Introduzione al sistema Linux

- Stazioni di lavoro: PC con sistema operativo Linux

- Connessione al sistema

  Username : cognome1

  Password : FirstLogin

  (cambiate la password con yppasswd appena possibile)

- Linux: multitask e multiuser

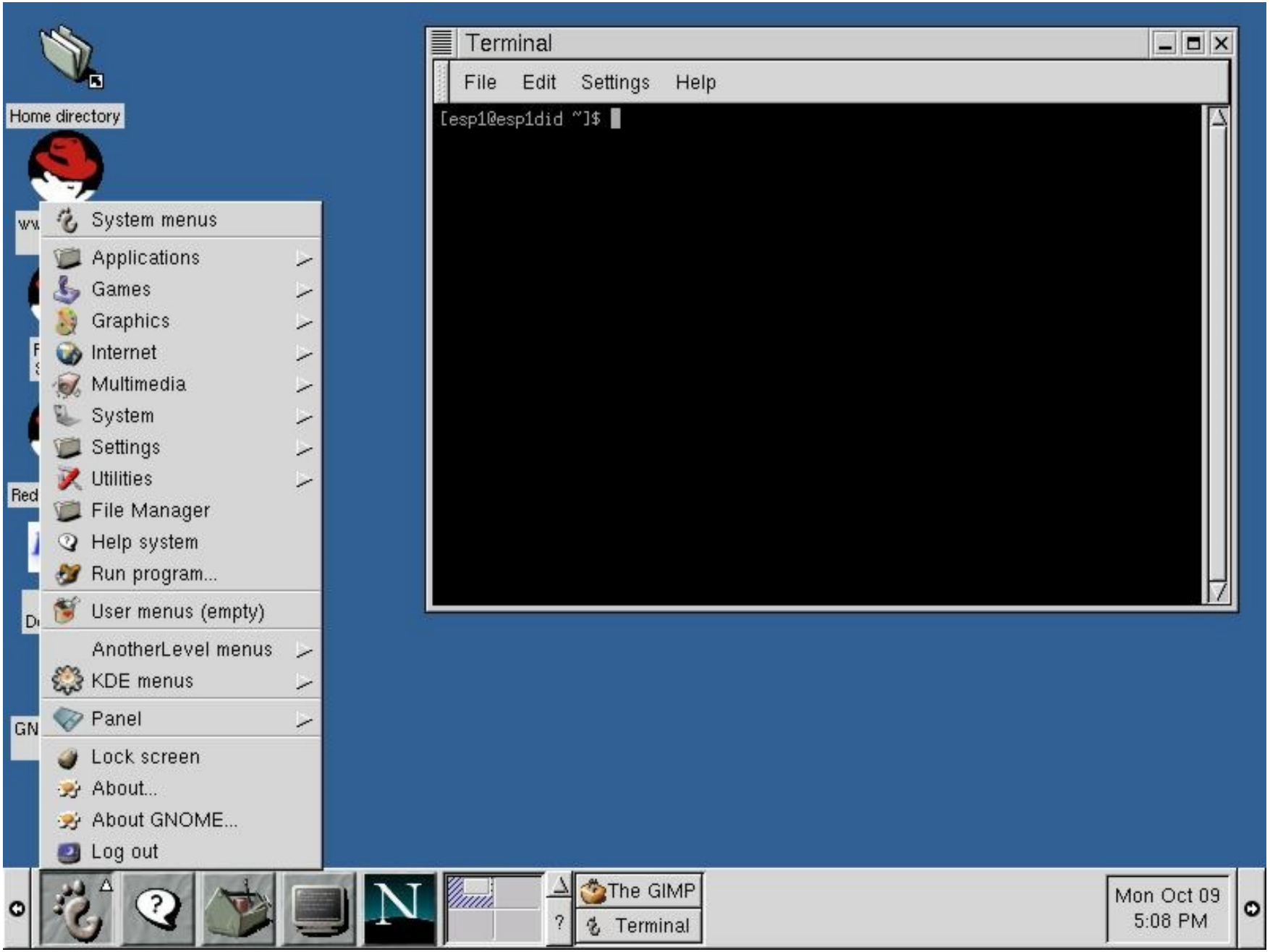
  Kernel – cuore del sistema

  Shell - interfaccia con l' utente

  (Shell = conchiglia: assicura che ogni terminale comunichi indipendentemente con il sistema)

- Comandi:
  - terminati dal tasto Enter
  - "case" sensitive

    (attenzione a maiuscole/minuscole !!)


Home directory
Terminal
File Edit Settings Help
[esp1@esp1did ~]$
System menus
Applications
Games
Graphics
Internet
Multimedia
System
Settings
Utilities
File Manager
Help system
Run program...
User menus (empty)
AnotherLevel menus
KDE menus
Panel
Lock screen
About...
About GNOME...
Log out
The GIMP
Terminal
Mon Oct 09
5:08 PM

Controllo dei processi

- Ogni SHELL puo' eseguire contemporaneamente diversi processi
  ps -> process status
  kill PID -> ferma il processo con identificatore PID
- Tutti i comandi sono associati a programmi: alcuni ripassano immediatamente il controllo alla shell (ls=lista file), altri aprono finestre grafiche e bloccano la shell fino a che non viene chiusa la finestra
- Esecuzione di programmi
  nome -> blocca il terminale (foreground)
  nome & -> lascia il terminale libero (background)

Programma in foreground

Ctrl-C -> fermato bruscamente
Ctrl-Z bg -> sospeso e mandato in background

Programmi in background

jobs -> lista
fg n -> manda in foreground il numero di job
bg n -> manda in background il numero di job

Esercizi:

Provate a gestire xcalc (calcolatrice) e xclock (orologio) in background/foreground

Files e directories

- File: contenitore di informazioni memorizzate in modo permanente (porzione di disco magnetico)
- Directory: contenitore di files
- Intero sistema: "filesystem"
- Dove siamo ?

  pwd -> Working Directory (WD) = .

  Home Directory (HD) - Directory di login = ~
- Per muoversi sull' albero: cd

  cd .. -> risale di un passo nella catena

  cd nome -> scende nella dir nome
- Per creare una nuova dir: mkdir *nome*
- Per cancellare una dir: rmdir *nome*
- Esercizi:

  - Provate a muovervi nel vostro albero creando sottodirectories e spostandosi con .., ~ , o con cd nome completo. Verificate la vostra posizione con pwd
- Directory /tmp disponibile (solo files temporanei)

Editore di testo

- Un editore di testo serve a creare un file *leggibile*
- L'editore che useremo si chiama **emacs**: per attivarlo basta dare il comando:

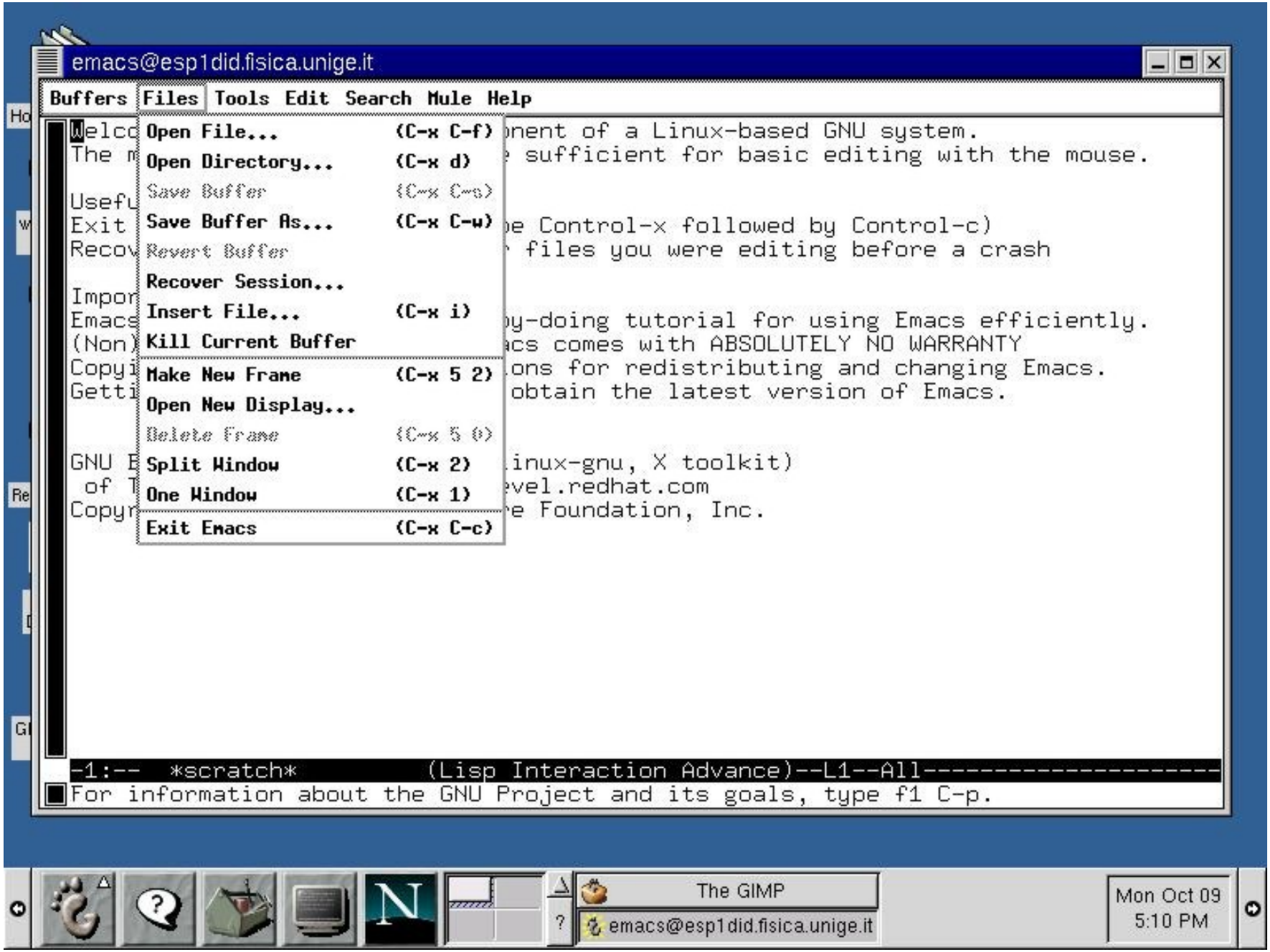
  emacs *nomefile* **&** oppure

  emacs &

(& per lavorare in background)

- Viene creata una nuova finestra per l'editore di testo
- Copiate nella vostra dir principale (~) il file test.txt con il comando:

  cp /usr/users/labc/test.txt ~/test.txt
- Con il menu Files (Open file) aprite il file
- Ora provate a modificarlo usando i comandi del menu Edit (tagliare, copiare e incollare) o semplicemente da tastiera
- Provate a usare il comando search per cercare delle stringhe (delle parole)
- Il menu Files vi permette di salvarlo (Save Buffer as)


emacs@esp1did.fisica.unige.it
Buffers Files Tools Edit Search Mule Help
Open File... (C-x C-f)
Open Directory... (C-x d)
Save Buffer (C-x C-s)
Save Buffer As... (C-x C-w)
Revert Buffer
Recover Session...
Insert File... (C-x i)
Kill Current Buffer
Make New Frame (C-x 5 2)
Open New Display...
Delete Frame (C-x 5 0)
Split Window (C-x 2)
One Window (C-x 1)
Exit Emacs (C-x C-c)
-1:-- *scratch* (Lisp Interaction Advance)--L1--All
For information about the GNU Project and its goals, type f1 C-p.
The GIMP
emacs@esp1did.fisica.unige.it
Mon Oct 09
5:10 PM

Comandi per la manipolazione di file e directories

echo *linea* stampa la linea di testo

ls *dir* lista di file nella directory specificata

-l stampa info aggiuntive sui file (priorita',data...)

-a stampa tutti i file (compresi i .nome)

-F stampa tipo di file (dir, eseguibile...)

mv *nome1 nome2* ridenomina nome1 in nome2

cp *nome1 nome2* copia nome1 in nome2

-r copia il contenuto della dir nome1 in nome2

rm *file* rimuove file

-i chiede conferma prima di cancellare

cat *file* stampa su schermo il contenuto di file

less *file* impagina in schermate il contenuto di file

tail –nN *file* stampa le ultime N linee di file

head –nN *file* stampa le prime N linee di file

grep *string file* cerca la stringa in file

-i ignora maiuscole e minuscole

find –name *file* cerca file nella wd e sottodir

diff *file1 file2* stampa linne diverse tra file1 e file2

ln –s *file link-name* crea equivalenza simbolica tra link-name e file

Nel nome del file ci possono essere wildcard:

* Sostituisce una qualsiasi stringa di caratteri (ls *.txt)

? Sostituisce un singolo carattere

Redirezione e Pipe

- Standard input(tastiera)/output(terminale)
- Ridirezione dell'output:
  - **command > filename**: l'output del comando, invece di essere visualizzato sullo schermo, viene salvato nel file filename

  (ls > lista e poi cat lista)
  - **command >> filename**: l'output del comando è appeso a filename (che se non esiste viene creato)

  (ls >> lista e poi cat lista)

- Pipe: connette l'output di un comando all'input del successivo (è equivalente a mettere lo standard output in un file e poi usare quel file come input per il command successivo)
  - ls –lt | head -n2

Protezioni

- Eseguite il comando ls –l

  Cosa sono i primi caratteri sulla sinistra di fianco ad ogni files ?

- Indicano chi e come puo' accedere ai files

Type	User	Group	Other
*	***	***	***
d	rwx	rwx	rwx
l	---	---	---
-			

  Type: d=dir, l=link, - =file

  r=read, w=write, x=execute

- Per cambiare le protezioni: chmod

  chmod u+x file: lo user puo' eseguire

  chmod o-r file: gli "altri" non possono piu' leggere

  chmod a+r file: tutti possono leggere il file

Manipolazione file

Es.1: Con emacs cambiate un file e poi verificate le differenze

Es.2: Stampate una stringa, provate poi a riindirizzarla in un file

Es.3: Cercate in tutti i vostri files una particolare parola; salvate l'elenco dei file che la contengono nel file *lista*

Es.4: Cercate nel vostro filesystem il file *test.txt*

Es.5: Dati due file, create uno nuovo che sia il concatenamento dei due. Provate ad usare sia > che >> e guardate le differenze.

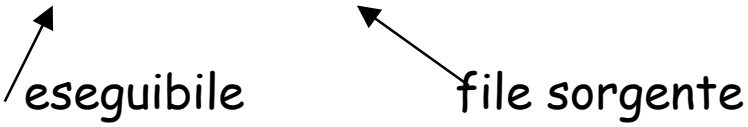
Es.6: Create un link simbolico nella vostra pwd e verificate che se agite con dei comandi sul link, il file a cui punta viene modificato di conseguenza (ad es. cat, appendere un file…)

Es.7: cambiate le protezioni di un file (per all,other,user,group)

Primo programma in C

- Si usa un editore di testo per scrivere il file sorgente (hello.c)
- Compilatore per creare l'eseguibile:

cc –o *hello hello.c*



- Es:

```
main(){
   printf("Hello World");
   printf("This is just a test");
}
```

- Esercizio 1:
  - **Compilare il programma e verificare che si è creato un file eseguibile nella wd**
  - **Eseguire il programma**
- Esercizio 2:
  - **Migliorare l' output con \n**
  - **Provate a ridirezionare l'output in un file**