

NOZIONI BASE PER ESERCITAZIONI

DIPARTIMENTO DI FISICA



SAPIENZA
UNIVERSITÀ DI ROMA

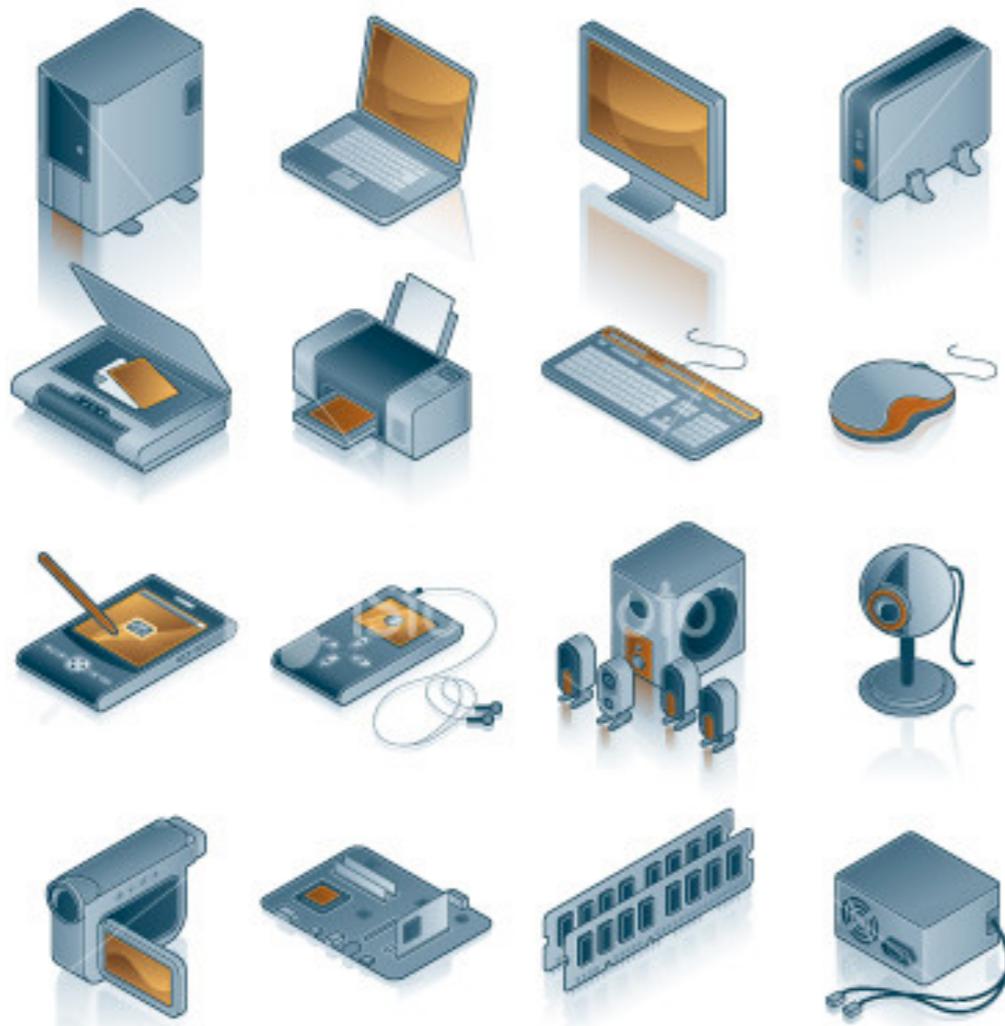
Shahram Rahatlou

Laboratorio di Calcolo, Anno Accademico 2025-2026

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

Sistema Operativo

Hardware

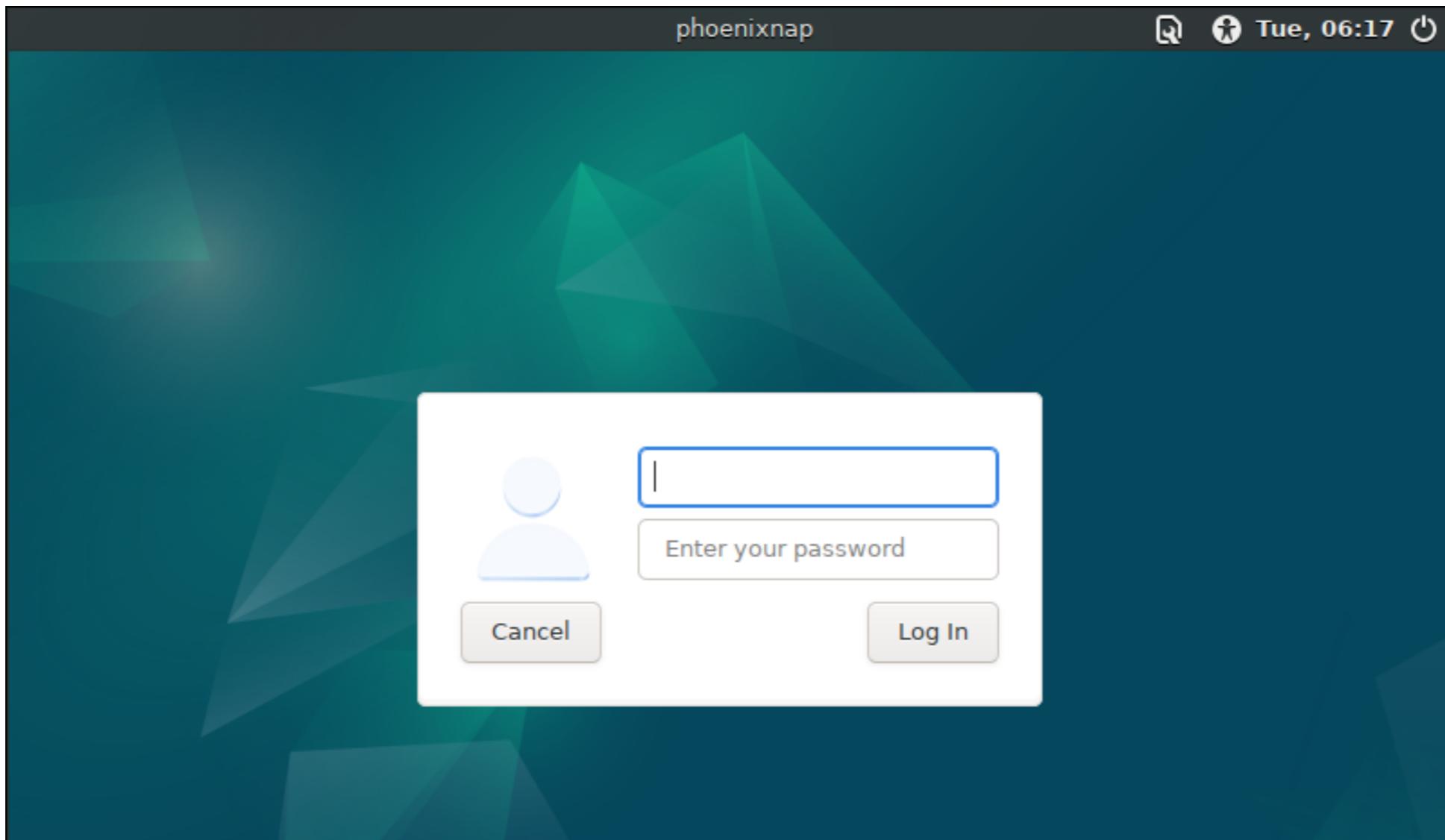


Software



- 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

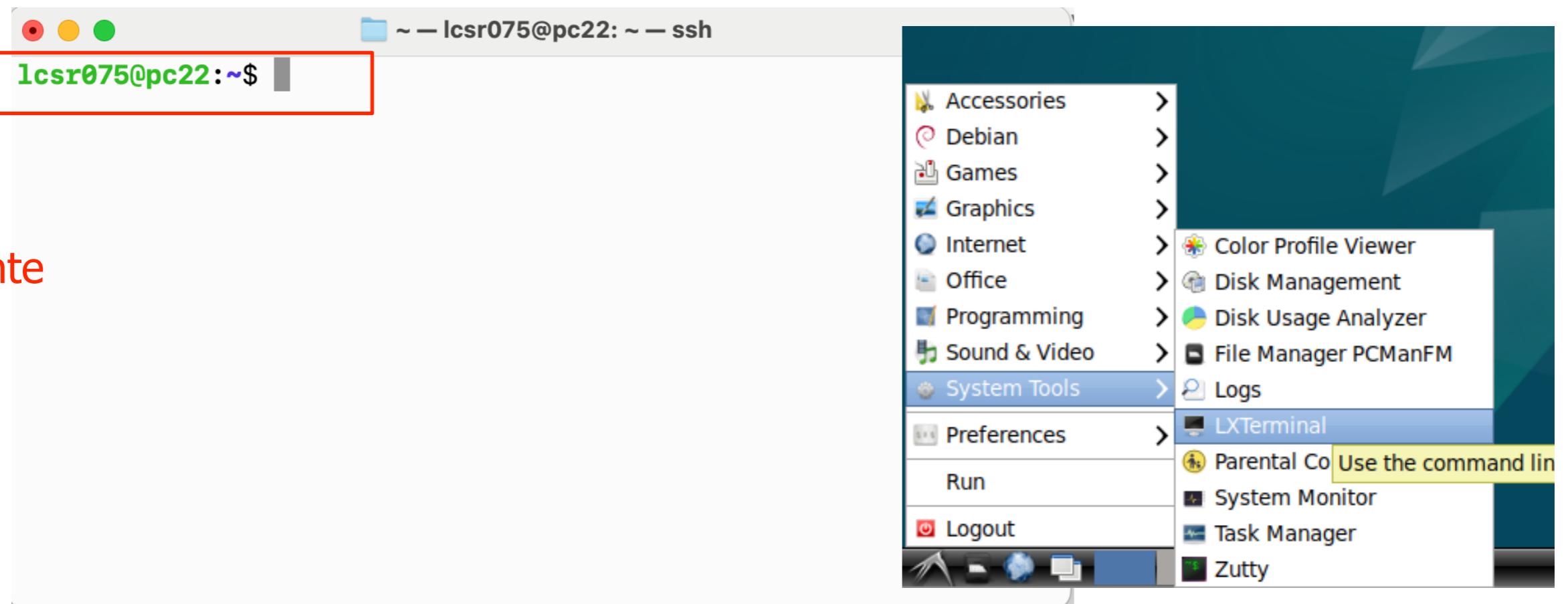
Accesso al sistema (Login)



- Per l'accesso servono uno **username** ed una **password**
- A ciascun gruppo e` assegnato uno username del tipo **1csrXXX** dove **XXX=001,002,...,099**

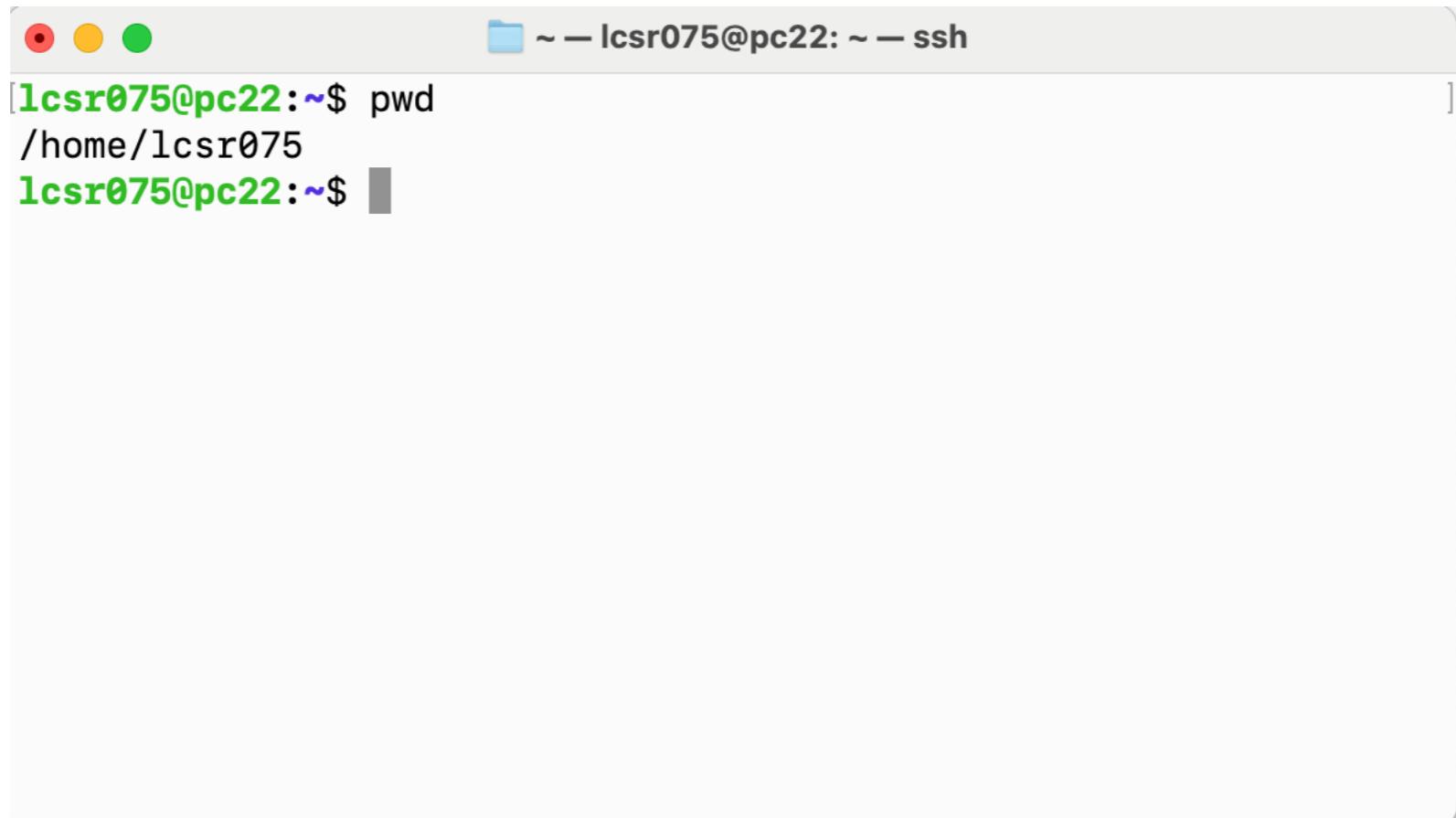
Terminale e Shell

Prompt
dell'utente



- L'utente puo` impartire comandi al sistema utilizzando la **shell**
- La **shell** e` accessibile all'interno di una finestra di riga di comando chiamato comunemente il **terminale**

Home Directory

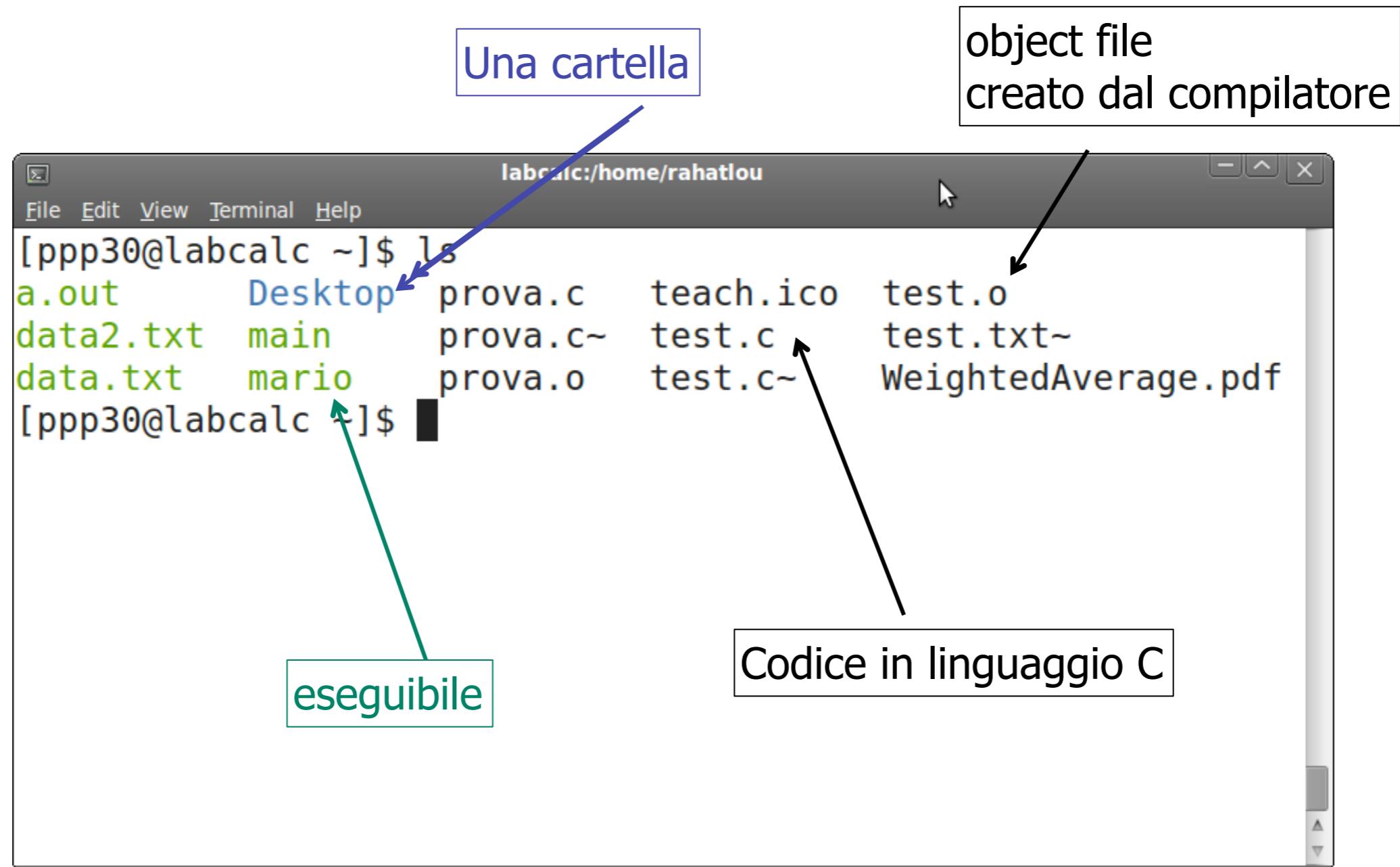


A screenshot of a terminal window on a Mac OS X system. The window title bar says "lcsr075@pc22: ~ -- ssh". The terminal prompt is "lcsr075@pc22:~\$". The user has run the command "pwd" which outputs the path "/home/lcsr075". The terminal window has a light gray background and a dark gray sidebar on the right.

```
[lcsr075@pc22:~$ pwd  
/home/lcsr075  
lcsr075@pc22:~$ ]
```

- 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`



- 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 e` 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!

Opzioni utili di ls

```
[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 16:31 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`

```
LS(1)                               User Commands                               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 -cftuvSUX nor --sort is specified.

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

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

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

--author
with -l, print the author of each file

-b, --escape
Manual page ls(1) line 1 (press h for help or q to quit)
```

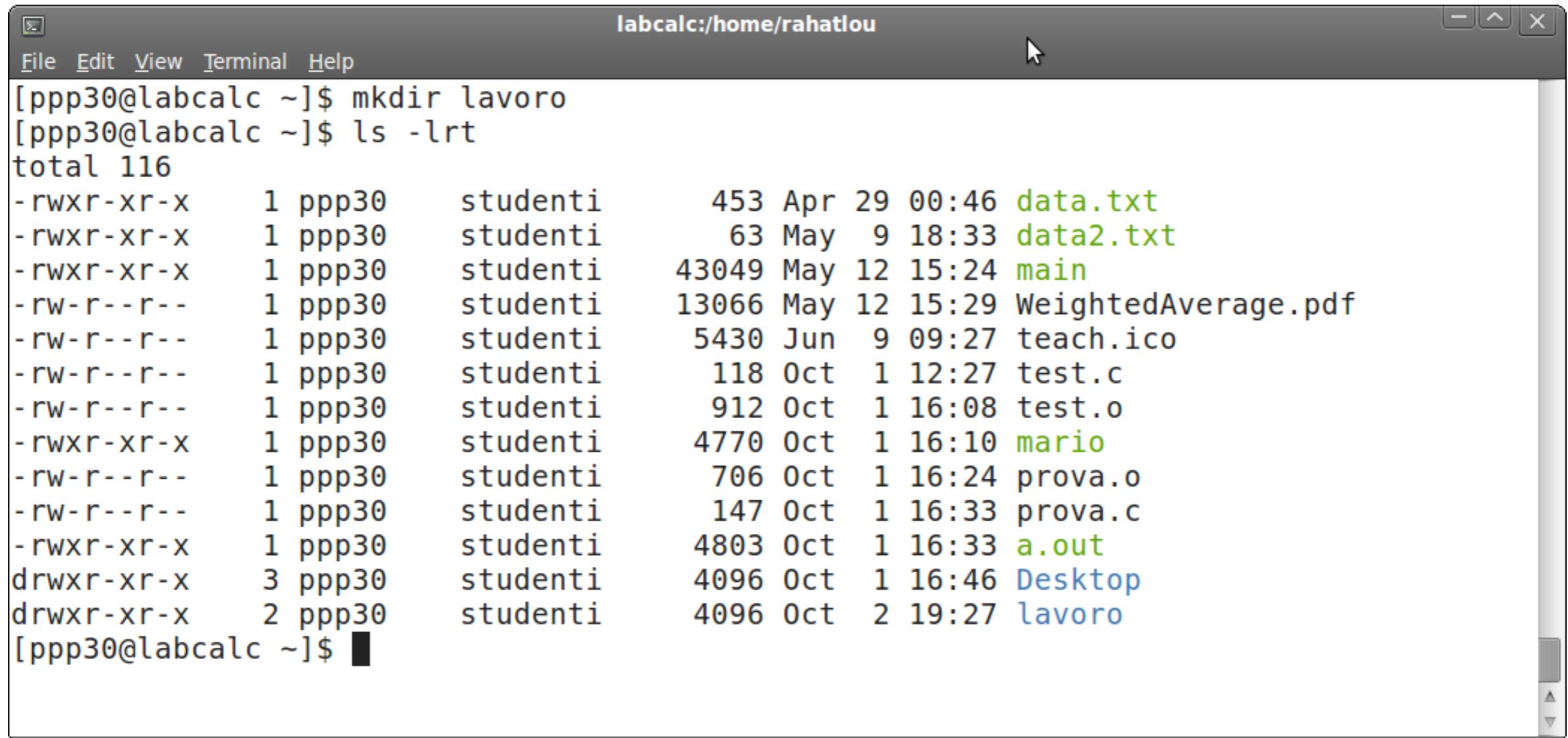
- Per tutti i comandi di Linux potete accedere al manuale con il comando `man` ad esempio

`man ls`

risultata nella schermata di sopra

- Per uscire dal manuale premere il tasto `q`

Creare cartelle con `mkdir`



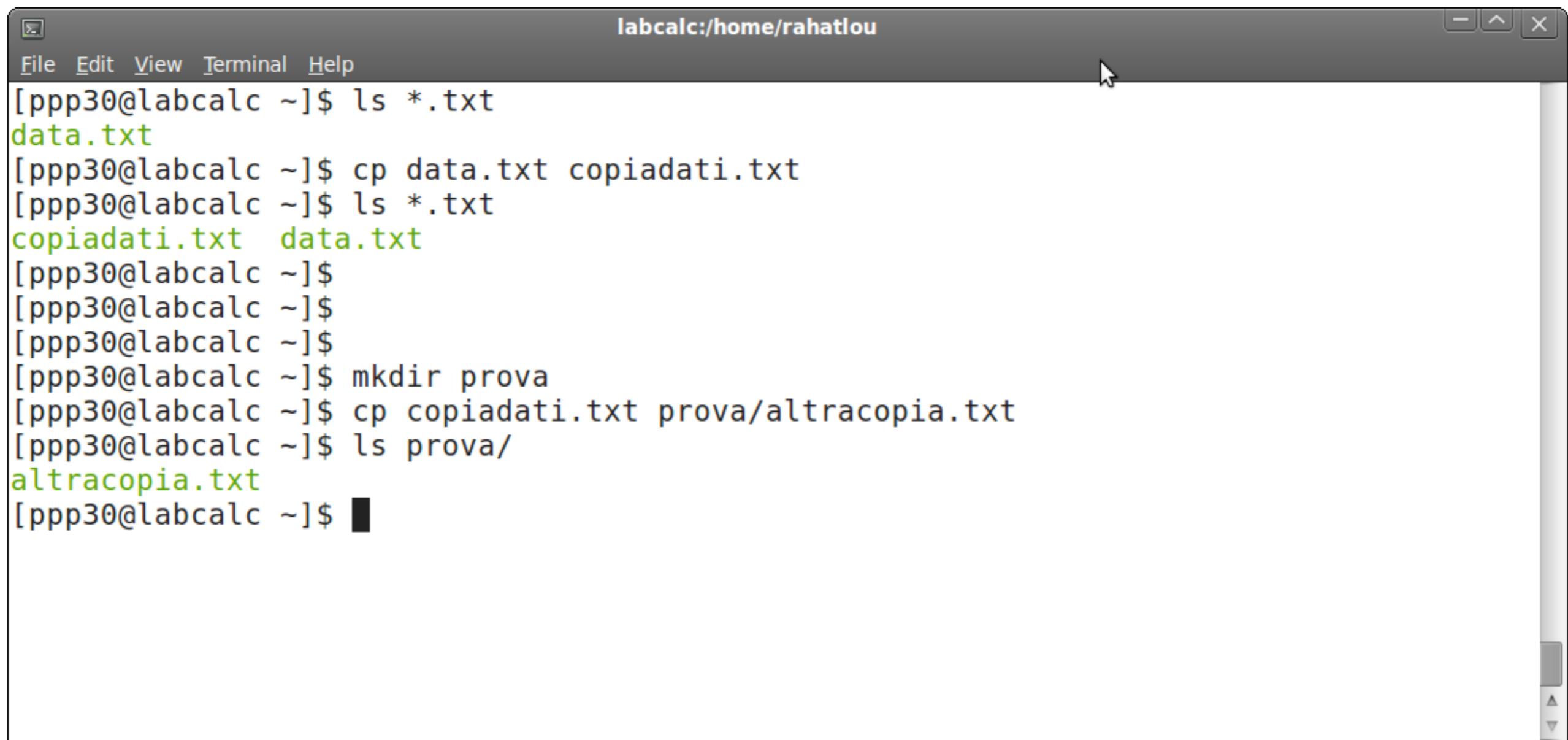
The screenshot shows a terminal window titled "labcalc:/home/rahatlou". The window contains the following command-line session:

```
[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
-rw-r--r-- 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

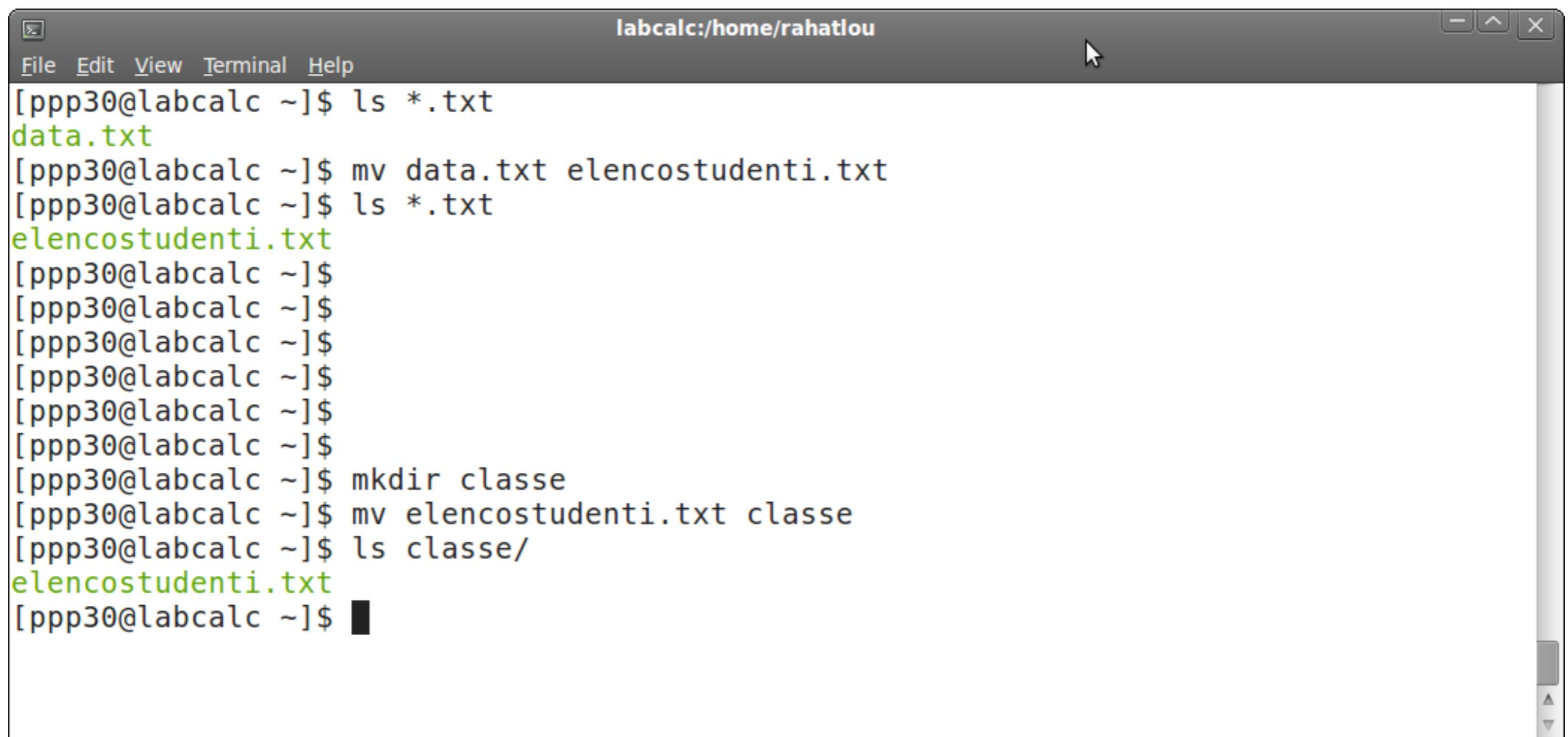


The screenshot shows a terminal window titled "labcalc:/home/rahatlou". The window contains the following command-line session:

```
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

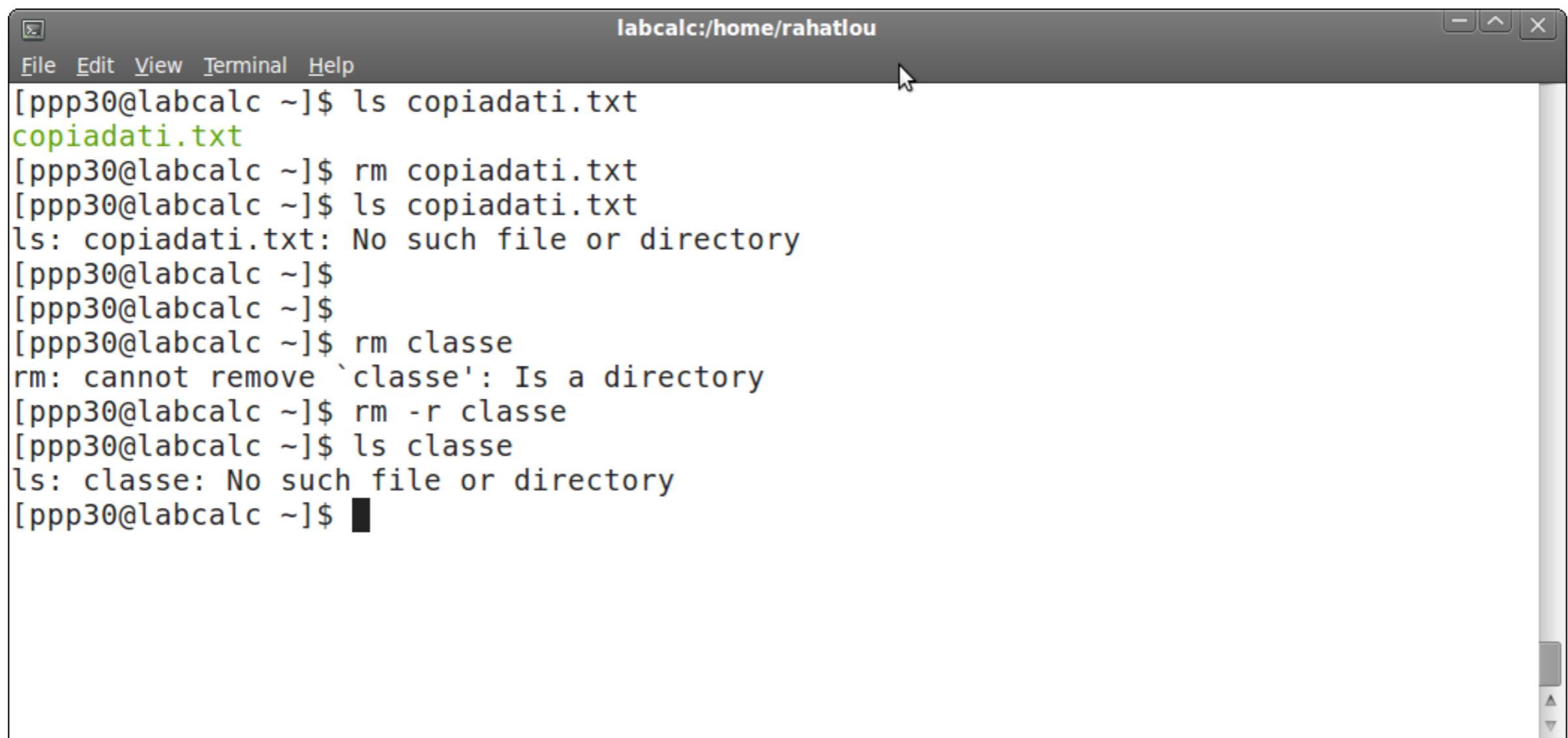


The screenshot shows a terminal window titled "labcalc:/home/rahatlou". The window contains the following session:

```
[ppp30@labcalc ~]$ ls *.txt
data.txt
[ppp30@labcalc ~]$ mv data.txt elencostudenti.txt
[ppp30@labcalc ~]$ ls *.txt
elencostudenti.txt
[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'e` modo di recuperarli
- Rimuovere una cartella rimuove anche tutti i file che contiene



The screenshot shows a terminal window titled "labcalc:/home/rahatlou". The terminal's menu bar includes "File", "Edit", "View", "Terminal", and "Help". The main area displays the following command history:

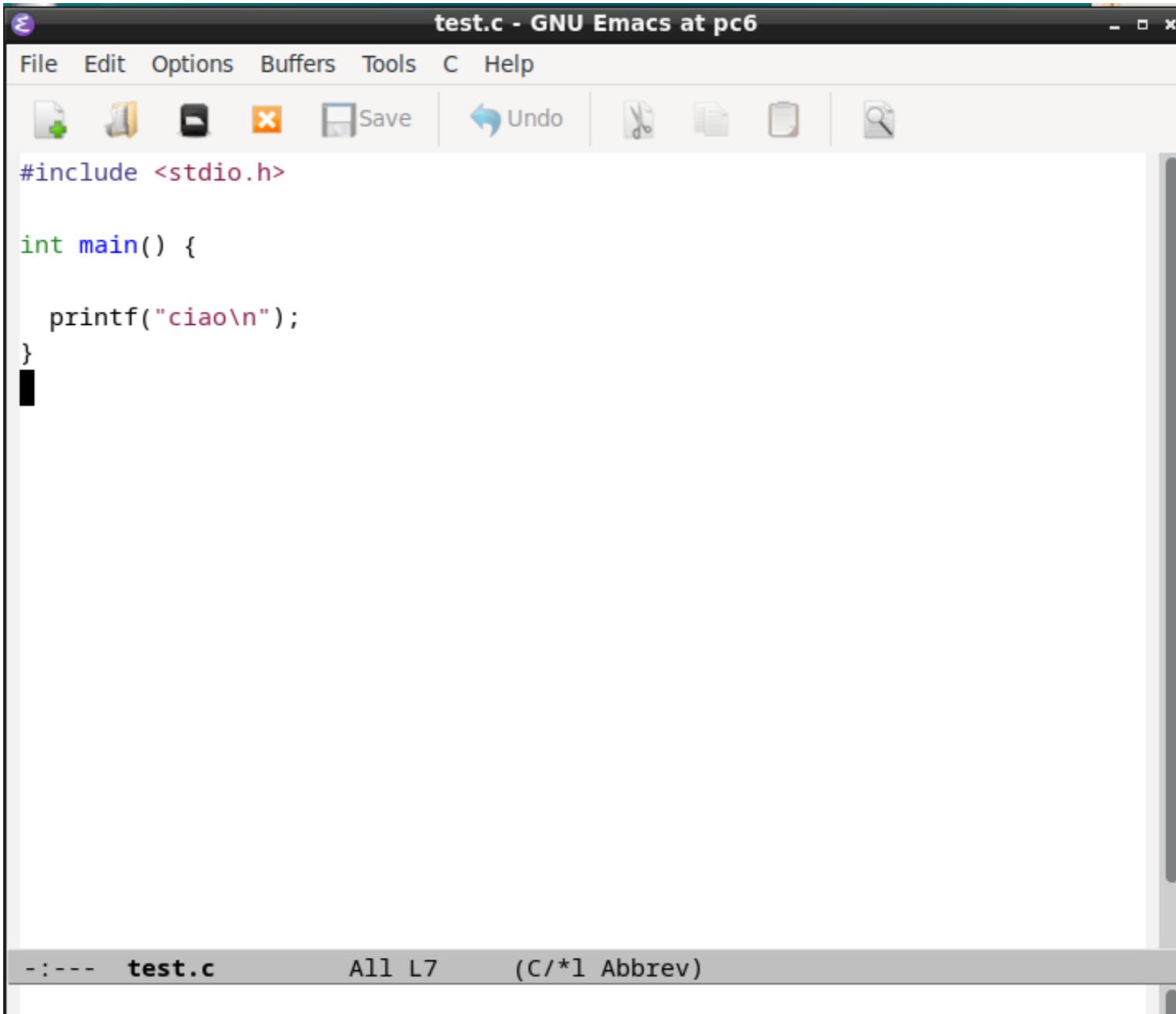
```
[ppp30@labcalc ~]$ ls copiadati.txt
copiadati.txt
[ppp30@labcalc ~]$ rm copiadati.txt
[ppp30@labcalc ~]$ ls copiadati.txt
ls: copiadati.txt: No such file or directory
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$
[ppp30@labcalc ~]$ rm classe
rm: cannot remove `classe': Is a directory
[ppp30@labcalc ~]$ rm -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

Editor di testo emacs

- Editor che useremo per questo corso



- 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 utilità visuale per l'utente e per riconoscere più facilmente errori comuni come
 - parentesi aperte ma non chiuse: viene evidenziata la parentesi aperta
 - testo commentato: di colore diverso rispetto al testo che verrà 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

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

#define NBIT 20

int main() {
    int in, cifra=0, resto;
    int binary[NBIT] = {0};

    do{
        printf("Inserisci numero positivo < %d in base 10: ", (int)pow(2,NBIT));
        scanf("%d", &in);
    } while( in <= 0 || in >= pow(2, NBIT) );

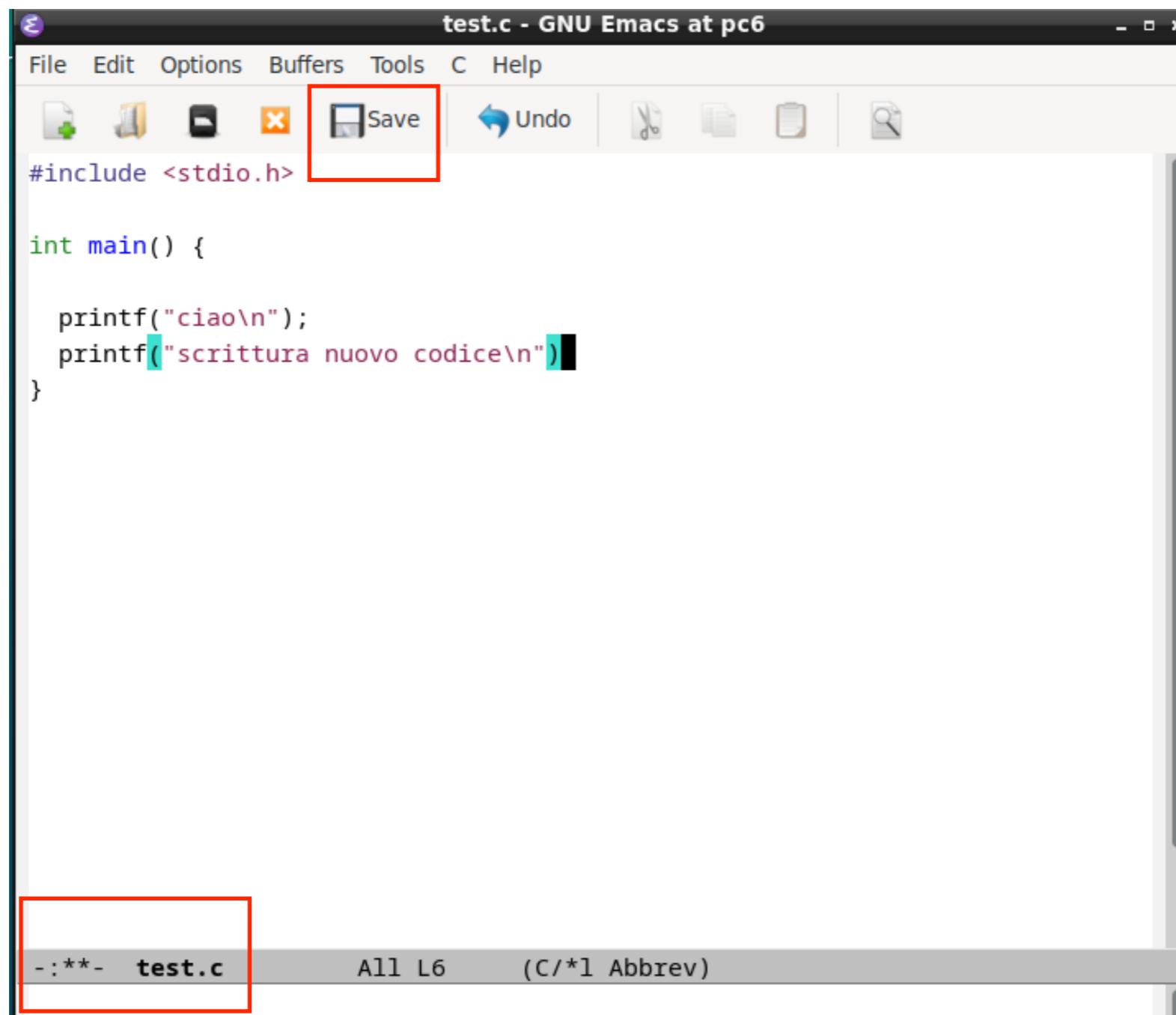
    do {
        resto = in % 2;
        printf("%3d-esima cifra: %d \t moltiplica 2 ^%3d (%8.0f)\n",cifra+1, resto, cifra,pow(2,cifra));

        binary[cifra] = resto;
        in /= 2;
        cifra++;
    } while (in != 0);

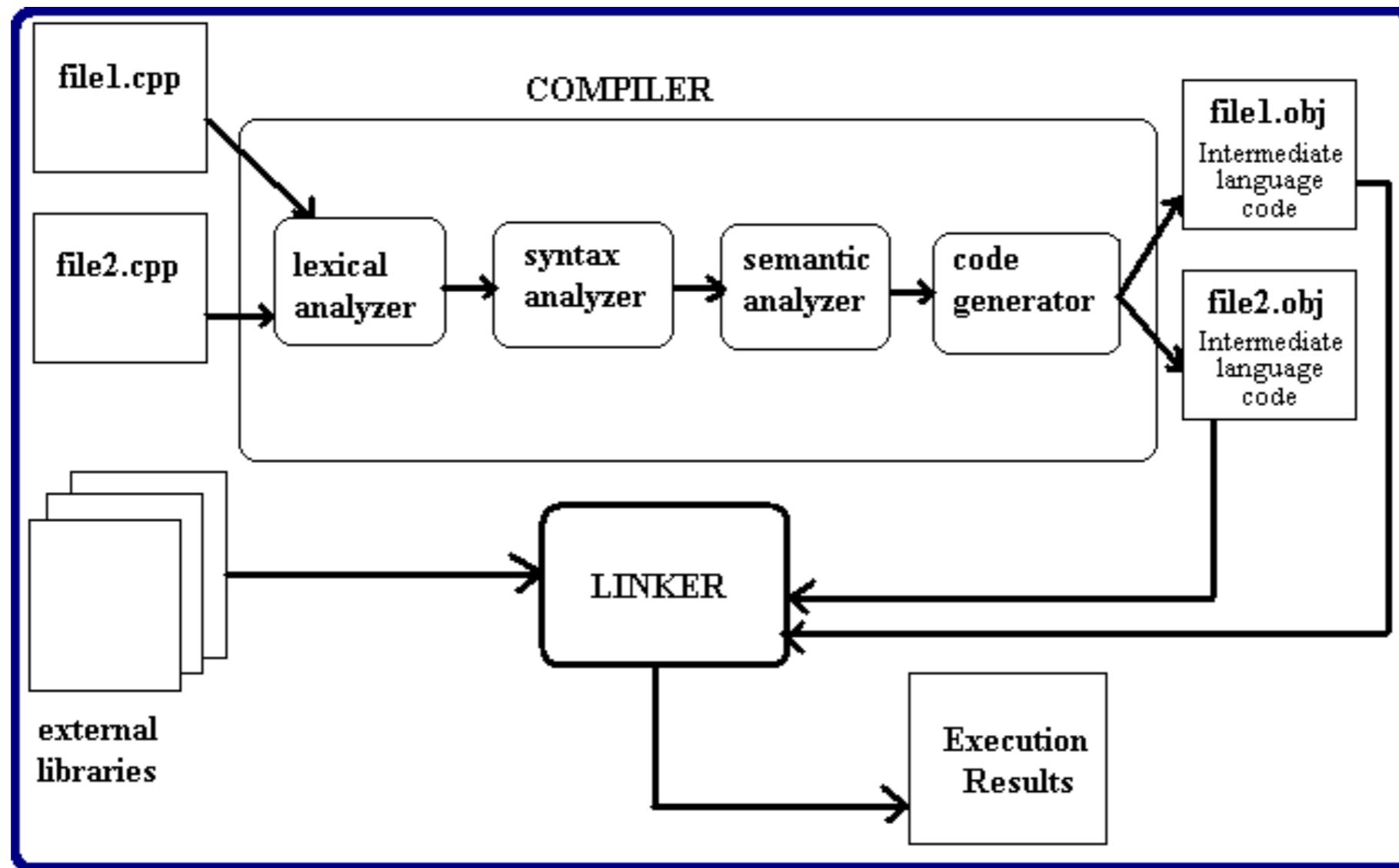
    printf("base 10: %d \t base 2: ", in);
    for(int i = NBIT-1; i>= 0; i--) {
        printf("%1d", binary[i]);
    }
    printf("\n");
}
```

Salvare un nuovo file

- Per creare un nuovo file: emacs test.c
 - 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



Compilare e 'linkare' eseguibili



- Compilare: tradurre istruzione 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

```
● ● ● ~ /Sites/Didattica/2025.labcalc/material — sh

[shamacmini:material rahatlou$ ls secondo.c
secondo.c
[shamacmini:material rahatlou$ [shamacmini:material rahatlou$ [shamacmini:material rahatlou$ gcc -c secondo.c
[shamacmini:material rahatlou$ ls secondo.*
secondo.c      secondo.o
shamacmini:material rahatlou$ ]]
```

- 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 chiamera` `a.out` oppure `a.exe`
 - Non si usa l'opzione `-c` proprio per invocare il linker dopo la compilazione
- Si puo` fare l'eseguibile solo per un file `.c` che contenga una funzione `main`

```
● ● ● ~/Sites/Didattica/2025.labcalc/material — -sh

[shamacmini:material rahatlou$ ls secondo.c
secondo.c
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$ gcc -c secondo.c
[shamacmini:material rahatlou$ ls secondo.*
secondo.c      secondo.o
[shamacmini:material rahatlou$ gcc -Wall -o app.exe secondo.c
[shamacmini:material rahatlou$ ls app.exe
app.exe
shamacmini:material rahatlou$ ]
```

Sommario uso compilatore

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

GIRARE IL PROGRAMMA ESEGUIBILE

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

● ○ ● ⚡ ~/Sites/Didattica/2025.labcalc/material — -sh

```
[shamacmini:material rahatlou$ gcc -Wall -o app.exe secondo.c -lm
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$ ls app.exe
app.exe
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$ ./app.exe
ciao a tutti3n = 3
```

Notate bene ./ prima del nome dell'eseguibile

```
3
n = 3
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$
[shamacmini:material rahatlou$ app.exe
-sh: app.exe: command not found
shamacmini:material rahatlou$
```

The screenshot shows a Mac OS X desktop environment. In the top-left corner, there is a terminal window with a dark background and white text. It displays a command-line session where the user has compiled a C program named 'secondo.c' into an executable file 'app.exe'. The user then attempts to run the executable directly from the terminal, which results in an error message indicating that 'app.exe' is not a recognized command. In the top-right corner, there is a code editor window titled 'secondo.c'. The code editor has a light gray background and displays the source code for the program. The code includes standard library includes, a main function that prints 'ciao a tutti' and the value of 'n', and a brace at the end of the function. The code editor interface includes various icons for file operations like New, Open, Save, Undo, Redo, Cut, Copy, Paste, and Search.

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

int main() {
    int n;
    n = 3;

    printf("ciao a tutti");

    printf("%d", n);

    printf("n = %d", n);

    printf("\n\n");

    printf("%d\n", n);
    printf("n = %d\n", n);

}
```

-:--- secondo.c All (1,0) Git-2025 (C/I Abbrev)