora si trovi nella directory corrente o in una delle sue sottodirectory (equivale al comando windows <code>dir /s nomeFile</code> ) Esempi: <b>find -name *.cpp</b> , <b>find /home/studenti -name main.cpp</b>
<code>chmod +x nomeFile</code>	Rende eseguibile il file <i>nomeFile</i> (x sta per eXecutable). Altre opzioni interessanti del comando sono: <b>chmod +w nomeFile</b> (lo rende modificabile), <b>chmod -w nomeFile</b> rende il file accessibile in sola lettura.
<code>iceweasel sitoWeb</code>	Apri la pagina web <a href="http://sitoWeb">http://sitoWeb</a> usando il browser <b>iceweasel</b>
<code>epdfview nomeFilePdf</code>	Visualizza il file pdf <i>nomeFilePdf</i> ( <b>ePDFViewer</b> è una alternativa ad Acrobat Reader)
<code>cmd --help</code>	Fornisce l'help per il comando <i>cmd</i> (Es: <code>ls --help</code> )
<code>soffice -impress filePpt</code>	Visualizza il file Power Point <i>filePpt</i> , mediante <b>Open Office Impress</b>

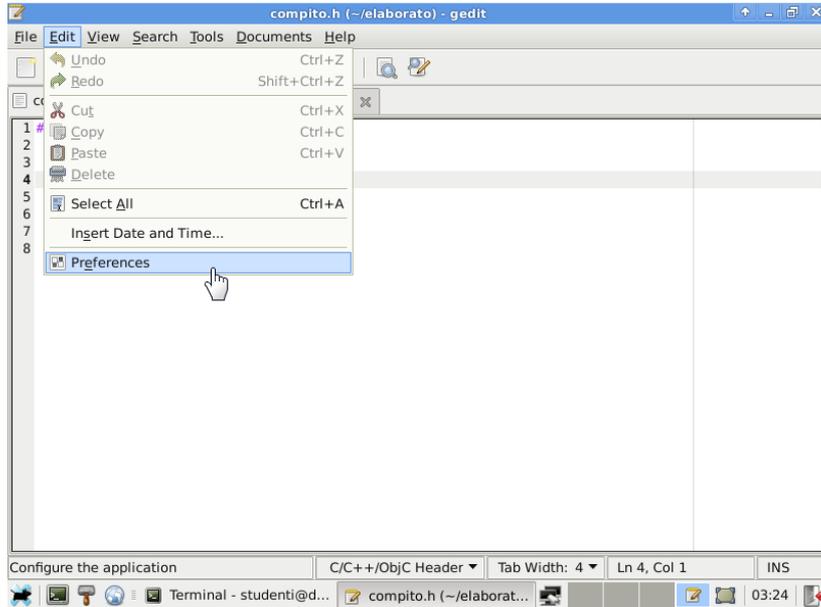
# L'editor (gedit)



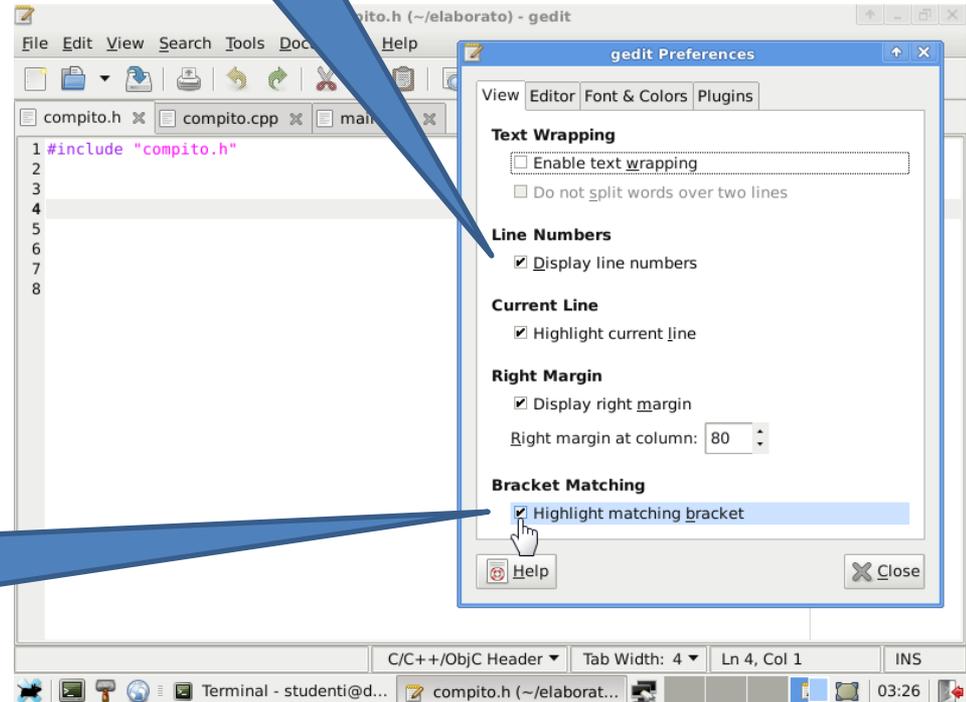
Può essere avviato anche da riga di comando:  
**gedit nomefile**

Un modo molto conveniente di avviare gedit è il seguente:  
**gedit main.cpp compito.cpp compito.h & >/dev/null**

# Cambiare le impostazioni di gedit

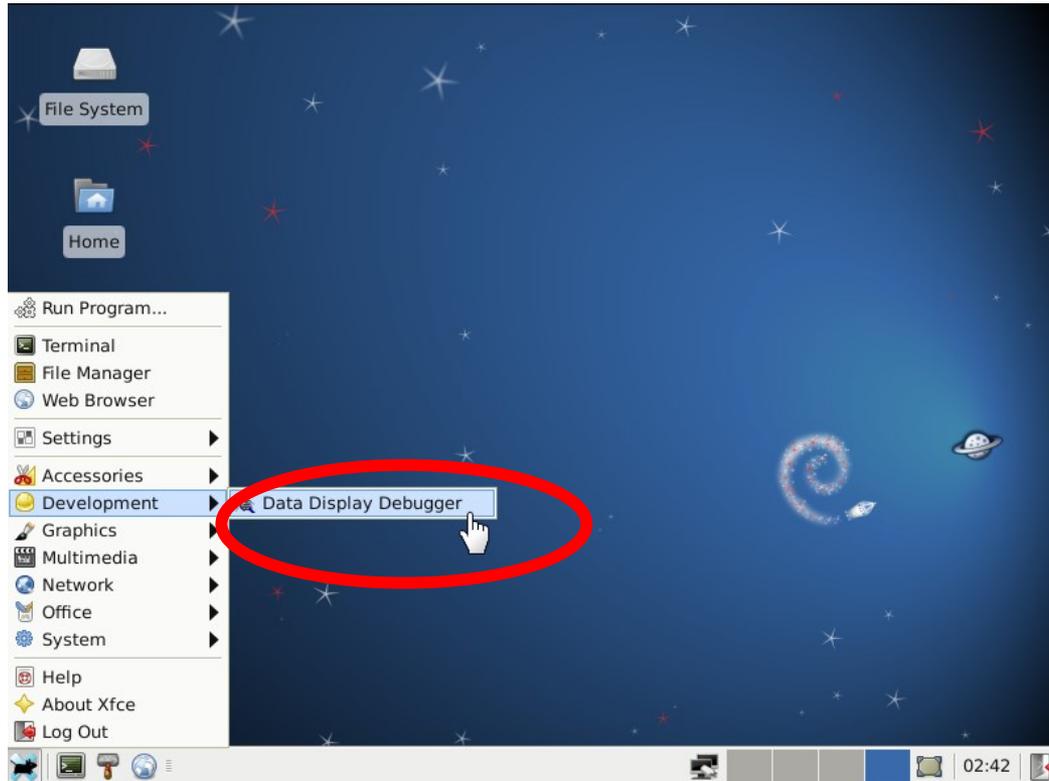


Visualizza il numero di riga (molto utile in fase di debug)



Visualizza la parentesi chiusa (aperta) corrispondente ogni volta che il mouse si trova sopra una parentesi aperta (chiusa)

# Il Debugger Grafico (ddd)



Può essere avviato anche da riga di comando:

```
ddd esercizio.exe
```

purché l'eseguibile contenga le informazioni di debug, ossia sia stato compilato con opzione `-g`: 

```
g++ -g -o esercizio.exe esercizio.cpp
```