Unix & Linux

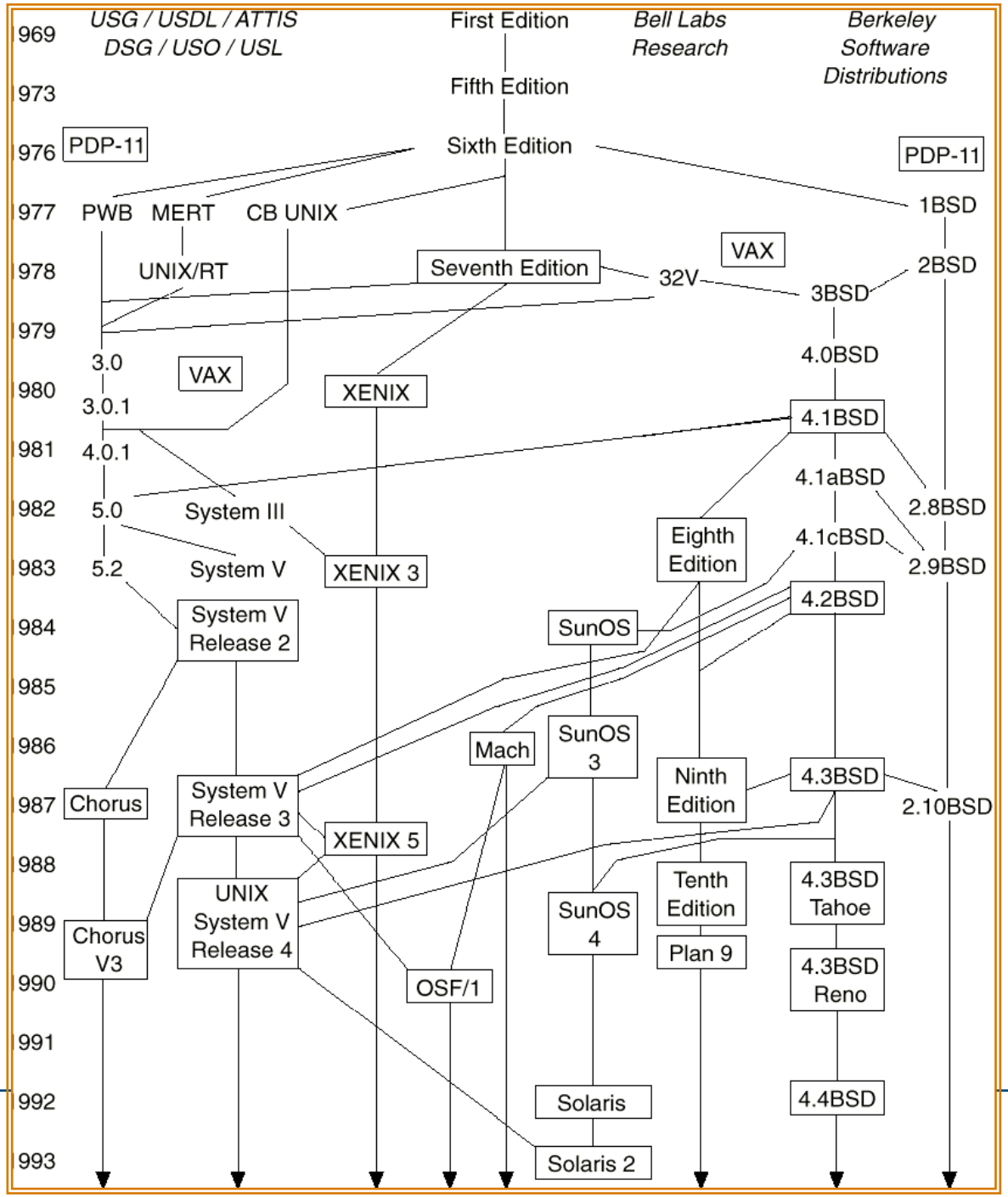
Storia di Unix

- **1969**: AT&T, sviluppo di un ambiente di calcolo multiprogrammato e portabile per macchine di medie dimensioni.
- **1970**: prima versione di UNIX (multiprogrammata e monoutente) interamente sviluppata nel linguaggio assembler del calcolatore PDP-7.
- **Anni 1970**: nuove versioni, arricchite con altre caratteristiche e funzionalità. Introduzione del supporto alla multiutenza.

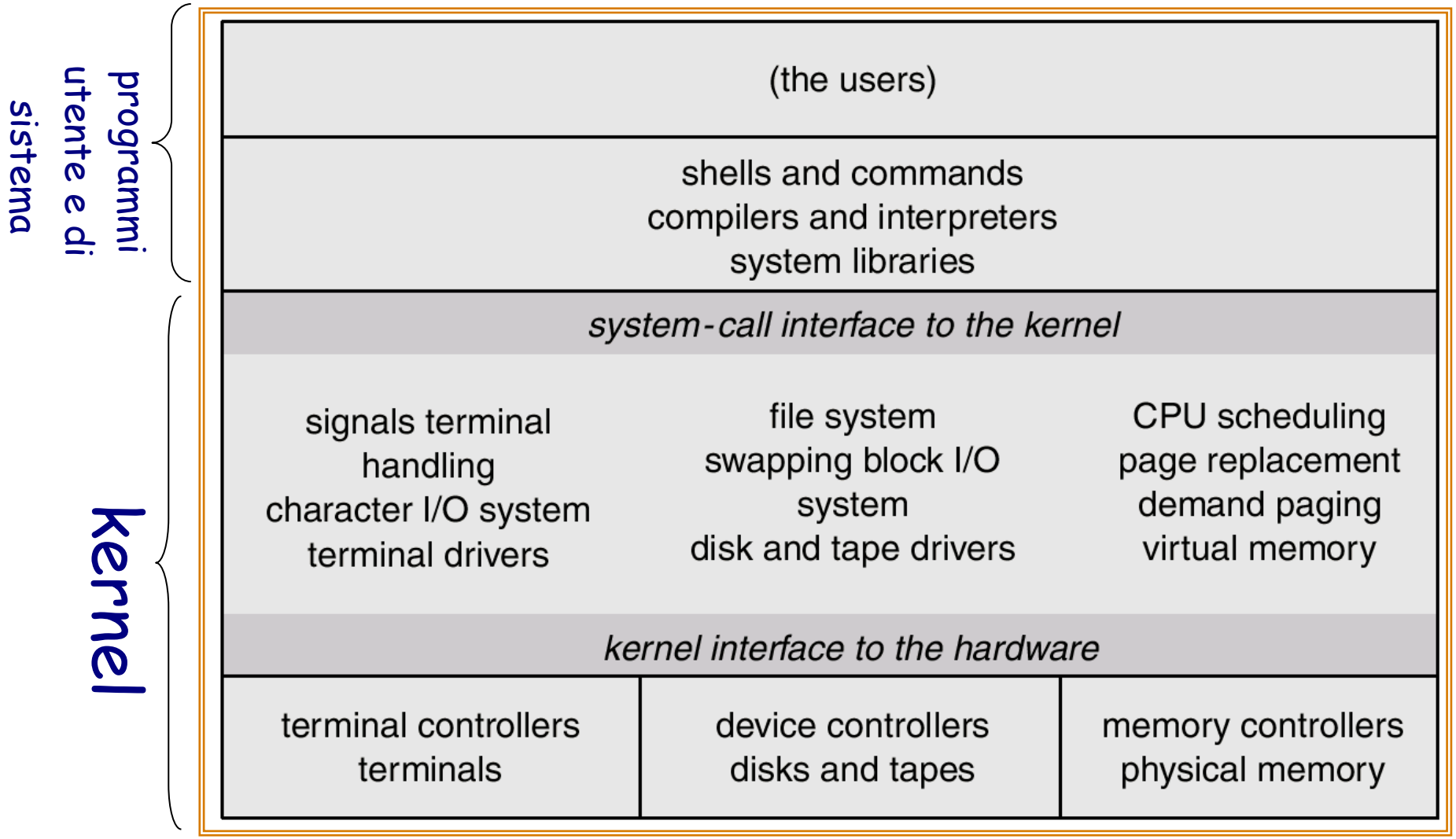
Unix e il linguaggio C

- **1973**: Unix viene realizzato nel linguaggio di programmazione C:
  - Elevata portabilità
  - Leggibilità
  - Diffusione presso la comunità scientifica e accademica.

- **Anni 80**: la grande popolarità di Unix ha determinato il proliferare di versioni diverse. Due famiglie:
  - Unix System V (AT&T Laboratories)
  - Unix Berkeley Software Distributions, o BSD (University of California at Berkeley)


969
973
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
USG / USDL / ATTIS
DSG / USO / USL
Bell Labs
Research
Berkeley
Software
Distributions
First Edition
Fifth Edition
Sixth Edition
PDP-11
PDP-11
PWB
MERT
CB UNIX
1BSD
UNIX/RT
Seventh Edition
32V
VAX
2BSD
3BSD
3.0
VAX
4.0BSD
XENIX
3.0.1
4.1BSD
4.0.1
4.1aBSD
2.8BSD
5.0
System III
Eighth
Edition
4.1cBSD
2.9BSD
5.2
System V
XENIX 3
4.2BSD
System V
Release 2
SunOS
Mach
SunOS
3
Chorus
System V
Release 3
Ninth
Edition
4.3BSD
2.10BSD
XENIX 5
UNIX
System V
Release 4
Tenth
Edition
4.3BSD
Tahoe
Chorus
V3
SunOS
4
Plan 9
4.3BSD
Reno
OSF/1
Solaris
4.4BSD
Solaris 2

Organizzazione di UNIX



Caratteristiche di Unix

- multi-utente
- time sharing
- kernel monolitico
- Ambiente di sviluppo per programmi in linguaggio C
- Programmazione mediante linguaggi comandi
- portabilità

POSIX

- **1988**: POSIX (*Portable Operating Systems Interface*) è lo standard definito dall'IEEE. Definisce le caratteristiche relative alle modalità di utilizzo del sistema operativo.
- **1990**: POSIX viene anche riconosciuto dall' International Standards Organization (ISO).
- **Anni 90**: Negli anni seguenti, le versioni successive di Unix SystemV e BSD (versione 4.3), si uniformano a POSIX.

Introduzione a GNU/Linux

- GNU project:
  - **1984**: Richard Stallman avvia un progetto di sviluppo di un sistema operativo ***libero* compatibile con Unix:**

    "GNU is Not Unix"
  - Furono sviluppate velocemente molte utilita` di sistema:
    - editor Emacs,
    - Compilatori: gcc,
    - shell: bash,
    - ...
  - lo sviluppo del kernel (Hurd), invece, subi` molte vicissitudini e vide la luce molto piu` tardi (1996)

GNU/Linux

- **1991**: Linus Tornvalds realizza un kernel Unix-compatibile (Minix) per l'architettura intel x86 e pubblica su web i sorgenti
- In breve tempo, grazie a una comunita` di *hacker* in rapidissima espansione, Linux acquista le caratteristiche di un prodotto affidabile e in continuo miglioramento.
- **1994**: Linux viene integrato nel progetto GNU come kernel del sistema operativo: nasce il sistema operativo GNU/Linux

GNU/Linux

Caratteristiche:

- Open Source / Free software
- multi-utente, multiprogrammato e multithreaded
- Kernel monolitico con possibilita` di caricamento dinamico di moduli
- estendibilita`
- affidabilita`: testing in tempi brevissimi da parte di migliaia di utenti/sviluppatori
- portabilità

Distribuzioni GNU/Linux

Attualmente varie **distribuzioni GNU/Linux** (comunemente *distro*):

- Interfacce grafiche diverse (Gnome, KDE, Xfce, ecc.)
- collezione di **pacchetti** (applicativi) **software**: archivi compressi usati per **automatizzare** e **semplificare** l'installazione di applicazioni (compilazione dei sorgenti, impostazione delle variabili di ambiente, configurazione di permessi, ecc...)
- alcuni esempi: Redhat/Fedora, Slackware, Debian, Gentoo, Ubuntu, SUSE, ecc...

Gestori di pacchetti: sono pacchetti a loro volta

- differenti per *famiglie* di distribuzioni (RPM, APT, Portage,...)
- operazioni di installazione, rimozione, aggiornamento di pacchetti software

Ai fini del corso...

Necessità di utilizzare un sistema operativo **Unix-like**; varie possibilità:

- **Installazione** di una distribuzione Linux su una macchina fisica:
  - maggiore apprendimento ma complessità e problematiche maggiori (partizionamento del disco fisso, dual booting, ecc...)
- Uso distribuzioni **Linux Live CD**
  - nessuna installazione, ambiente di lavoro "stateless" e ripetibile caricato in RAM, elevati requisiti hardware, prestazioni penalizzate
- **Virtualizzazione**
  - installazione necessaria ma priva di implicazioni per la macchina fisica (configurazione dual boot, partizionamento del disco), prestazioni ragionevoli

Linux / Unix: la shell

utenti e gruppi, shell, comandi

Utenti e gruppi

- Sistema multiutente ⇒ problemi di protezione (possibili interferenze): necessità di proteggere / nascondere informazione
- Concetto di gruppo (es. staff, users, students, ...): possibilità di lavorare sugli stessi documenti
- Ogni utente appartiene a un gruppo ma può far parte anche di altri a seconda delle esigenze e configurazioni

Accesso a Linux: *login*

- Per iniziare una sessione bisogna essere in possesso di una *combinazione*:
  - username (es. **x135462, dll28493, ...**)
  - password (es. **dfh@2#q, **a890, aPP&x., ...**)
- nota: maiuscole / minuscole sono caratteri diversi!! (la password **a890 è diversa da **A890)
- Accesso al sistema: `login: x135462`
  `Password: ******`

NB per ottenere le credenziali per accedere alle macchine dei laboratori:

https://infoy.ing.unibo.it/

shell...

- Una volta superata la fase di login, l'utente è collegato al sistema Linux. Di norma è presente una finestra di *shell*
- La shell è l'interprete del linguaggio comandi; e` un programma che consente di far **interagire** l'utente col sistema.
- Comportamento shell:
  - attesa di comandi (immessi dall'utente con la tastiera),
  - Ogni comando ricevuto, viene mandato in esecuzione una volta ricevuto l'<ENTER>

...shell

- interfaccia di alto livello tra utente e S.O.
- processore comandi evoluto: interpreta e mette in esecuzione comandi da:
  - standard input (tastiera)
  - file comandi
- linguaggio comandi con **elevato potere espressivo**

Varie shell di Unix

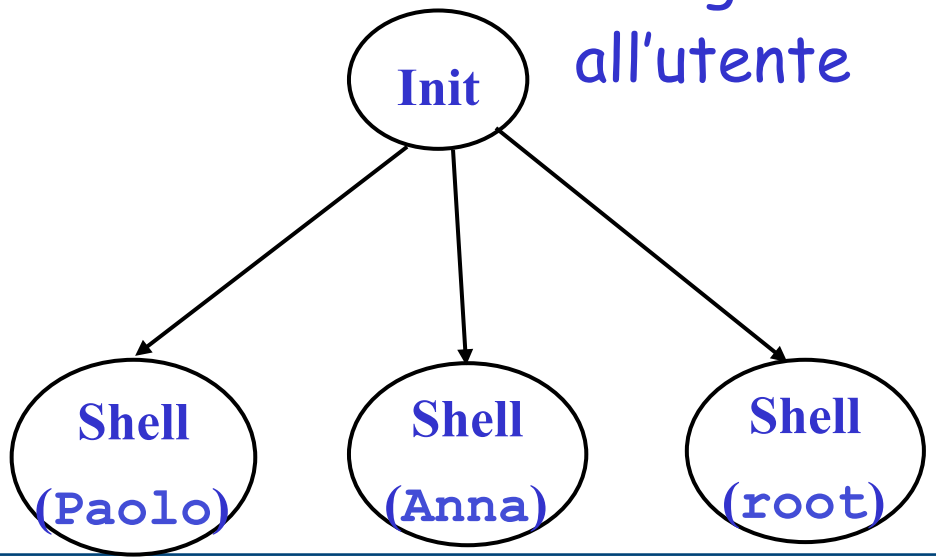
- esistono diverse Shell in Unix:
  - Bourne Shell (standard)
  - C Shell
  - Korn Shell
  - Tc Shell
  - etc

- L'implementazione della Bourne shell sotto Linux si chiama ***bash*** (Bourne-Again shell).

Shell: Accesso al Sistema

- un utente può attivare più shell, anche diverse: tcsh, csh, bash, ...
- una shell in particolare è chiamata shell di login (quella per cui viene chiesta inizialmente la password)
- la shell di login fornisce un accesso al sistema a ciascun utente:

la shell è rappresentata da un processo assegnato all'utente



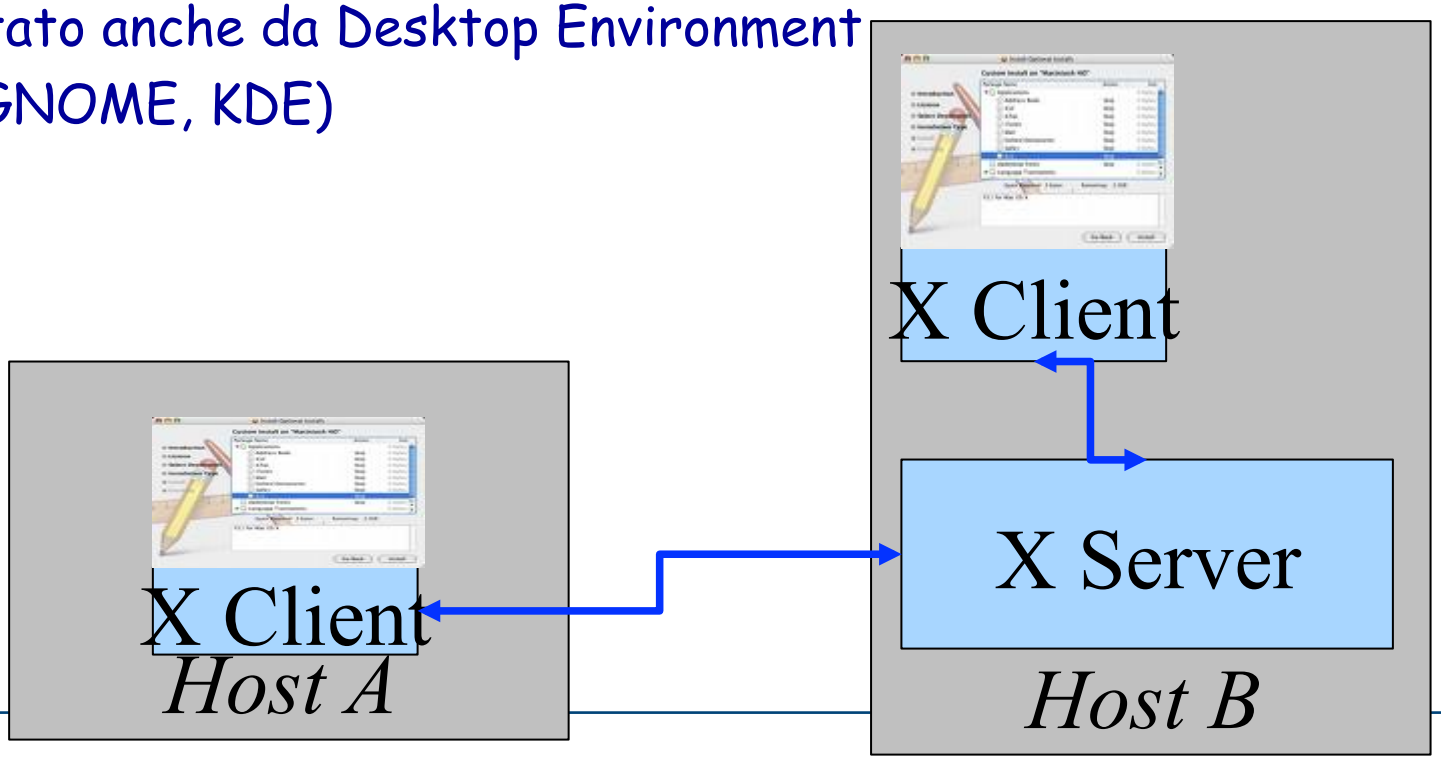
uscita da una shell

- per uscire dal ciclo di una shell di login si può:
  - usare il comando `logout`, oppure
  - digitare CTRL+D (carattere di end-of-file)
- una volta effettuato il logout, per riprendere a usare linux bisogna inserire nuovamente username e password (login)
- per uscire da una shell anche non di login esiste il comando `exit`.

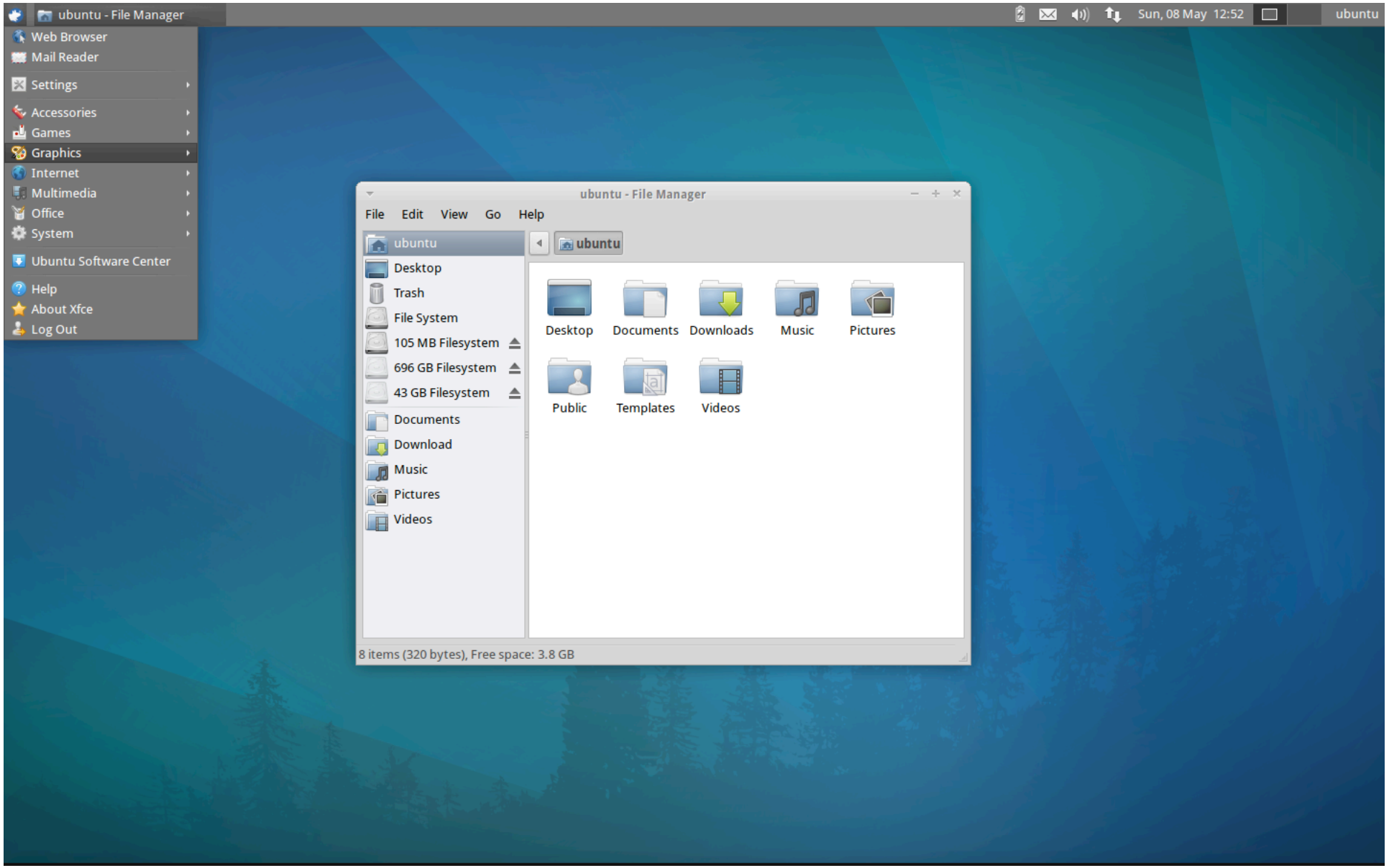
X Window System

Strumento per la gestione di GUI (Graphical User Interface)

- approccio network-based client/server
- cross-platform: non legato ad un particolare S.O.
- sfruttato anche da Desktop Environment (es. GNOME, KDE)



Interfaccia Grafica: es. XFCE



Comandi della shell di Unix

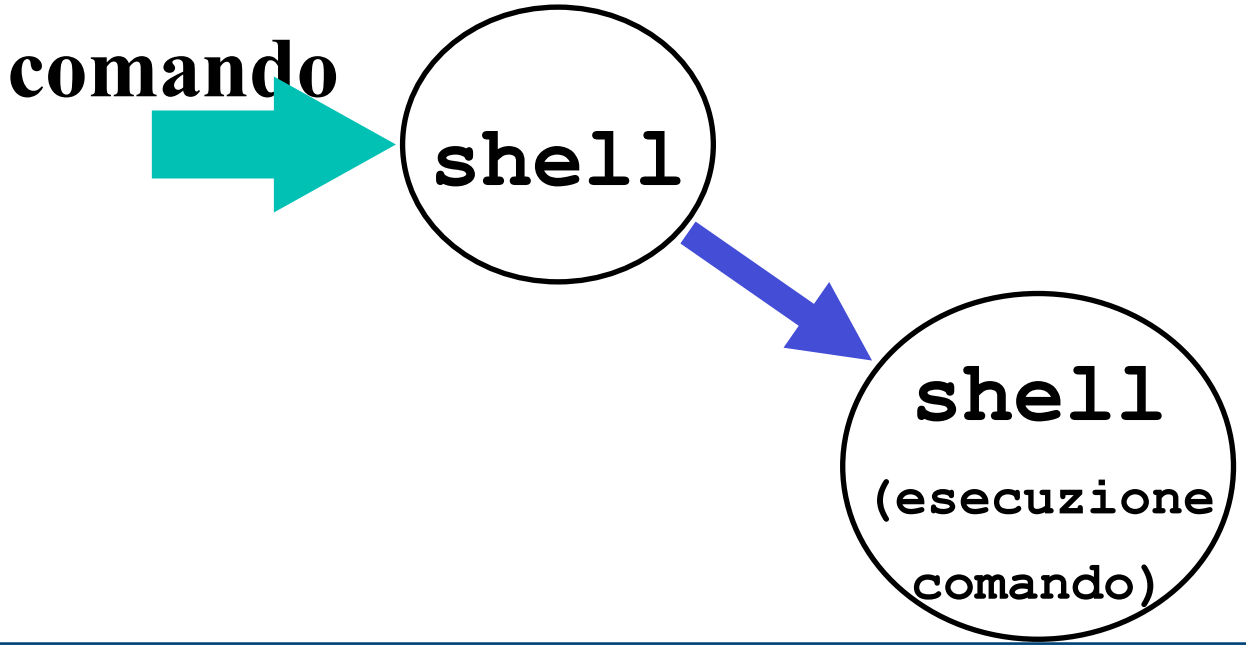
standard input, output, error;
tipi di comandi

Lo Shell di Unix:
Comandi

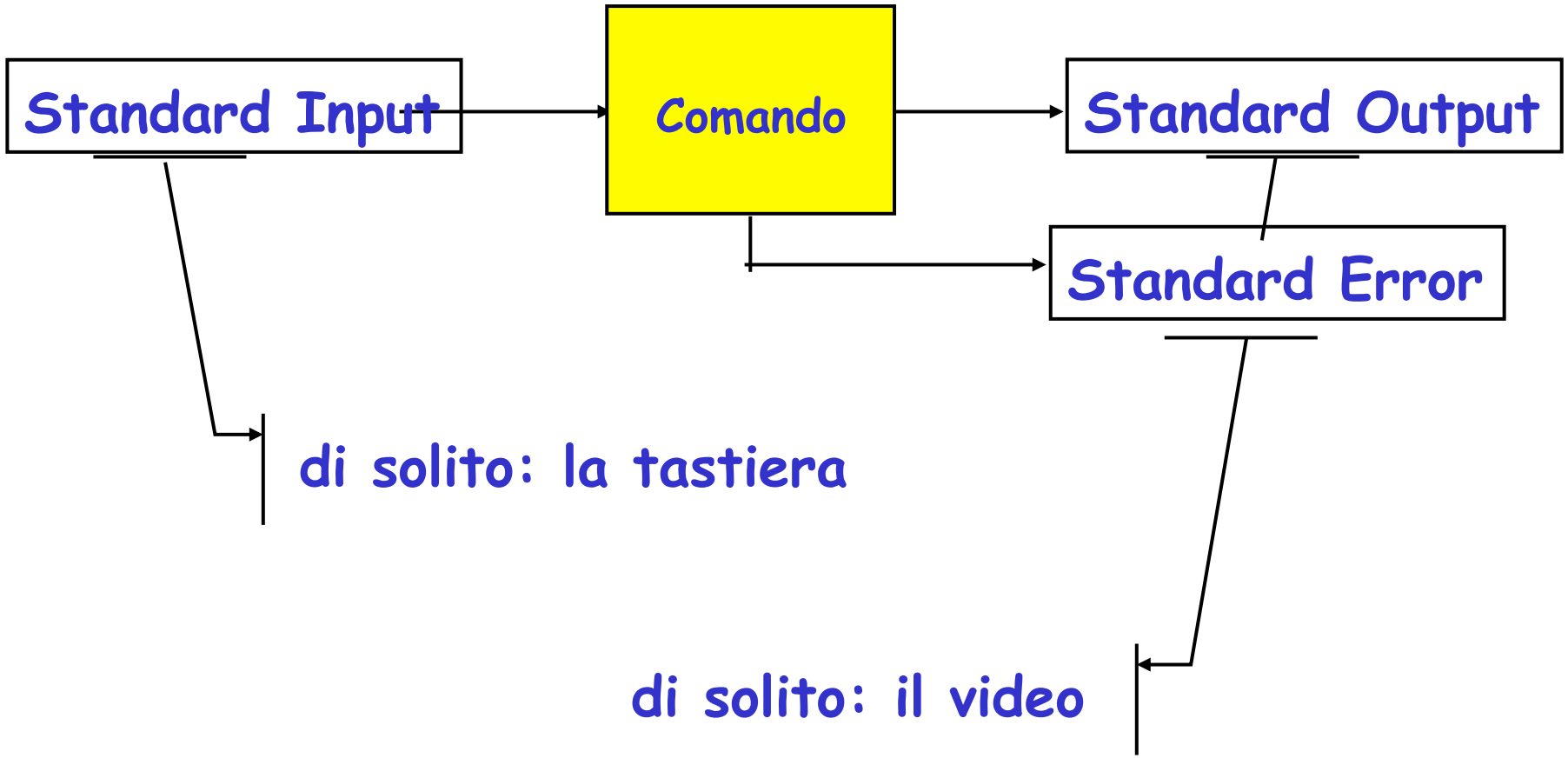
- ogni comando richiede al nucleo l'esecuzione di una particolare azione
- i comandi esistono nel file system come file binari, generalmente eseguibili da tutti gli utenti (direttorio **`/bin`**)
- possibilità di realizzare nuovi comandi: programmazione in shell

esecuzione di comandi

- per ogni comando da eseguire lo shell crea uno shell figlio, dedicato all'esecuzione del comando:



input / output di un comando



alcuni tipi di comandi

- interazione con il file system:
  - gestione di file e direttori
- gestione del sistema:
  - informazioni sulle risorse
  - modifica di dati di sistema

esempi di comandi

```
pippo@lab3-linux:~$ date
Wed Apr 27 21:48:24 CEST 2005

pippo@lab3-linux:~$ who (connected users info)
root      pst/3     Apr 9 14:02
root      pst/4     Apr 22 17:11 (:0.0)
Paolo     pst/12    Apr 27 12:21 (deis32...

pippo@lab3-linux:~$ whoami
Paolo     pst/12    Apr 27 12:21 (deis32...
```

File System

struttura logica del file system:
tipi di file, persorsi assoluti e
relativi, comando cd

file

- logicamente, un file è una sequenza di bit, a cui viene dato un nome
- in pratica, il file è una astrazione molto potente che consente di trattare allo stesso modo entità fisicamente diverse come: file di testo, dischi rigidi, stampanti, direttori, soft link, la tastiera, il video, etc.

tipi di file

- ordinari: archivi di dati, testi, comandi, programmi sorgente, eseguibili, …
- directory: file gestiti direttamente solo dal S.O., che contengono riferimenti ad altri file
- speciali: dispositivi hardware, memoria centrale, hard disk, …
- FIFO (pipe): file per la comunicazione tra processi
- soft link: riferimenti (puntatori) ad altri file o direttori

file: nomi

- È possibile nominare un file con una qualsiasi sequenza di caratteri (max. 255), a eccezione di '.' e '..' (sono nomi che hanno un significato particolare)
- È sconsigliabile utilizzare per il nome di file dei caratteri speciali, ad es. metacaratteri e segni di punteggiatura

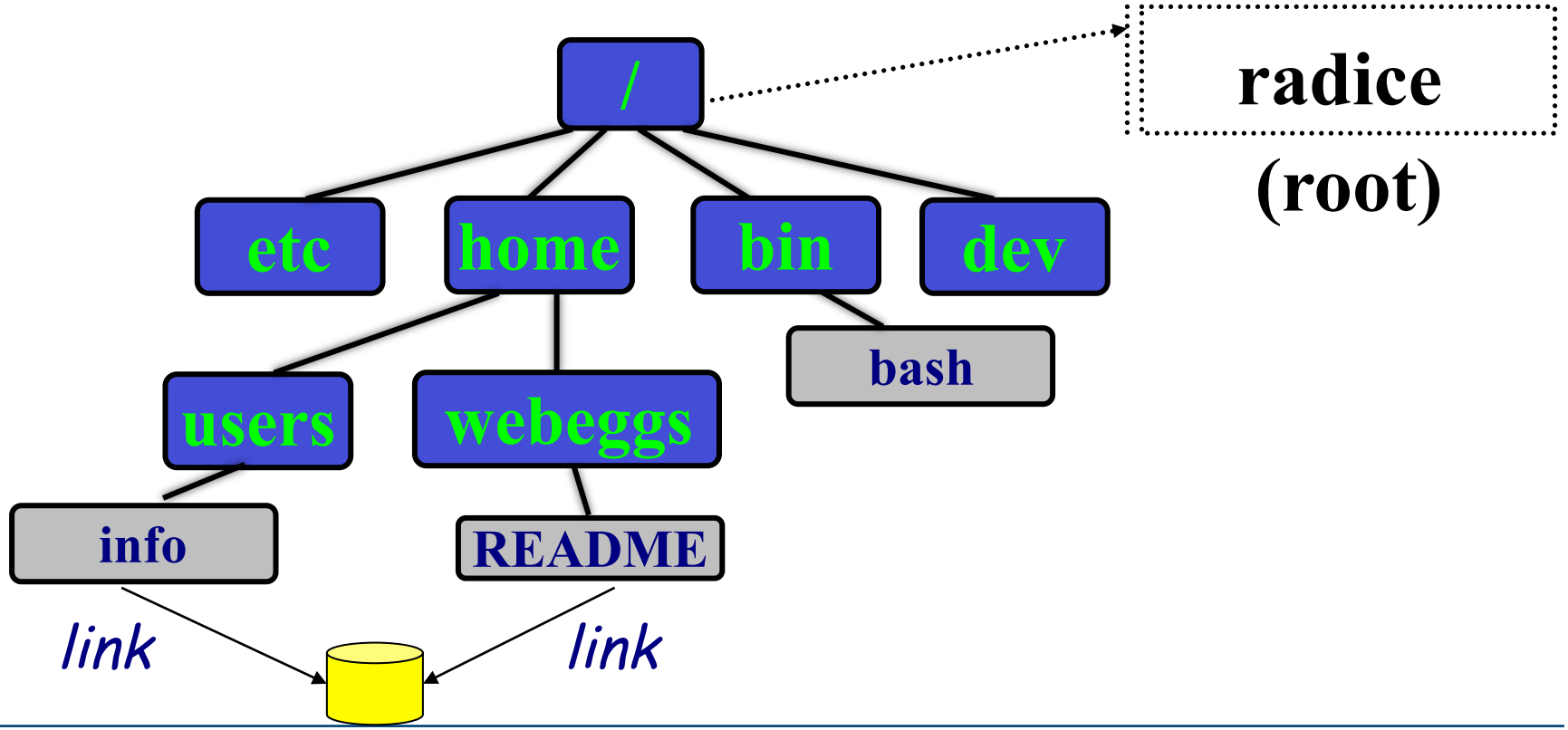
- ad ogni file possono essere associati **uno o più** nomi simbolici (link)

ma

ad ogni file è associato **uno ed un solo descrittore** (i-node) identificato da un intero (i-number)

direttori (directory)

- Il file system è organizzato come un grafo diretto aciclico (DAG).



gerarchie di direttori

- all'atto del login, l'utente può cominciare a operare all'interno di uno specifico direttorio (la sua home). In seguito è possibile cambiare direttorio.
- È possibile visualizzare il percorso completo attraverso il comando **pwd** (print working directory)
- Essendo i file organizzati in gerarchie di direttori, il sistema operativo mette a disposizione dei comandi per muoversi all'interno di essi

Filesystem hierarchy standard (FHS)

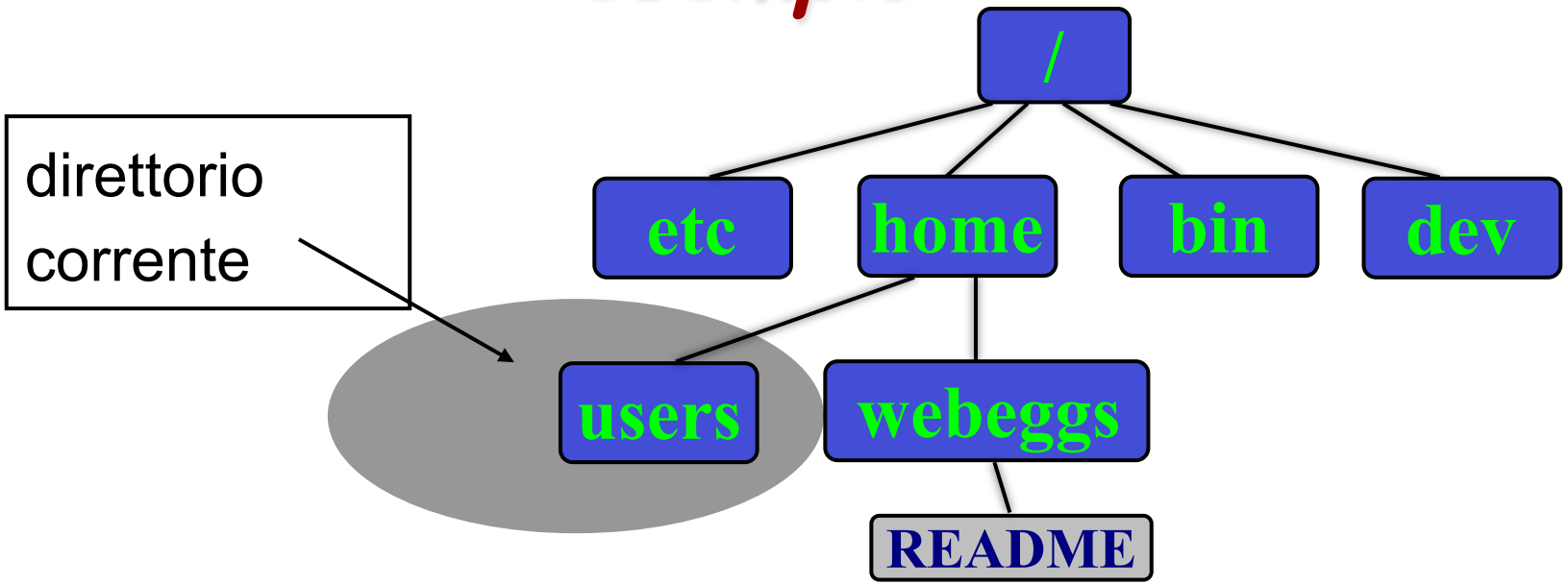
http://www.pathname.com/fhs/

- **/** : Root
- **/bin** : File binari dei comandi essenziali
- **/sbin** : File binari dei comandi di sistema essenziali
- **/home** : Home degli utenti
- **/var** : Dati variabili
- **/boot** : File statici per operazioni di boot (avvio) della macchina
- **/dev** : File dispositivi
- **/etc** : File di configurazione
- **/lib** : Shared libraries e moduli del kernel
- **/media** : *Mount point* per *media* rimovibili
- **/mnt** : *Mount point* per operazioni di mount temporanee di FS
- **/opt** : Software applicativi
- **/tmp** : File temporanei
- ...

nomi relativi / nomi assoluti

- ogni utente può specificare un file attraverso:
  - nome relativo: è riferito alla posizione dell' utente nel file system (direttorio corrente)
  - nome assoluto: è riferito alla radice della gerarchia (/)
- nomi particolari
  - . è il direttorio corrente (visualizzato da pwd)
  - .. è il direttorio 'padre'

nomi relativi / assoluti: *esempio*



nome assoluto: `/home/webeggs/README`

nome relativo: `../webeggs/README`

file

concetto di file, comando ls,
metacaratteri

file

- file = insieme (possibilmente vuoto) di byte organizzati in sequenza e identificato da un nome
- creazione di un file vuoto: **> *nome_file***
- esempio:

```
pippo@lab3-linux:~$ > f1.txt
```

gestione dei file: comando ls

- consente di visualizzare nomi di file
- varie opzioni: es. `ls -l` per avere più informazioni (non solo il nome del file)
- possibilità di usare metacaratteri (*wildcard*)
- Per es. se esistono i file f1, f2, f3, f4,
  - ci si può riferire ad essi scrivendo: f*,
  - oppure: f[1-4]

opzioni del comando ls...

- *sintassi (sempl.):*

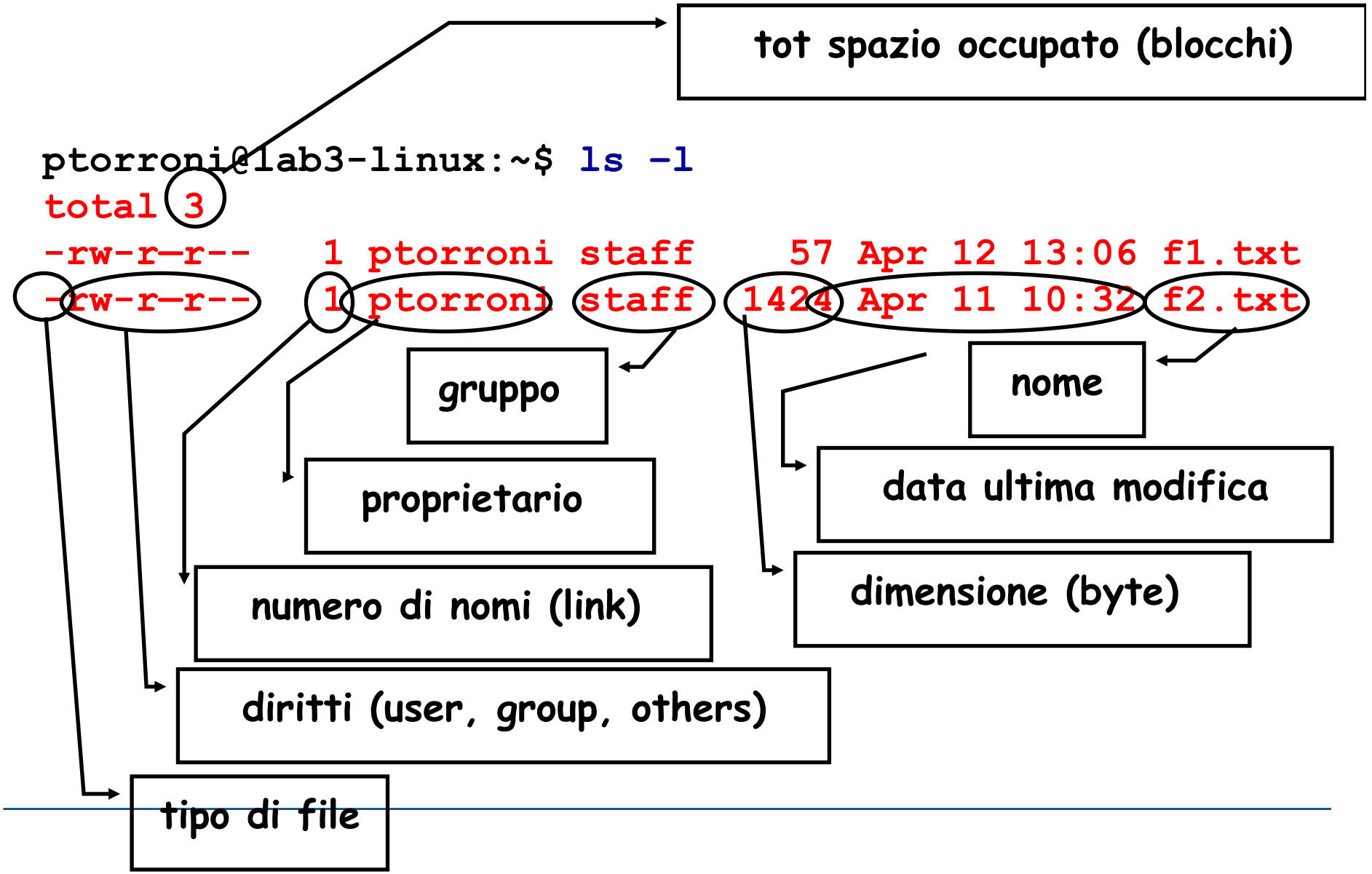
  `ls [-opzioni...] [file...]`

- opzioni:
  - l (long format): per ogni file una linea che contiene i diritti, il numero di link, il proprietario del file, il gruppo del proprietario, l'occupazione di disco (blocchi), la data e l'ora dell'ultima modifica o dell'ultimo accesso, e il nome
  - t (time): la lista è ordinata per data dell'ultima modifica

...opzioni del comando ls

- **u**: la lista è ordinata per data dell'ultimo accesso
- **r** (reverse order): inverte l'ordine
- **a** (all files): fornisce una lista completa (normalmente i file che cominciano con il punto non vengono visualizzati)
- **F** (classify): aggiunge al termine del nome del file un carattere che ne indica il tipo (eseguibile: *, direttorio: /, link simbolico: @, FIFO: |, socket: =, niente per file regolari)

ls -l (esempio)


tot spazio occupato (blocchi)
ptorroni@lab3-linux:~$ ls -l
total 3
-rw-r--r-- 1 ptorroni staff 57 Apr 12 13:06 f1.txt
-rw-r--r-- 1 ptorroni staff 1424 Apr 11 10:32 f2.txt
gruppo
nome
proprietario
data ultima modifica
numero di nomi (link)
dimensione (byte)
diritti (user, group, others)
tipo di file

comandi, opzioni ??

- esiste un manuale on-line (**man**), che si può consultare ogni volta che si hanno dubbi su un comando Linux. Indica:
  - formato del comando (input)
  - risultato atteso (output)
  - descrizione delle opzioni
  - possibili restrizioni
  - file di sistema interessati dal comando
  - comandi correlati
  - eventuali difetti (bugs)
- per uscire dal manuale, digitare 'q' (quit)

il comando passwd

- È possibile cambiare la propria password di utente, mediante il comando **`passwd`**
- Verrà prima chiesta la vecchia password (per motivi di sicurezza)
- Se ci si dimentica della password, bisogna chiedere all'amministratore di sistema (utente *root*)

protezione

proprietà, accessi, bit di protezione

proprietà di file

- Come abbiamo visto, a ciascun utente viene assegnato uno username e una password, e gli utenti sono classificati in gruppi. Es:

  Username: **anna** [User-id: 1530]

  Group: **staff** [Group-id: 22]

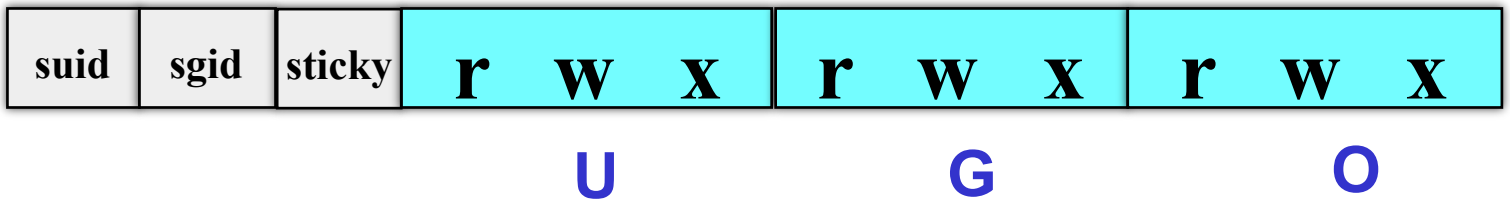
- ad ogni file è associato lo username ed il gruppo dell'utente **proprietario** (inizialmente, chi lo crea)
- In Unix è possibile **cambiare la proprietà di un file** (assegnandola a un altro utente / gruppo): comandi **`chown, chgrp`**

accesso ai file

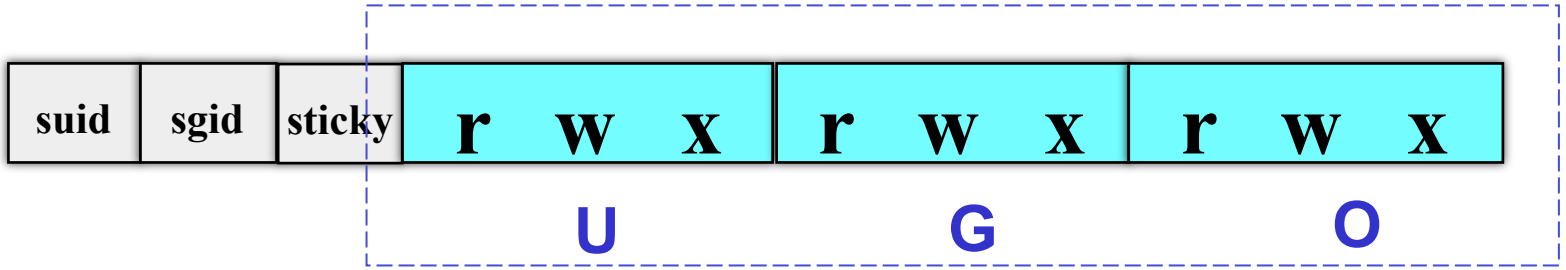
- esistono tre modalità di accesso ai file: **lettura, scrittura, esecuzione**
- il proprietario può concedere o negare agli altri utenti il permesso di accedere ai propri file
- esiste un utente privilegiato (`root`) che ha accesso incondizionato ad ogni file del sistema

bit di protezione

- Ad ogni file sono associati 12 bit di protezione:



Bit di Protezione:
lettura, scrittura, esecuzione

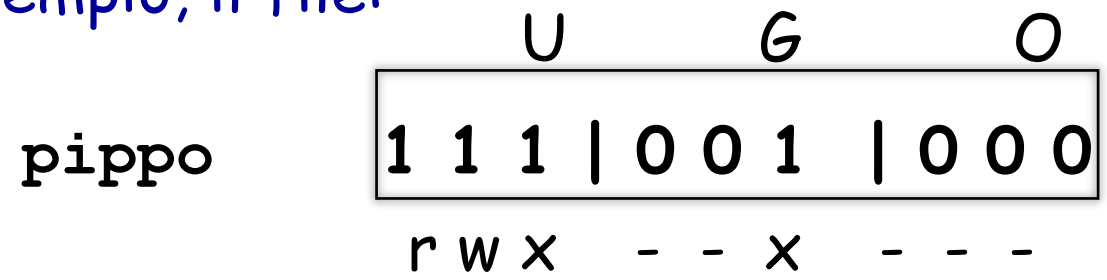


9 bit di lettura (read),scrittura (write), esecuzione(execute) per:

- utente proprietario (User)
- utenti del gruppo (Group)
- tutti gli altri utenti (Others)

bit di protezione:
lettura, scrittura, esecuzione

Ad esempio, il file:



- è leggibile, scrivibile, eseguibile per il proprietario
- è solo eseguibile per gli utenti dello stesso gruppo
- nessun tipo di modalità per gli altri

- formato ottale: 111 => 7; 010 => 2; ... -rwx--x--- => 0710