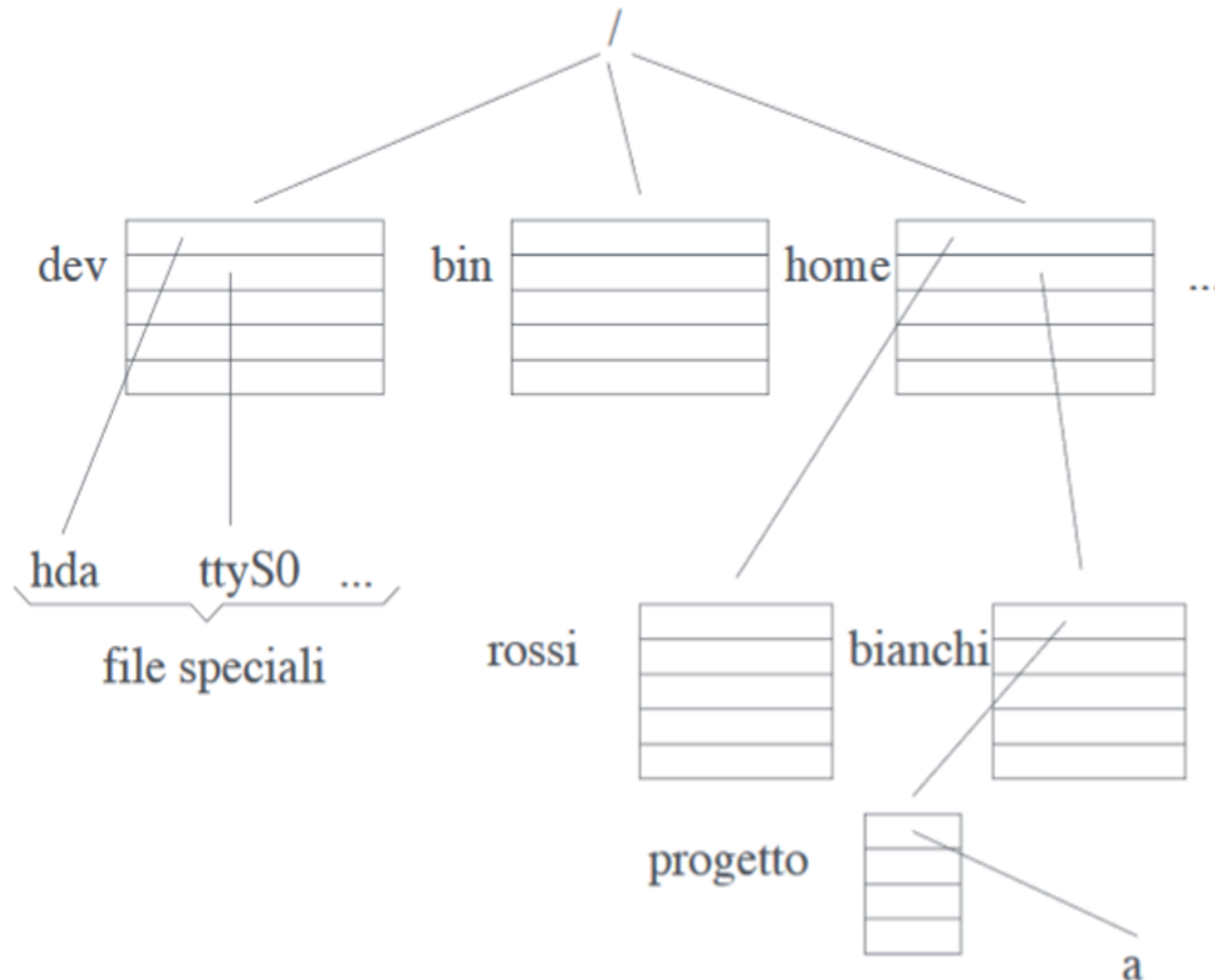


Laboratorio 1

Sistema Operativo Linux, Struttura del file system



- / è la **radice** (o root) del file-system
- i nodi interni sono le **directory**
- le foglie sono i **file**

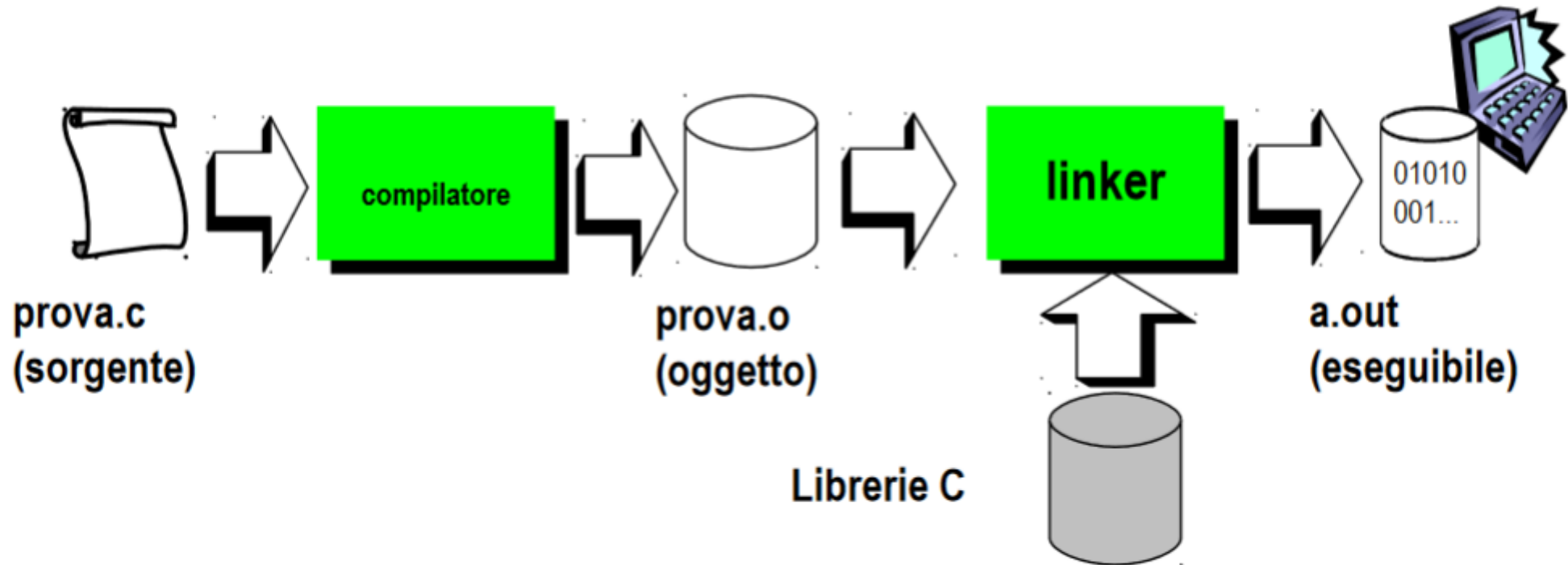
per prima cosa apriamo un terminale o shell: saremo in un nodo dell'albero del file system e possiamo spostarci nell'albero con appositi comandi

- pwd (print working directory) per sapere dove siamo
- cd (change directory) per spostarsi nell'albero del file system
- ls (list) sapere il contenuto della directory corrente (opzioni, -l....)
- protezione

- mkdir nome, per costruire la directory nome
- rmdir nome, per eliminare la directory nome
- man comando per avere la descrizione del comando
- per scrivere un programma si deve chiamare un editore di testo: gedit o emacs o quello che piace a voi

chiamiamo **gedit esempio.cpp** &
scriviamo il programma di ieri
lo salviamo con estensione .cpp: **esempio.cpp**
lo compiliamo con **g++ esempio.cpp -o esempio**
lo eseguiamo con **./esempio**

g++ invoca il compilatore C++ GCC della GNU



possiamo compilare senza linkare con

```
g++ -c esempio.cpp
```

produce esempio.o, un ulteriore comando g++

```
g++ -o esempio esempio.o
```

crea l'eseguibile esempio

debugging con GDB

`g++ -g esempio.cpp -o esempio`

crea un eseguibile con informazioni aggiuntive per fare successivamente il debugging

`gdb ./esempio` esegue il programma in modalità debugging, trovate un semplice tutorial qui:

<https://u.osu.edu/cstutorials/2018/09/28/how-to-debug-c-program-using-gdb-in-6-simple-steps/>

nella shell possiamo fare molte cose con i file

-- rm per eliminarli

--cp per copiarli in un'altra directory

--mv per muoverli o rinominarli

--cat file1 file2.... per vedere il contenuto

--head -n N file1 file2 ...

--tail -n N file1 file2

--history per vedere i comandi digitati in precedenza

-- le frecce verso alto/basso li scorrono per ripeterli

--con tab c'è il completamento automatico di comandi e nomi