

# NOZIONI BASE PER ESERCITAZIONI

---

DIPARTIMENTO DI FISICA



SAPIENZA  
UNIVERSITÀ DI ROMA

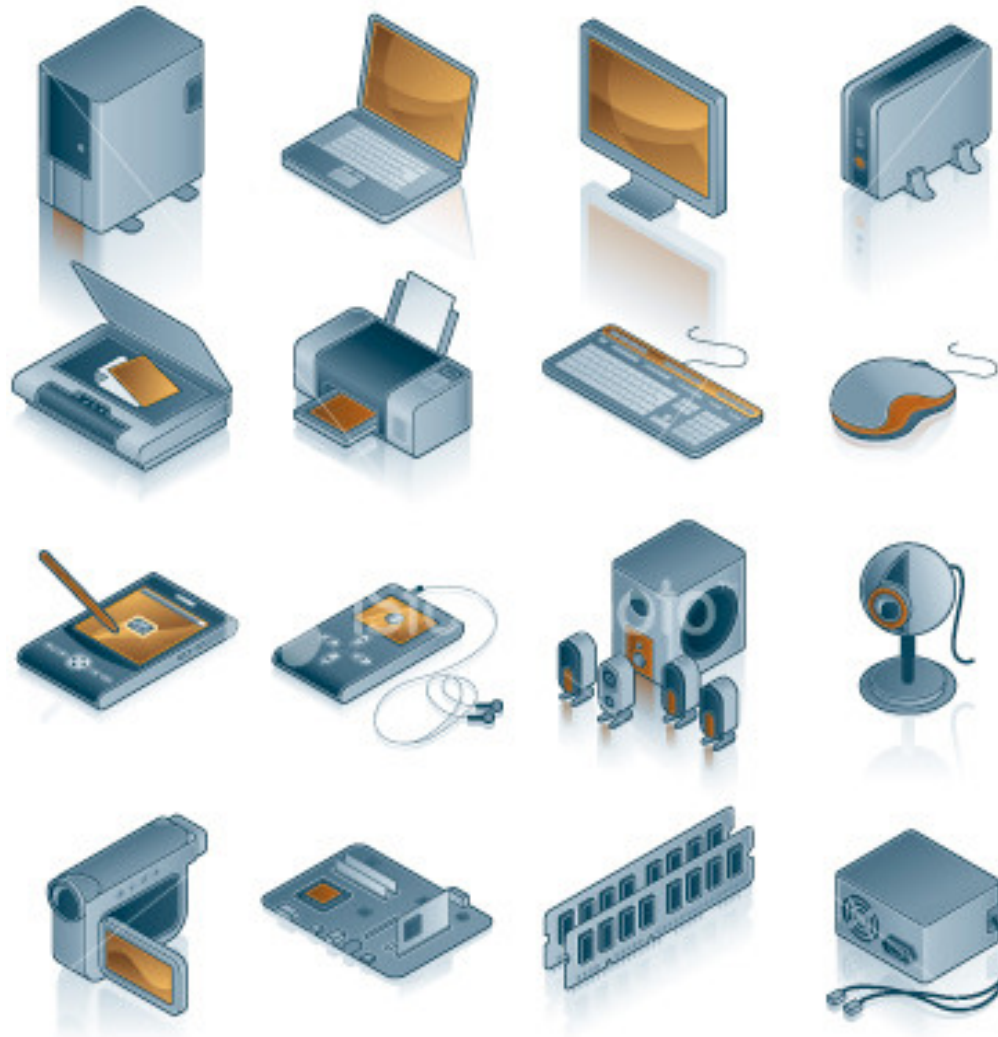
Shahram Rahatlou

Laboratorio di Calcolo, Anno Accademico 2024-2025

<http://www.roma1.infn.it/people/rahatlou/labcalc/>

# Sistema Operativo

Hardware

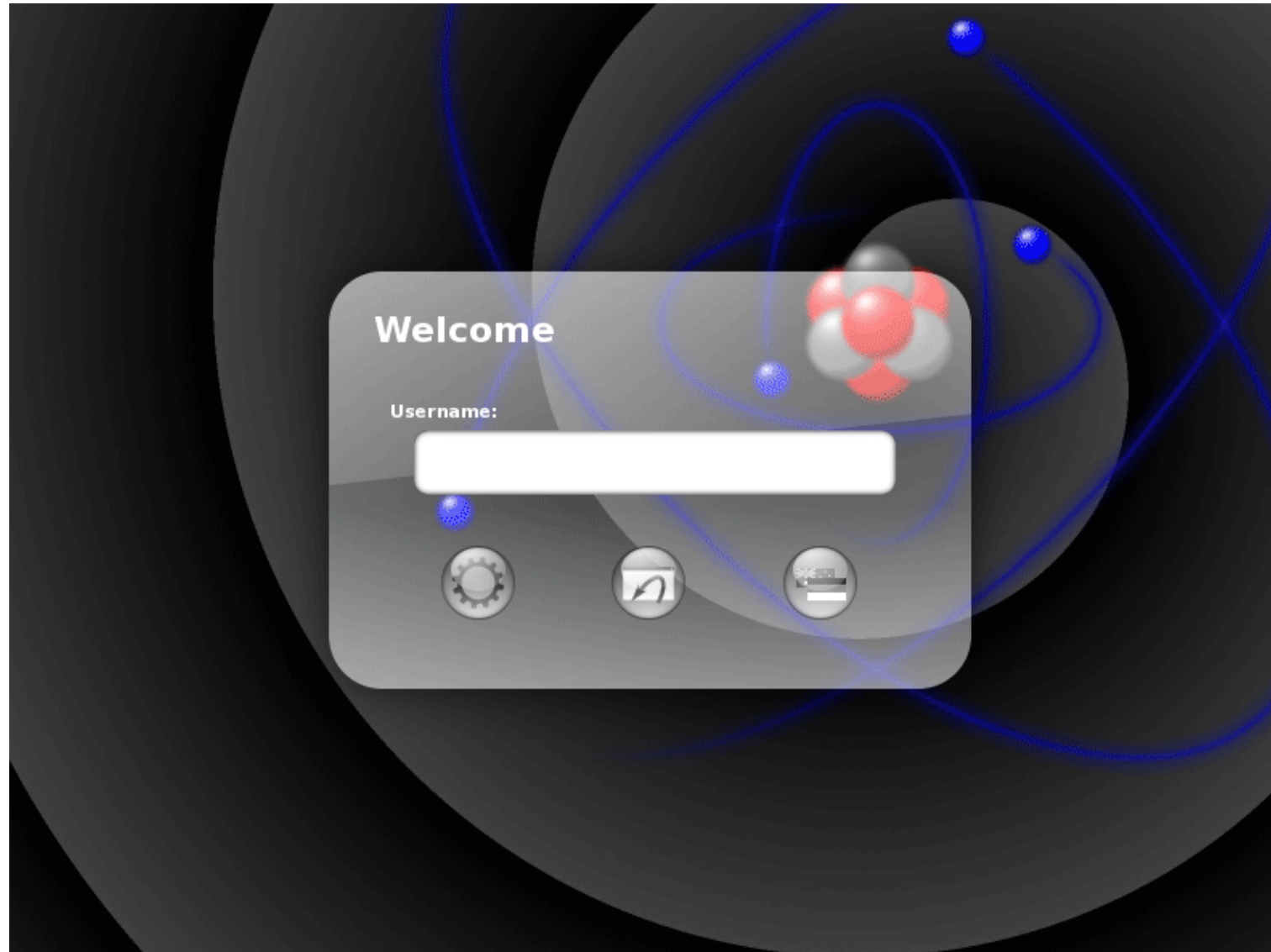


Software



- Routine e programmi di base per interagire con le componenti (hardware) di un calcolatore
  - Accedere alle risorse come il disco fisso
  - Utilizzare le periferiche per Input/Output (I/O)
    - Mouse, tastiera, schermo
  - Compilare programmi, creare eseguibili e girarli

# Accesso al sistema (Login)



- Per l'accesso servono uno **username** ed una **password**
- A ciascun gruppo è assegnato uno username del tipo ***lcsrXXX*** dove ***XXX=001, 02, ..., 099***

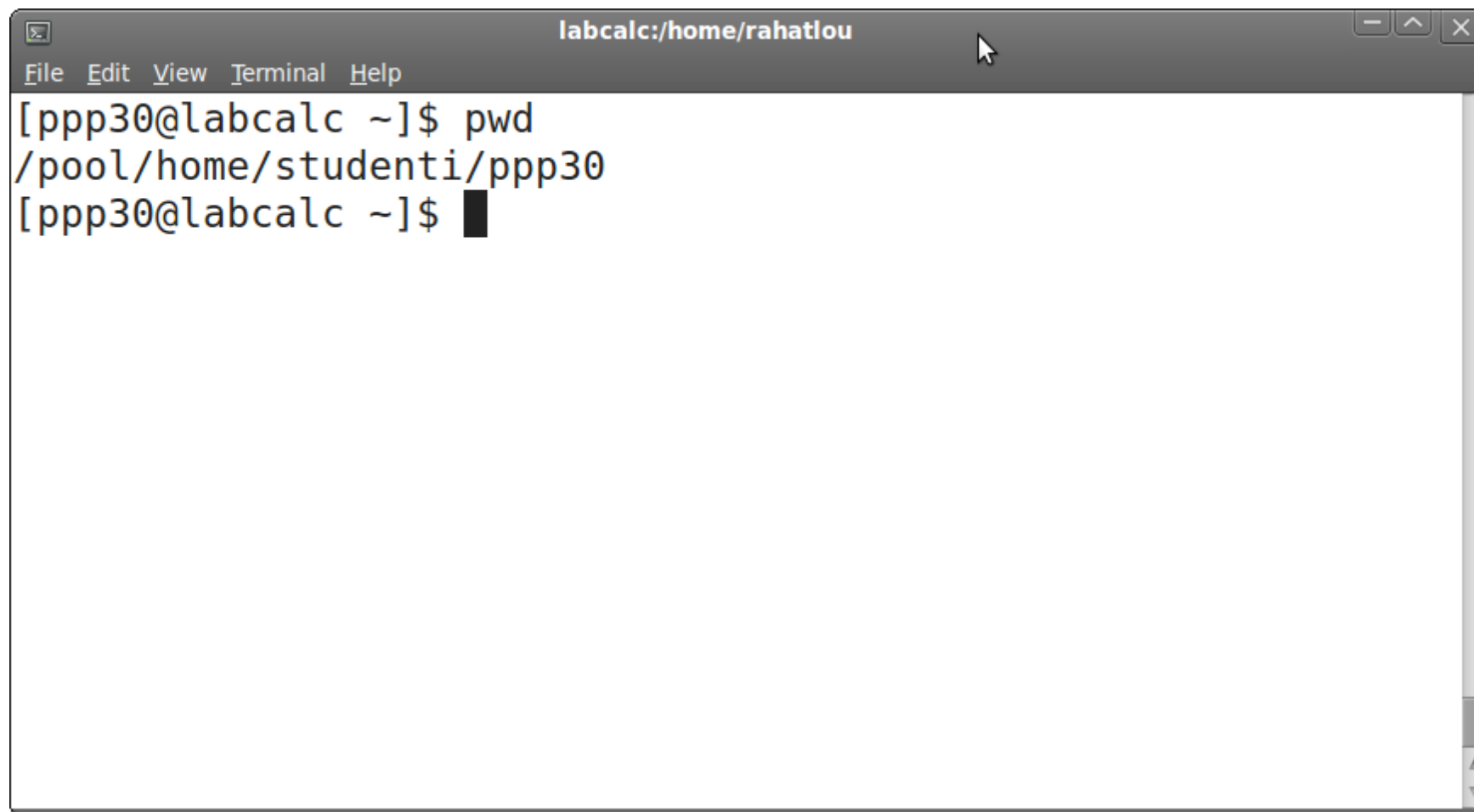
# Terminale e Shell

Prompt  
dell'utente



- L'utente può impartire comandi al sistema utilizzando la **shell**
- La **shell** è accessibile all'interno di una finestra di riga di comando chiamata comunemente il terminale

# Home Directory

A terminal window titled 'labcalc:/home/rahatlou' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the command 'pwd' being executed, resulting in the output '/pool/home/studenti/ppp30'. The prompt is '[ppp30@labcalc ~]\$' with a black cursor.

```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ pwd
/pool/home/studenti/ppp30
[ppp30@labcalc ~]$
```

- Aperta una shell l'utente si trova nella sua area 'home directory
- Si tratta di uno spazio assegnato a ciascun utente dove immagazzinare i suoi file
- Gli altri utenti (se autorizzati) possono vedere i file ma non modificarli
- E' possibile vedere il percorso(path) della directory con il comando **pwd**



# Lista dei file e cartelle con `ls`

```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ ls
a.out      Desktop   prova.c   teach.ico test.o
data2.txt  main     prova.c~ test.c   test.txt~
data.txt   mario    prova.o   test.c~  WeightedAverage.pdf
[ppp30@labcalc ~]$
```

- Potete elencare la lista dei file e cartelle contenuti nella cartella in cui vi trovate con il comando `ls`
- Alcuni tipi di file potrebbero essere visualizzati con un colore
  - Non è proprietà del file ma dipende dalla configurazione della shell

# Navigazione nelle cartelle (directory)

```
labcalc.fisica.uniroma1.it:/home/docenti/rahatlou — ssh — 85x26
$ pwd
/home/docenti/rahatlou
$ ls
animated_favicon1.gif  Desktop/  labcalc@  public_html/  tiburtina.tgz
C/  favicon.ico  private/  root@
C++/  index.html@  programmazione++@  temp/
$ cd C
$ pwd
/home/docenti/rahatlou/C
$ ls
mularray.c  multarr2.c  nav18*  ptr18*  ptr21*  ptr.exe*  testreturn.c
mularray.o  multarr2.o  nave1.18.c  ptr18.c  ptr.c  testreturn*  uso_++.c
$ cd ..
$ pwd
/home/docenti/rahatlou
```

comando **ls** per vedere la lista dei file in una cartella

comando **cd** per spostarsi in una nuova cartella

(punto punto) **..** : un simbolo per tornare rapidamente alla cartella contenente quella attuale

comando **pwd** per vedere il percorso della cartella dove ci troviamo

# Colori non sono importanti!

- Molti terminali in diversi sistemi operativi mostrano file di tipi diversi con colori diversi
- Solo un artificio grafico per aiutare l'utente
- Sistema operativo non sa nulla e non usa i colori!



A terminal window titled 'labcalc.fisica.uniroma1.it:/home/docenti/rahatlou — ssh — 85x26'. The terminal shows the output of the 'ls' command, which lists files and directories in a plain text format. The output is as follows:

```
$ ls
animated_favicon1.gif Desktop/ labcalc@ public_html/ tiburtina.tgz
C/ favicon.ico private/ root@
C++/ index.html@ programmazione++@ temp/
$
$
$
$
$
$
$
$
$
$
$ ls --color
animated_favicon1.gif Desktop/ labcalc@ public_html/ tiburtina.tgz
C/ favicon.ico private/ root@
C++/ index.html@ programmazione++@ temp/
$
```

The second output, following the 'ls --color' command, shows the same files and directories but with color coding: 'animated\_favicon1.gif' is purple, 'Desktop/' is blue, 'labcalc@' is cyan, 'public\_html/' is blue, and 'tiburtina.tgz' is red. The other files and directories are also color-coded in a similar manner.



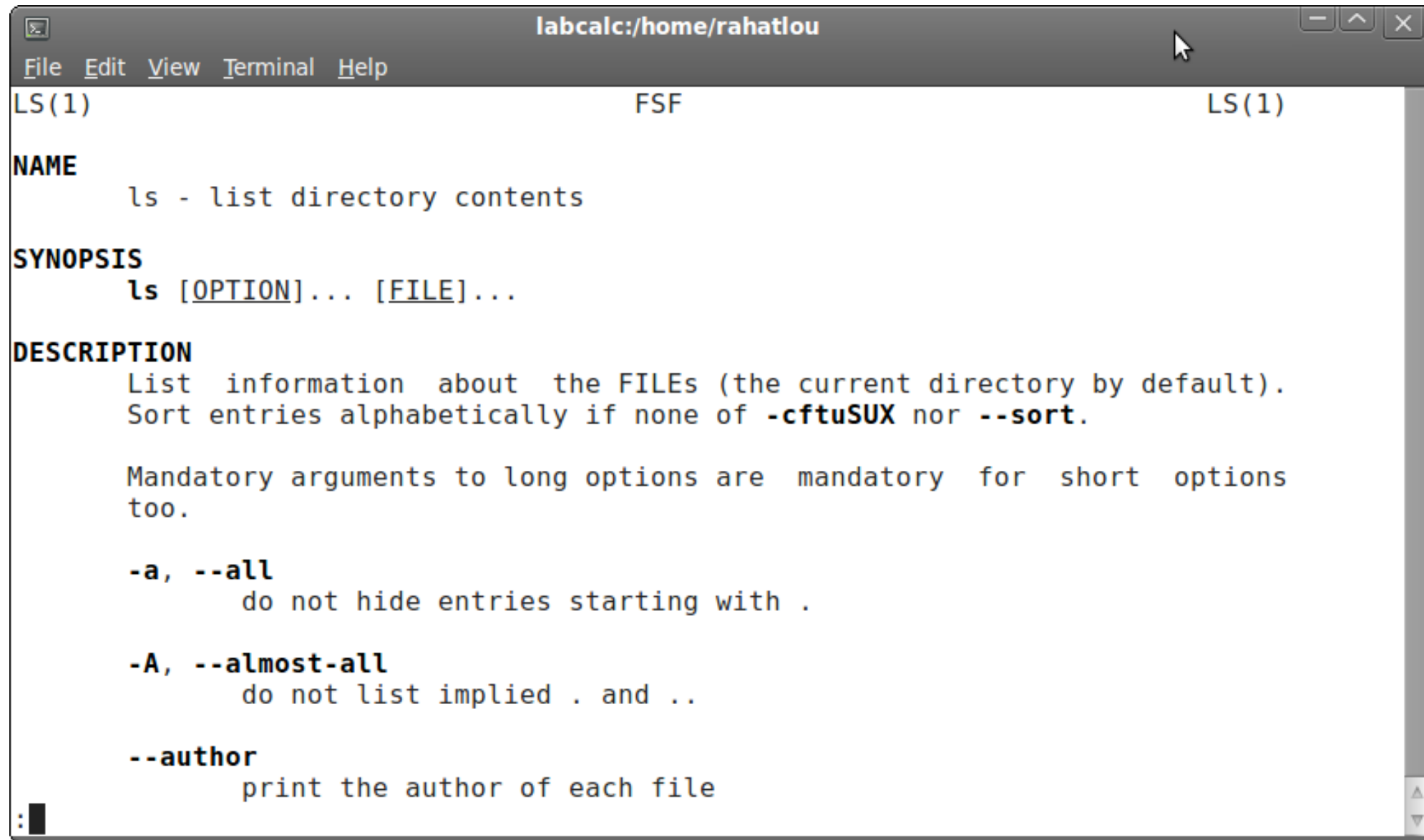
# Opzioni utili di ls

```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ ls -ltr
total 124
-rwxr-xr-x  1 ppp30  studenti    453 Apr 29 00:46 data.txt
-rwxr-xr-x  1 ppp30  studenti     63 May  9 18:33 data2.txt
-rwxr-xr-x  1 ppp30  studenti 43049 May 12 15:24 main
-rw-r--r--  1 ppp30  studenti 13066 May 12 15:29 WeightedAverage.pdf
-rw-r--r--  1 ppp30  studenti  5430 Jun  9 09:27 teach.ico
-rw-r--r--  1 ppp30  studenti   54 Oct  1 12:05 test.txt~
-rw-r--r--  1 ppp30  studenti  116 Oct  1 12:26 test.c~
-rw-r--r--  1 ppp30  studenti  118 Oct  1 12:27 test.c
-rw-r--r--  1 ppp30  studenti  912 Oct  1 16:08 test.o
-rwxr-xr-x  1 ppp30  studenti 4770 Oct  1 16:10 mario
-rw-r--r--  1 ppp30  studenti  706 Oct  1 16:24 prova.o
-rw-r--r--  1 ppp30  studenti  156 Oct  1 16:21 prova.c~
-rw-r--r--  1 ppp30  studenti  147 Oct  1 16:33 prova.c
-rwxr-xr-x  1 ppp30  studenti 4803 Oct  1 16:33 a.out
drwxr-xr-x  3 ppp30  studenti 4096 Oct  1 16:46 Desktop
[ppp30@labcalc ~]$
```

## ■ Ci sono diverse opzioni di ls per aumentare il livello di informazione o ordinare l'output

- **-l**: informazioni dettagliate come l'ora e la data della modifica ed i permessi
- **-t**: ordina i file secondo data della modifica. Primi file più recenti
- **-r**: inverte l'ordine di **-t**; primi i file più vecchi

# Trovare altre opzioni con **man**



The screenshot shows a terminal window titled 'labcalc:/home/rahatlou'. The window displays the manual page for the 'ls' command. The content is as follows:

```
LS(1)                                FSF                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuSUX nor --sort.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not hide entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        print the author of each file
```

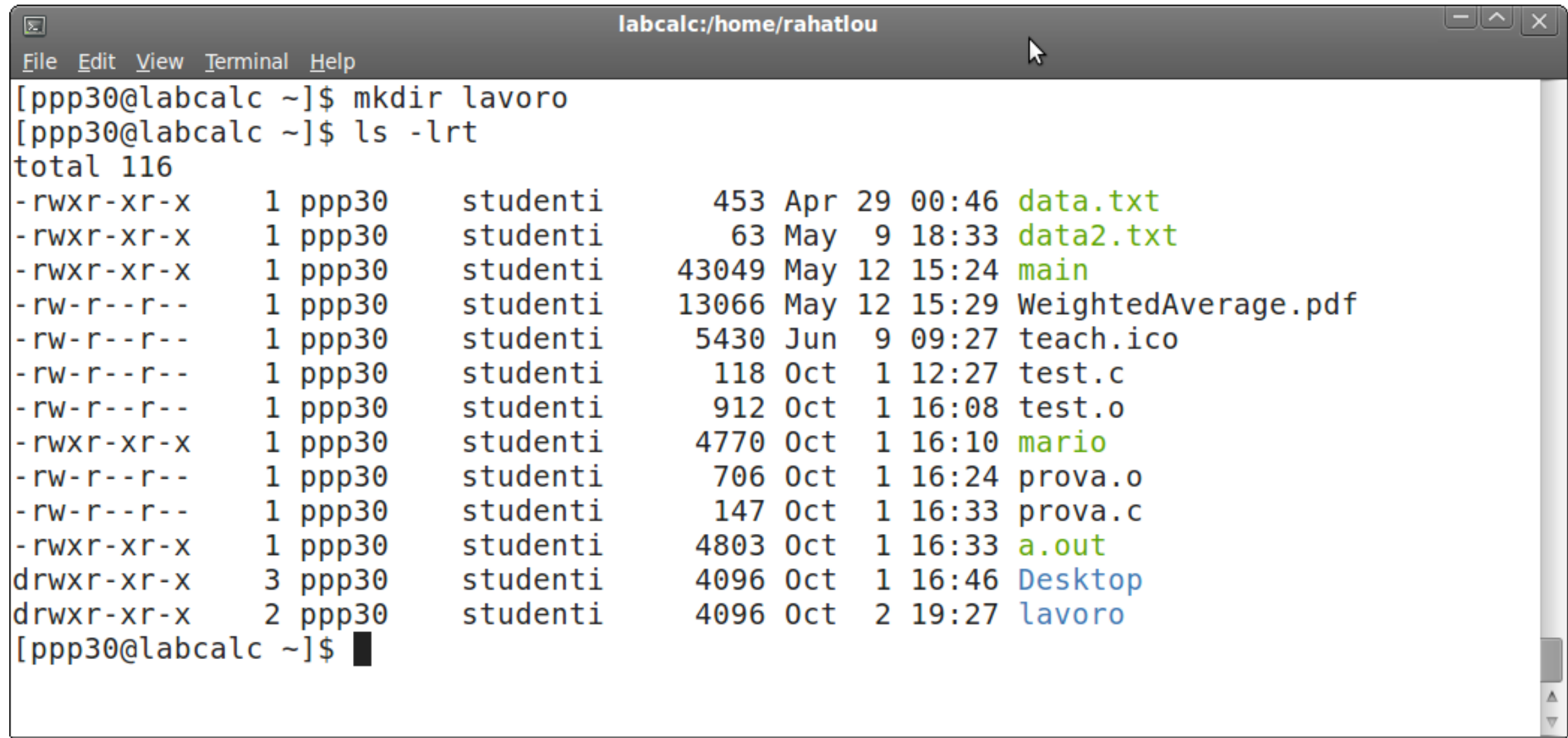
- Per tutti i comandi di Linux potete accedere al manuale con il comando **man** ad esempio

**man ls**

risultata nella schermata di sopra

- Per uscire dal manuale premere il tasto **q**

# Creare cartelle con `mkdir`



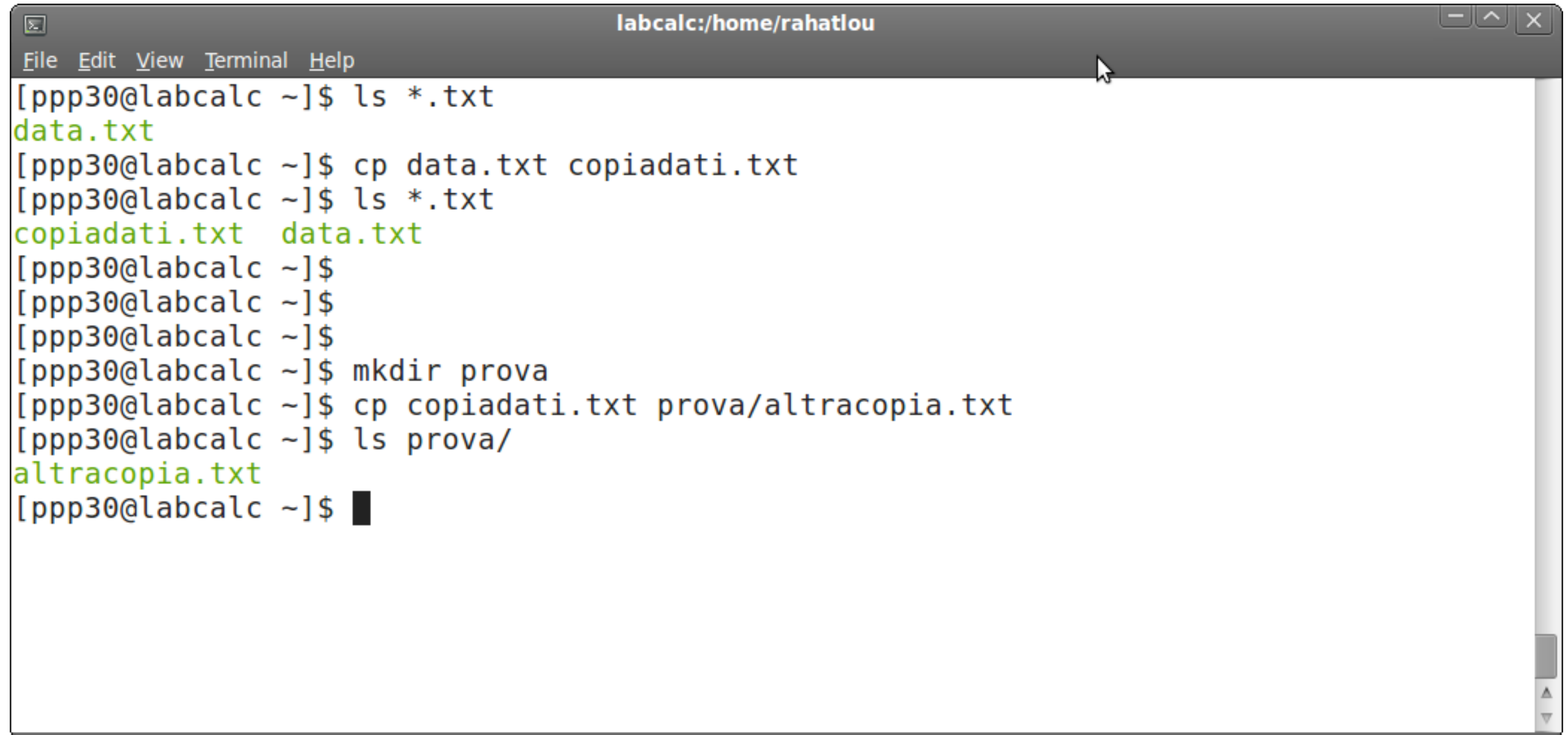
A terminal window titled 'labcalc:/home/rahatlou' showing the execution of the `mkdir` command and the subsequent `ls -lrt` command. The output of `ls -lrt` lists the contents of the home directory, including files like `data.txt`, `data2.txt`, `main`, `WeightedAverage.pdf`, `teach.ico`, `test.c`, `test.o`, `mario`, `prova.o`, `prova.c`, `a.out`, `Desktop`, and the newly created directory `lavoro`.

```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ mkdir lavoro
[ppp30@labcalc ~]$ ls -lrt
total 116
-rwxr-xr-x  1 ppp30  studenti    453 Apr 29 00:46 data.txt
-rwxr-xr-x  1 ppp30  studenti     63 May  9 18:33 data2.txt
-rwxr-xr-x  1 ppp30  studenti 43049 May 12 15:24 main
-rw-r--r--  1 ppp30  studenti 13066 May 12 15:29 WeightedAverage.pdf
-rw-r--r--  1 ppp30  studenti  5430 Jun  9 09:27 teach.ico
-rw-r--r--  1 ppp30  studenti   118 Oct  1 12:27 test.c
-rw-r--r--  1 ppp30  studenti   912 Oct  1 16:08 test.o
-rwxr-xr-x  1 ppp30  studenti  4770 Oct  1 16:10 mario
-rw-r--r--  1 ppp30  studenti   706 Oct  1 16:24 prova.o
-rw-r--r--  1 ppp30  studenti   147 Oct  1 16:33 prova.c
-rwxr-xr-x  1 ppp30  studenti  4803 Oct  1 16:33 a.out
drwxr-xr-x  3 ppp30  studenti  4096 Oct  1 16:46 Desktop
drwxr-xr-x  2 ppp30  studenti  4096 Oct  2 19:27 lavoro
[ppp30@labcalc ~]$
```

- Potete creare cartelle (o sotto-cartelle) con un nome

# Copiare i file con `cp`

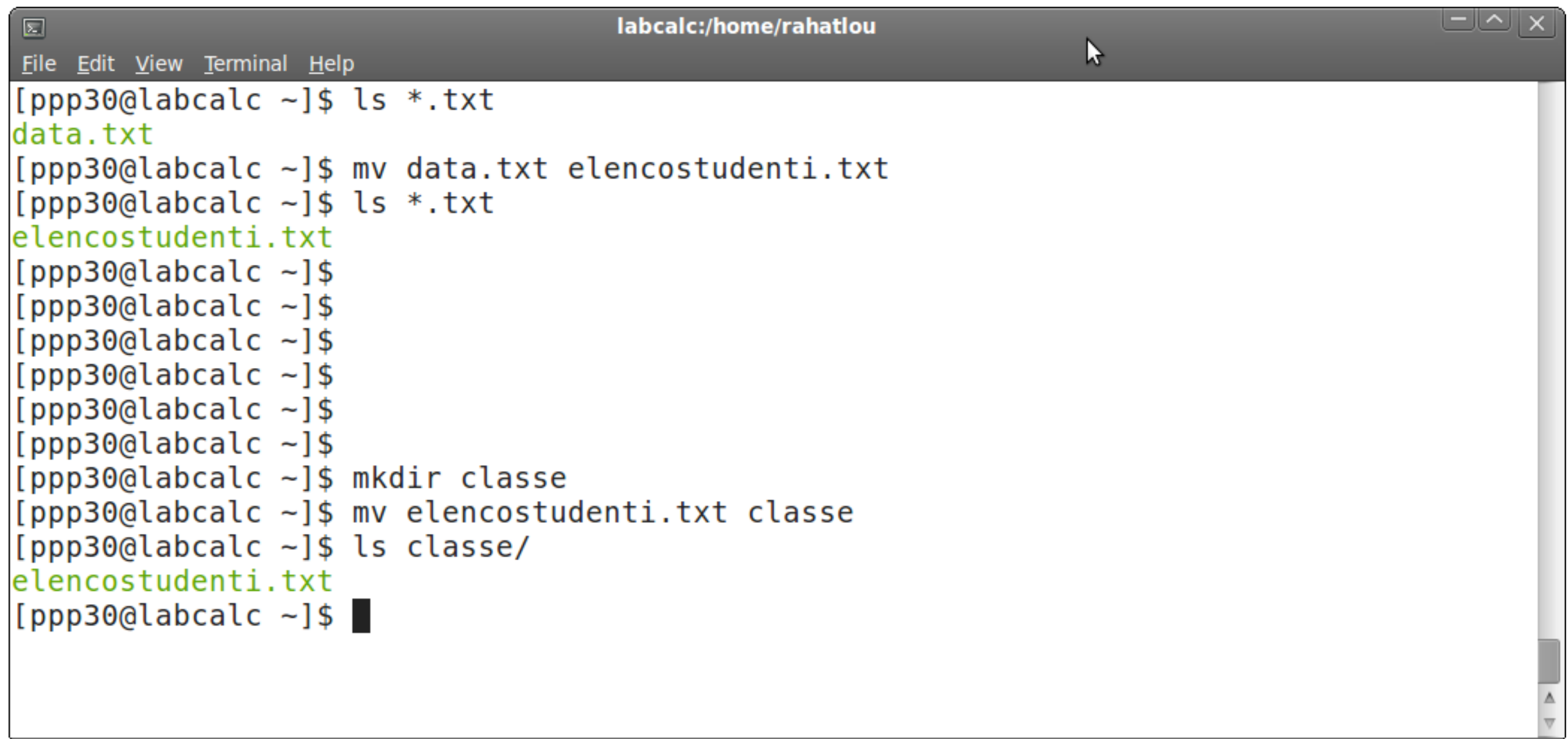
- Potete creare una copia di un file esistente nella cartella in cui vi trovate oppure in una nuova cartella

A screenshot of a terminal window titled 'labcalc:/home/rahatlou'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. The terminal shows a series of commands and their outputs. First, the user runs 'ls \*.txt' and sees 'data.txt'. Then, they run 'cp data.txt copiadati.txt'. Next, they run 'ls \*.txt' again and see 'copiadati.txt' and 'data.txt'. After that, they run 'mkdir prova'. Then, they run 'cp copiadati.txt prova/altracopia.txt'. Finally, they run 'ls prova/' and see 'altracopia.txt'. The prompt is '[ppp30@labcalc ~]\$' and there is a cursor at the end of the last line.

```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ ls *.txt
data.txt
[ppp30@labcalc ~]$ cp data.txt copiadati.txt
[ppp30@labcalc ~]$ ls *.txt
copiadati.txt data.txt
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$ mkdir prova
[ppp30@labcalc ~]$ cp copiadati.txt prova/altracopia.txt
[ppp30@labcalc ~]$ ls prova/
altracopia.txt
[ppp30@labcalc ~]$
```

# Spostare o rinominare i file con `mv`

- Si usa un unico comando per
  - Rinominare un file
  - Spostare un file da una directory in un'altra

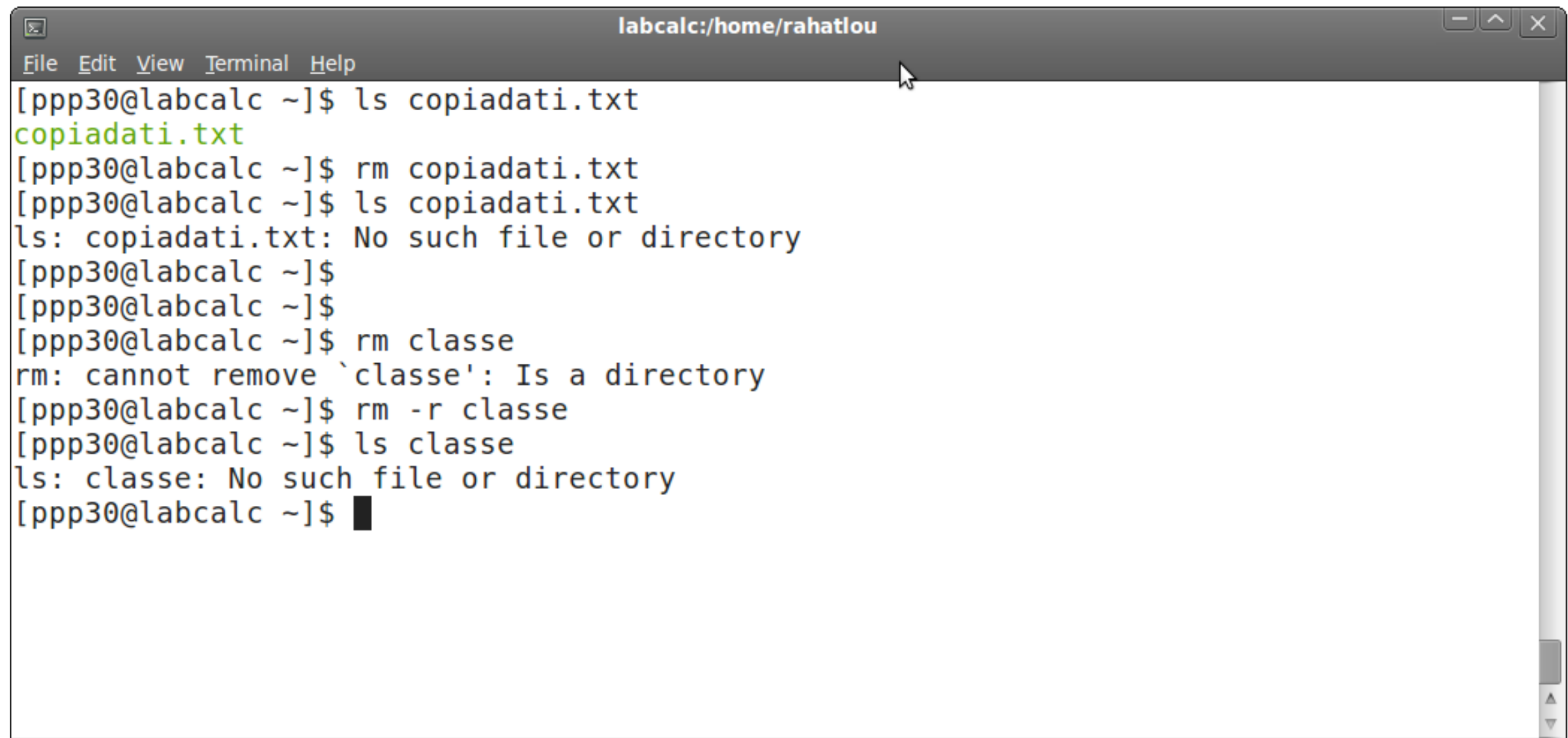
A terminal window titled 'labcalc:/home/rahatlou' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows a series of commands and their outputs. The user lists files, renames 'data.txt' to 'elencostudenti.txt', lists files again, creates a directory 'classe', moves 'elencostudenti.txt' to 'classe', and lists the contents of 'classe/'.

```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ ls *.txt
data.txt
[ppp30@labcalc ~]$ mv data.txt elencostudenti.txt
[ppp30@labcalc ~]$ ls *.txt
elencostudenti.txt
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$ mkdir classe
[ppp30@labcalc ~]$ mv elencostudenti.txt classe
[ppp30@labcalc ~]$ ls classe/
elencostudenti.txt
[ppp30@labcalc ~]$
```



# Cancellare (senza poterci ripensare) con `rm`

- Ricordatevi che una volta rimosso un file o una cartella non c'è modo di recuperarli
- Rimuovere una cartella rimuove anche tutti i file che contiene



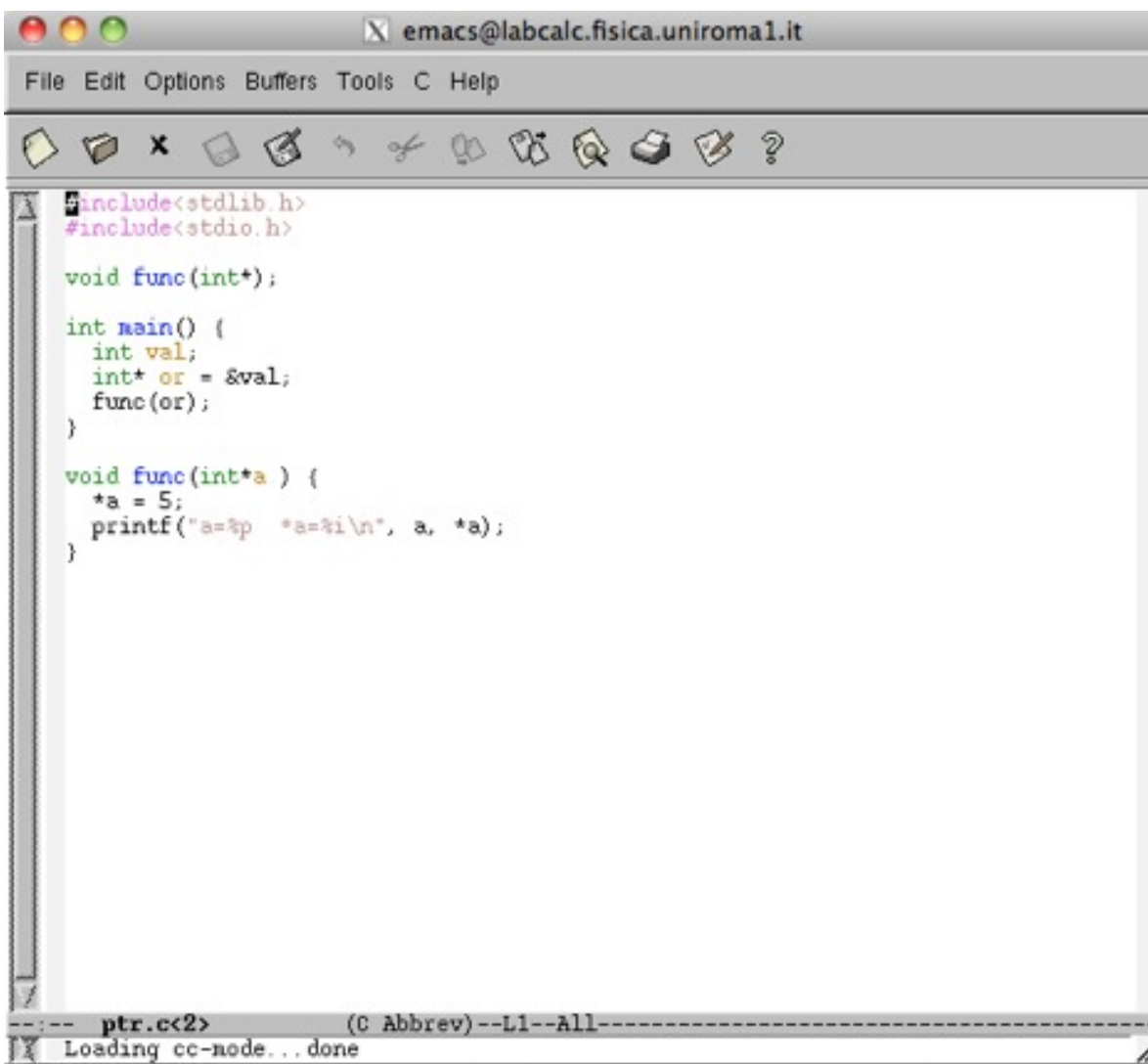
```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ ls copiadati.txt
copiadati.txt
[ppp30@labcalc ~]$ rm copiadati.txt
[ppp30@labcalc ~]$ ls copiadati.txt
ls: copiadati.txt: No such file or directory
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$ rm classe
rm: cannot remove `classe': Is a directory
[ppp30@labcalc ~]$ rm -r classe
[ppp30@labcalc ~]$ ls classe
ls: classe: No such file or directory
[ppp30@labcalc ~]$
```

# Editor di Testo

- Qualsiasi applicazione per scrivere testo senza formattazione
- Da non confondere con “Word Processor” come Microsoft Word o OpenOffice
  - Oltre al testo queste applicazioni aggiungono informazioni di formattazione quali: font, dimensione, indentatura, impaginazione
- Useremo l'applicazione gratuita EMACS nelle nostre sessioni di laboratorio
- Altri editor molto usati dagli studenti in passato
  - Nedit
  - Kedit
  - Gedit
  - Kate

# Esmepi: Emacs e gedit

- Editor che useremo per questo corso

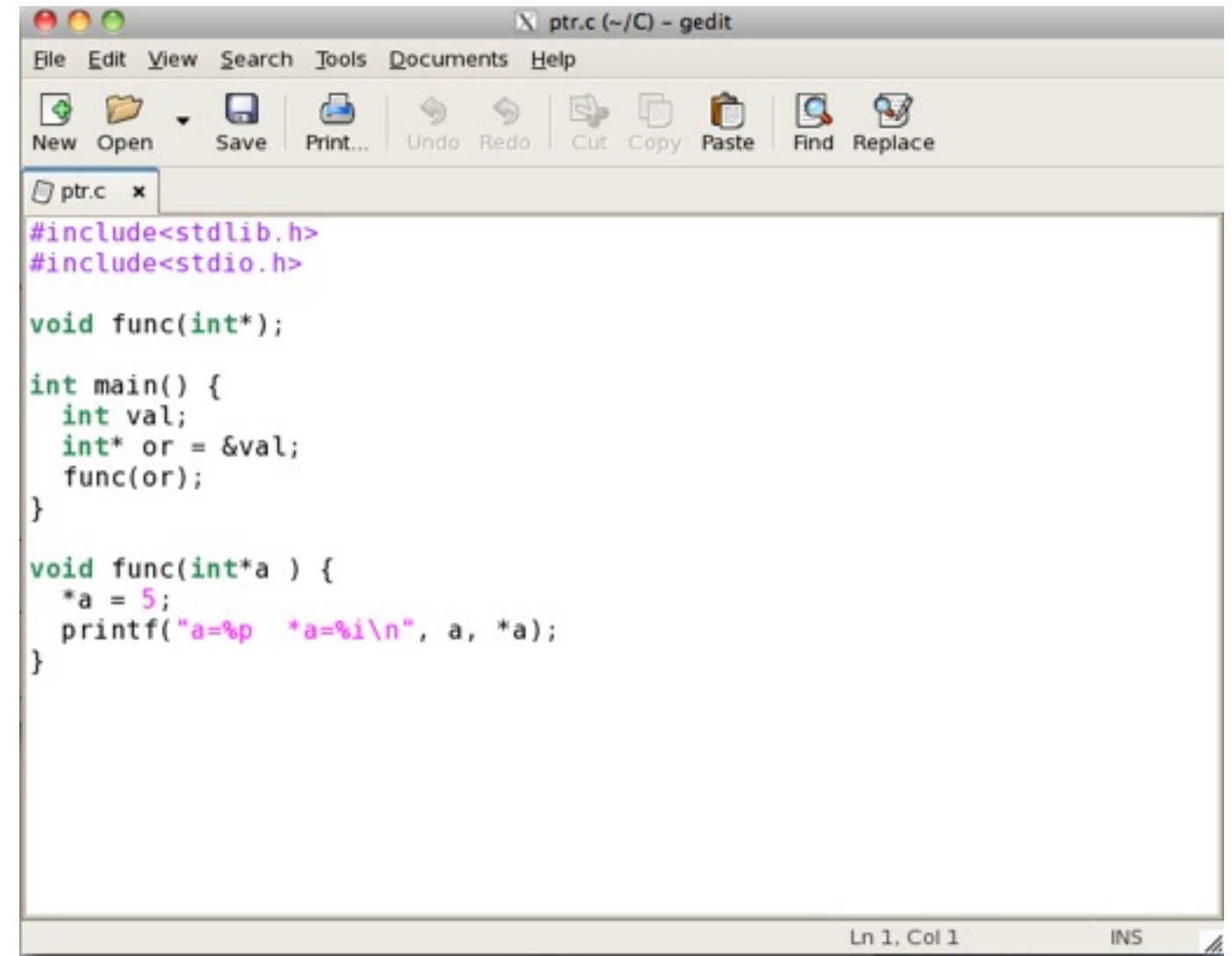


```
#include<stdlib.h>
#include<stdio.h>

void func(int*);

int main() {
    int val;
    int* or = &val;
    func(or);
}

void func(int*a ) {
    *a = 5;
    printf("a=%p *a=%i\n", a, *a);
}
```



```
#include<stdlib.h>
#include<stdio.h>

void func(int*);

int main() {
    int val;
    int* or = &val;
    func(or);
}

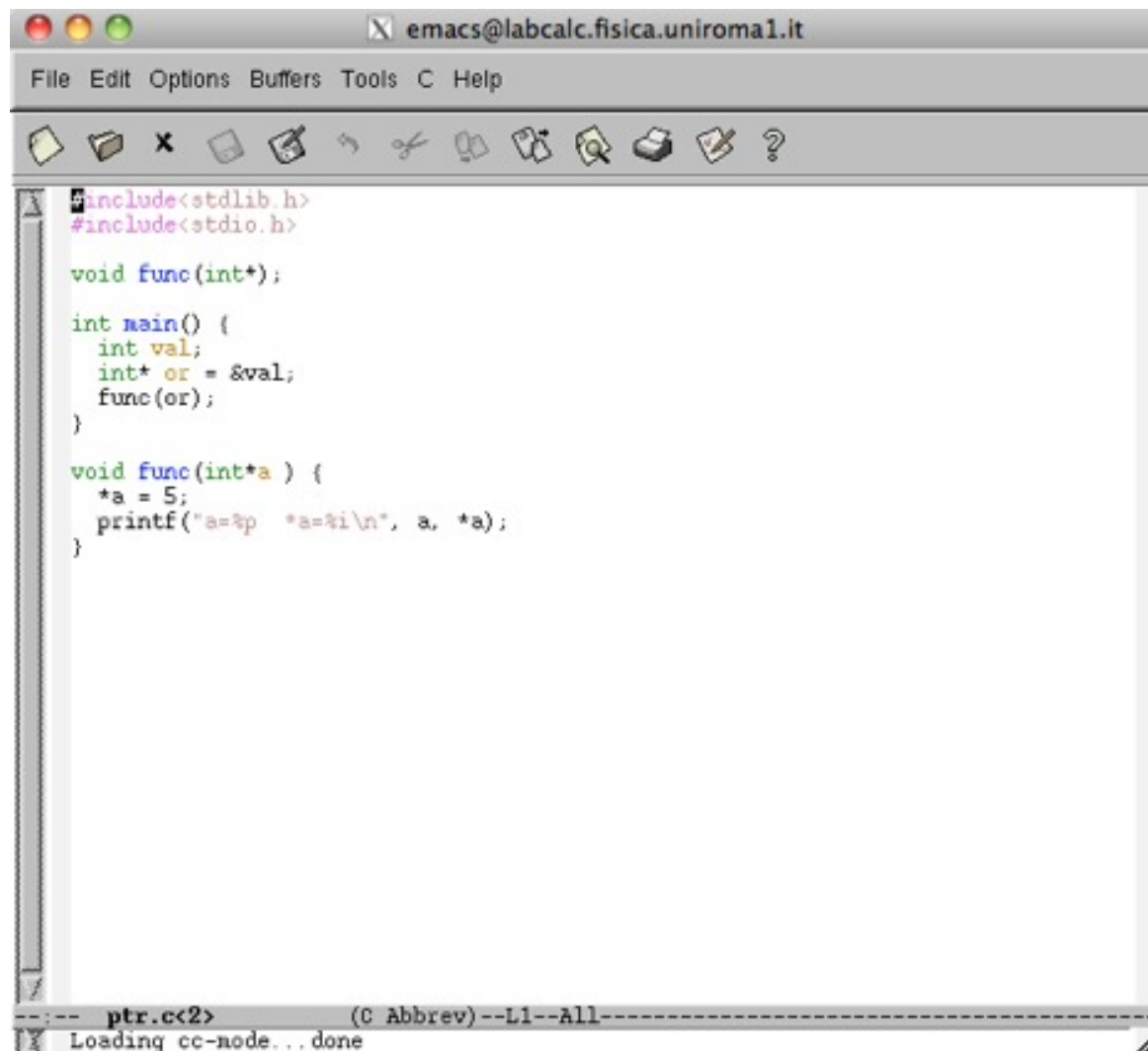
void func(int*a ) {
    *a = 5;
    printf("a=%p *a=%i\n", a, *a);
}
```

- Per invocarlo dalla riga di comando

```
prompt> emacs nomefile.c
```

# Colori nell'editor di testo

- Quasi tutti gli editor usano i colori per evidenziare la sintassi dei linguaggi di programmazione
  - Solo di utilita` visuale per l'utente e per riconoscere piu` facilmente errori comuni come
    - parentesi aperte ma non chiuse: viene evidenziata la parentesi aperta
    - testo commentato: di colore diverso rispetto al testo che verra` compilato
    - colore diverso per le parole chiave del linguaggio
- **main, open, printf, for, exit**
- I compilatori non capiscono e non usano i colori!
  - abituatevi a non dipendere unicamente dai colori per capire il codice



The screenshot shows the Emacs editor window titled 'emacs@labcalc.fisica.uniroma1.it'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'C', and 'Help'. The toolbar contains various icons for file operations and editing. The code in the buffer is C code with syntax highlighting: preprocessor directives are purple, keywords are green, identifiers are black, and string literals are red. The code is as follows:

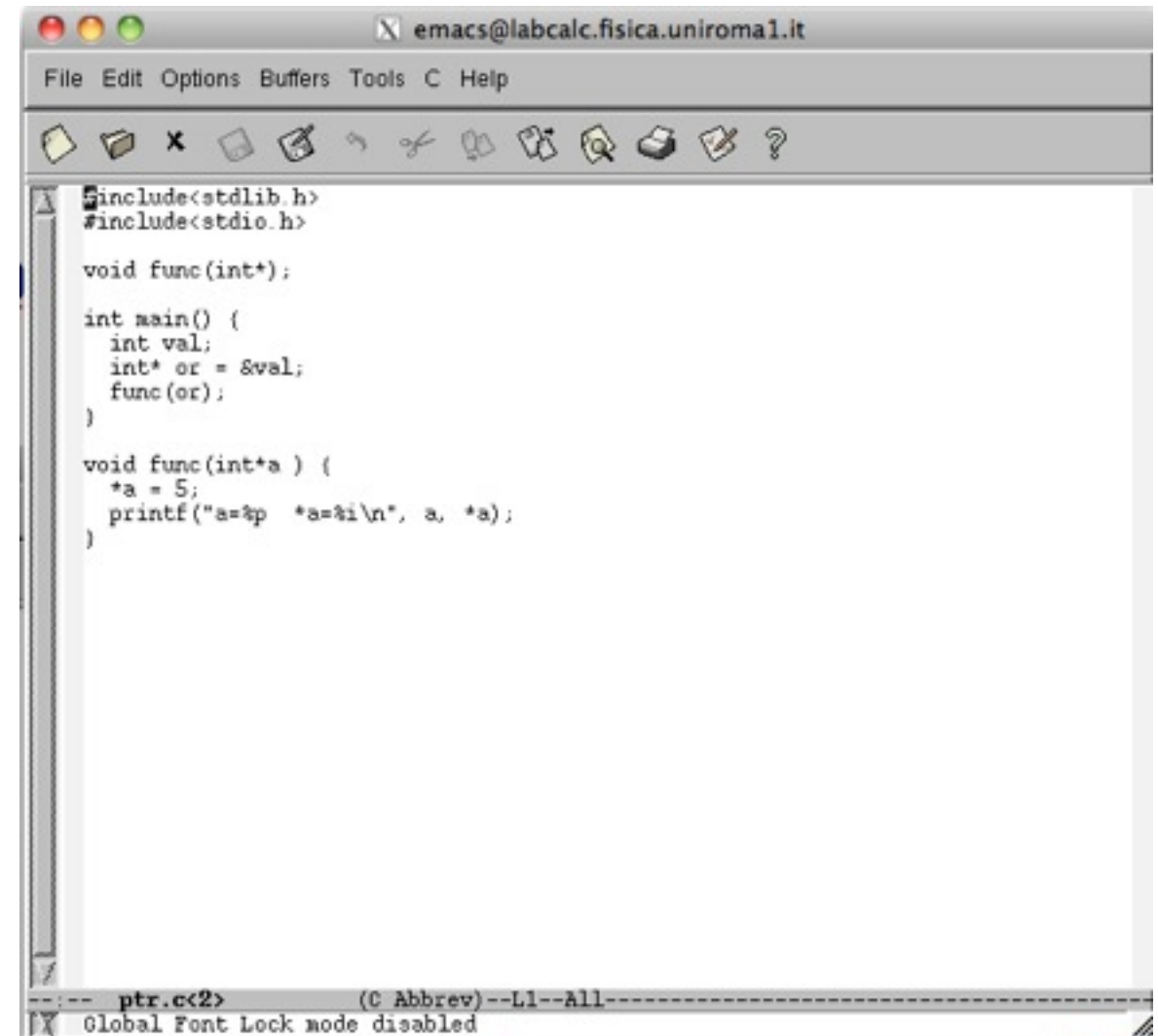
```
#include<stdlib.h>
#include<stdio.h>

void func(int*);

int main() {
    int val;
    int* or = &val;
    func(or);
}

void func(int*a ) {
    *a = 5;
    printf("a=%p  *a=%i\n", a, *a);
}
```

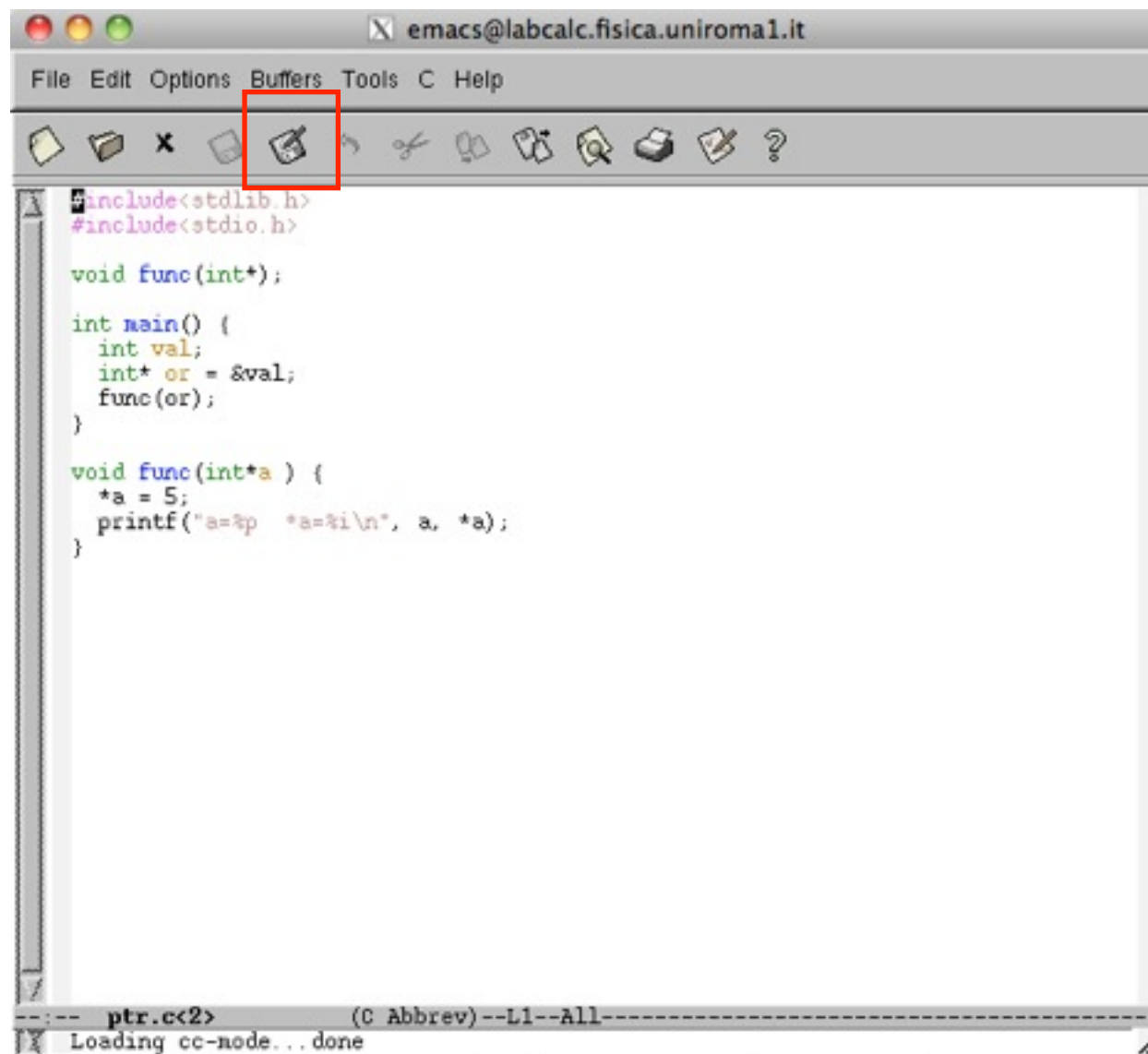
The status bar at the bottom shows 'ptr.c<2>' and '(C Abbrev)--L1--All--'. The message line at the bottom says 'Loading cc-node... done'.



The screenshot shows the same Emacs editor window with the same C code, but without syntax highlighting. All text is in black. The status bar at the bottom shows 'ptr.c<2>' and '(C Abbrev)--L1--All--'. The message line at the bottom says 'Global Font Lock mode disabled'.

# Salvare un nuovo file

- Un nuovo file si trova inizialmente solo nella memoria
  - si dice nel buffer dell'editor di testo
- Se non salvato sul disco il testo scritto verra` perso chiudendo l'editor
- Abituatevi a salvare regolarmente il file per non perdere le ultime modifiche



emacs@labcalc.fisica.uniroma1.it

File Edit Options Buffers Tools C Help

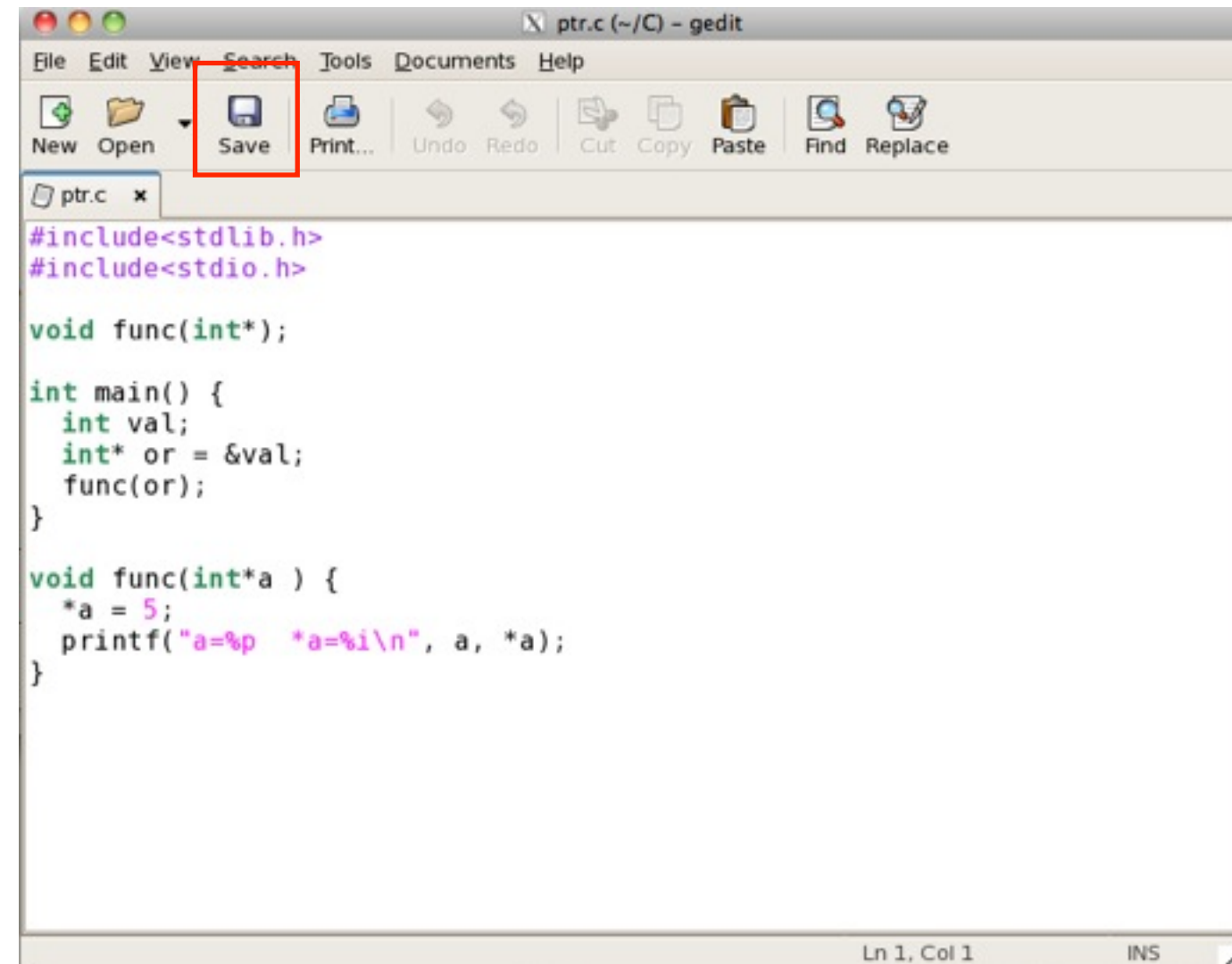
```
#include<stdlib.h>
#include<stdio.h>

void func(int*);

int main() {
    int val;
    int* or = &val;
    func(or);
}

void func(int*a ) {
    *a = 5;
    printf("a=%p  *a=%i\n", a, *a);
}
```

ptr.c<2> (C Abbrev)--L1--All-- Loading cc-node...done



ptr.c (~ / C) - gedit

File Edit View Search Tools Documents Help

New Open Save Print... Undo Redo Cut Copy Paste Find Replace

```
#include<stdlib.h>
#include<stdio.h>

void func(int*);

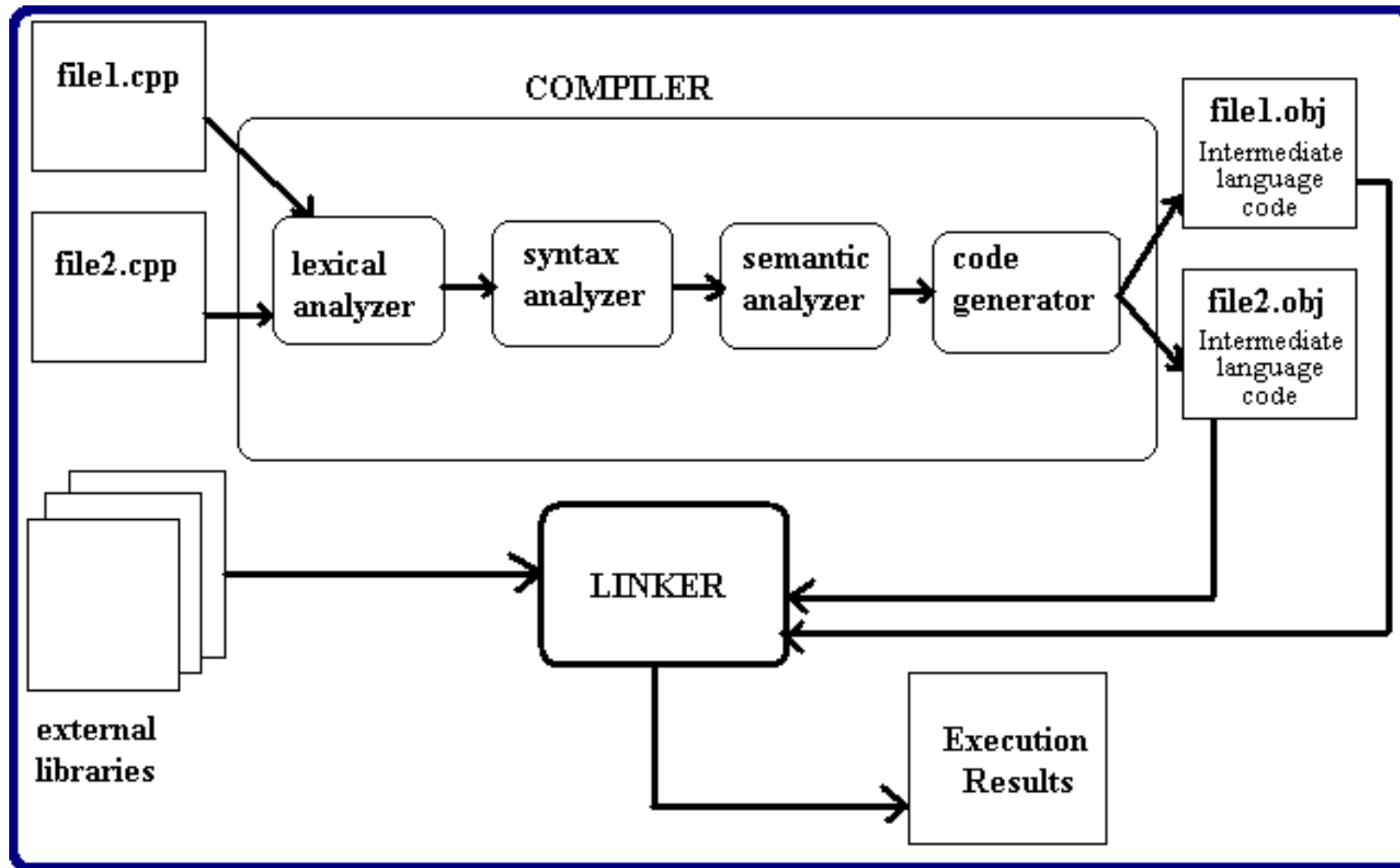
int main() {
    int val;
    int* or = &val;
    func(or);
}

void func(int*a ) {
    *a = 5;
    printf("a=%p  *a=%i\n", a, *a);
}
```

Ln 1, Col 1 INS



# Compilare e 'linkare' eseguibili



- **Compilare:** tradurre istruzioni di linguaggi ad alto livello (C/C++/Java) in istruzioni macchina
- **Linkare:** mettere insieme o collegare diversi pezzi di istruzioni binarie per creare un unico programma eseguibile

# Compilazione

- Useremo gcc durante il corso ma potete usare anche altri compilatori
- Per compilare senza linkare (ossia senza creare l'eseguibile)
  - Traduce solo le istruzioni dal linguaggio C in linguaggio macchina

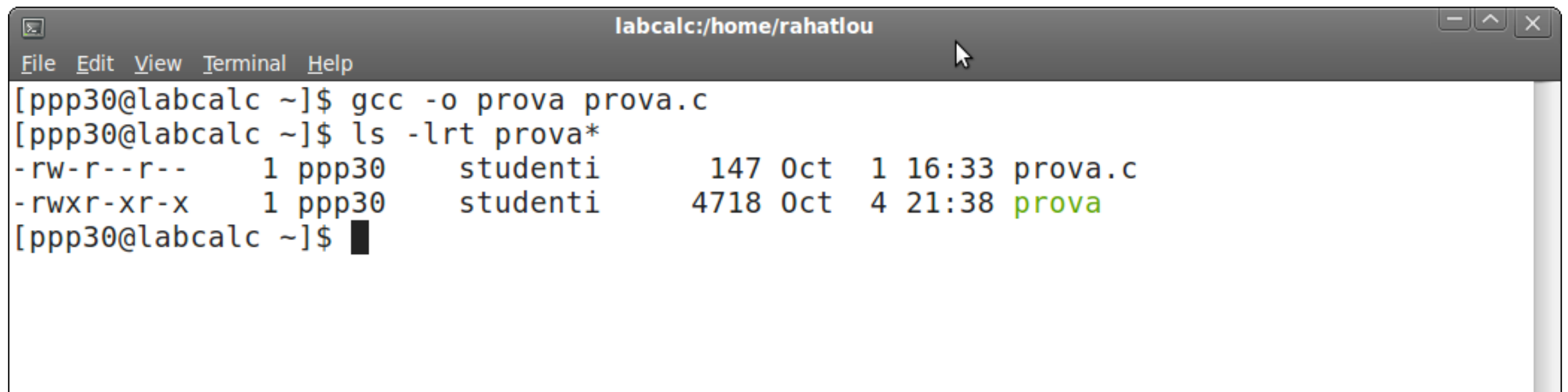


```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ gcc -c prova.c
[ppp30@labcalc ~]$ ls prova.*
prova.c  prova.o
[ppp30@labcalc ~]$
```

- Per ciascun file di sorgente (estensione .c) il compilatore genera un object file (estensione .o)

# Compilare e linkare l'eseguibile allo stesso tempo

- L'opzione `-o` specifica il nome dell'eseguibile
  - Altrimenti di solito il linker lo chiamerà `a.out` oppure `a.exe`
  - Non si usa l'opzione `-c` proprio per invocare il linker dopo la compilazione
- Si può fare l'eseguibile solo per un file `.c` che contenga una funzione `main`



```
labcalc:/home/rahatlou
File Edit View Terminal Help
[ppp30@labcalc ~]$ gcc -o prova prova.c
[ppp30@labcalc ~]$ ls -lrt prova*
-rw-r--r--  1 ppp30  studenti    147 Oct  1 16:33 prova.c
-rwxr-xr-x  1 ppp30  studenti   4718 Oct  4 21:38 prova
[ppp30@labcalc ~]$
```

# Sommario uso compilatore

Comando	Commento
<code>gcc -Wall -c prova.c</code>	compila il file sorgente <code>prova.c</code> e crea il file oggetto <code>prova.o</code> Non c'è nessun eseguibile
<code>gcc -Wall -o app.exe prova.c</code>	compila e linka il file sorgente <code>prova.c</code> e crea il file eseguibile <code>app.exe</code> Non c'è nessun <code>file.o</code> in questo caso
<code>gcc -Wall -lm -o app.exe prova.c</code>	compila e linka il file sorgente <code>prova.c</code> utilizzando la libreria matematica (opzione <code>-lm</code> ) e crea il file eseguibile <code>app.exe</code> Non c'è nessun <code>file.o</code> in questo caso
<code>gcc -Wall -o app.exe prova.c grafica.o</code>	compila il file sorgente <code>prova.c</code> e crea il file eseguibile <code>app.exe</code> utilizzando anche il contenuto del file <code>grafica.o</code> (compilato in precedenza)
<code>gcc -Wall -o app.exe prova.c grafica.c</code>	compila i due file sorgente <code>prova.c</code> e <code>grafica.c</code> e crea il file eseguibile <code>app.exe</code> Non viene creato nessun file oggetto

# GIRARE IL PROGRAMMA ESEGUIBILE

- Bisogna trovarsi nella cartella in cui si trova l'eseguibile

```
labcalc.fisica.uniroma1.it:/home/docenti/rahatlou/C/test — ssh — 85×26
$ ls
prova.c
$
$
$ gcc -Wall -lm -o app.exe prova.c
$
$
$ ls
app.exe*  prova.c
$
$ ./app.exe
a=0x7fffb247cb84  *a=5
$
$
$ app.exe
app.exe: Command not found.
$
```

mettere ./ prima del nome del programma eseguibile e` fondamentale!

senza mettere ./ ottenete questo errore