

Linux / Unix: la shell

utenti e gruppi, shell, comandi

Distribuzioni GNU/Linux

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

- ❑ alcuni esempi: Redhat/Fedora, Slackware, Debian/Ubuntu, Gentoo, SUSE, ecc...
- ❑ Diverse interfacce grafiche applicabili (Unity, Gnome, KDE, Xfce, ecc...)
- ❑ collezioni di **pacchetti software** (es: **unzip**): archivi compressi usati per **automatizzare** e **semplificare** l'installazione di applicazioni (compilazione dei sorgenti, impostazione delle variabili di ambiente, configurazione di permessi, ecc...)
- ❑ **Gestori di pacchetti**: sono pacchetti a loro volta. operazioni di installazione, rimozione, aggiornamento di pacchetti software (es: **rpm** in Redhat/Fedora, **apt-get** in Ubuntu,...)

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 (niente configurazione dual boot o partizionamento del disco), prestazioni ragionevoli

Utenti e gruppi

- Sistema multiutente \Rightarrow 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

Accesso a Linux: *login*

- Per iniziare una sessione bisogna essere in possesso di una *combinazione*:
 - username (es. **x135462**, **d1128493**, ...)
 - password (es. **dfh@2#q**, ****a890**, **aPP&x.**, ...)
- nota: maiuscole / minuscole sono caratteri diversi! (****a890** \neq ****A890**)
- Accesso al sistema: **login: x135462**
Password: *****

NB: per ottenere le credenziali di accesso alle macchine dei laboratori:

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

Prima della 1^o
esercitazione

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**

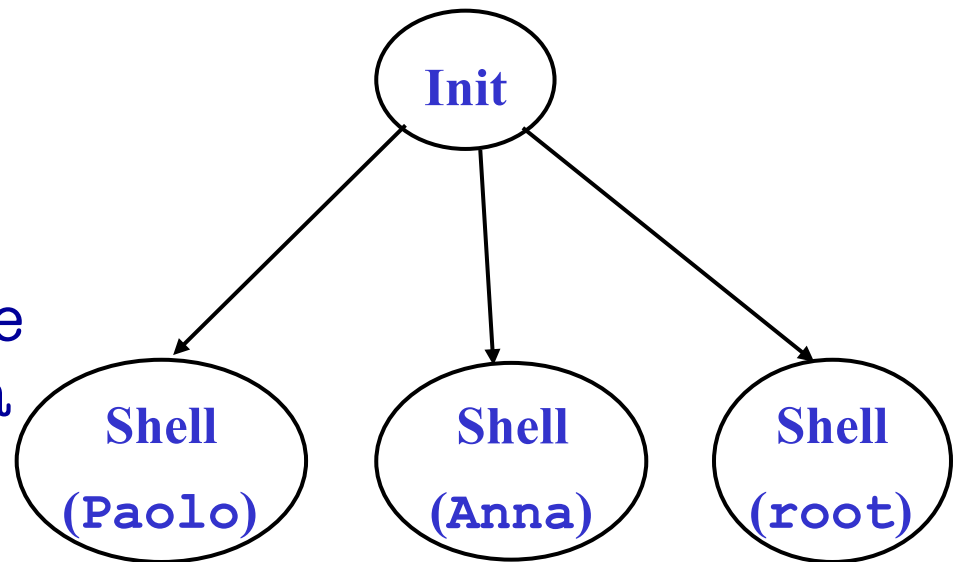
Varie shell di Unix

- 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



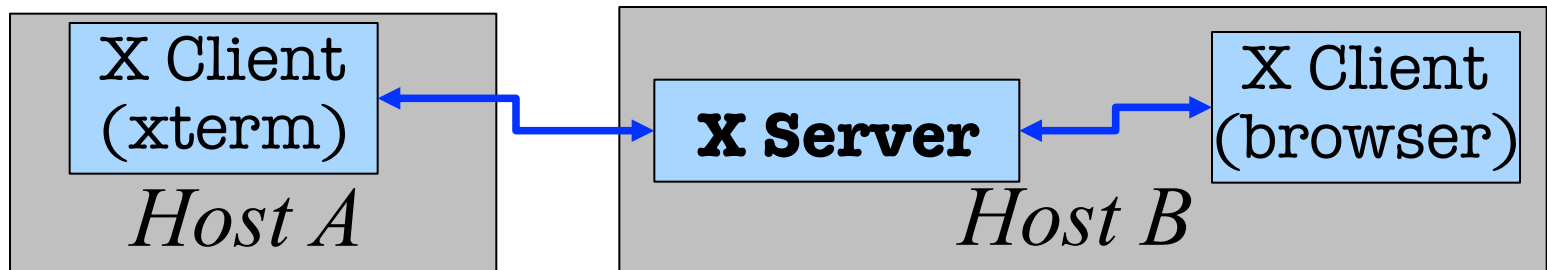
uscita da una shell

- 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**.

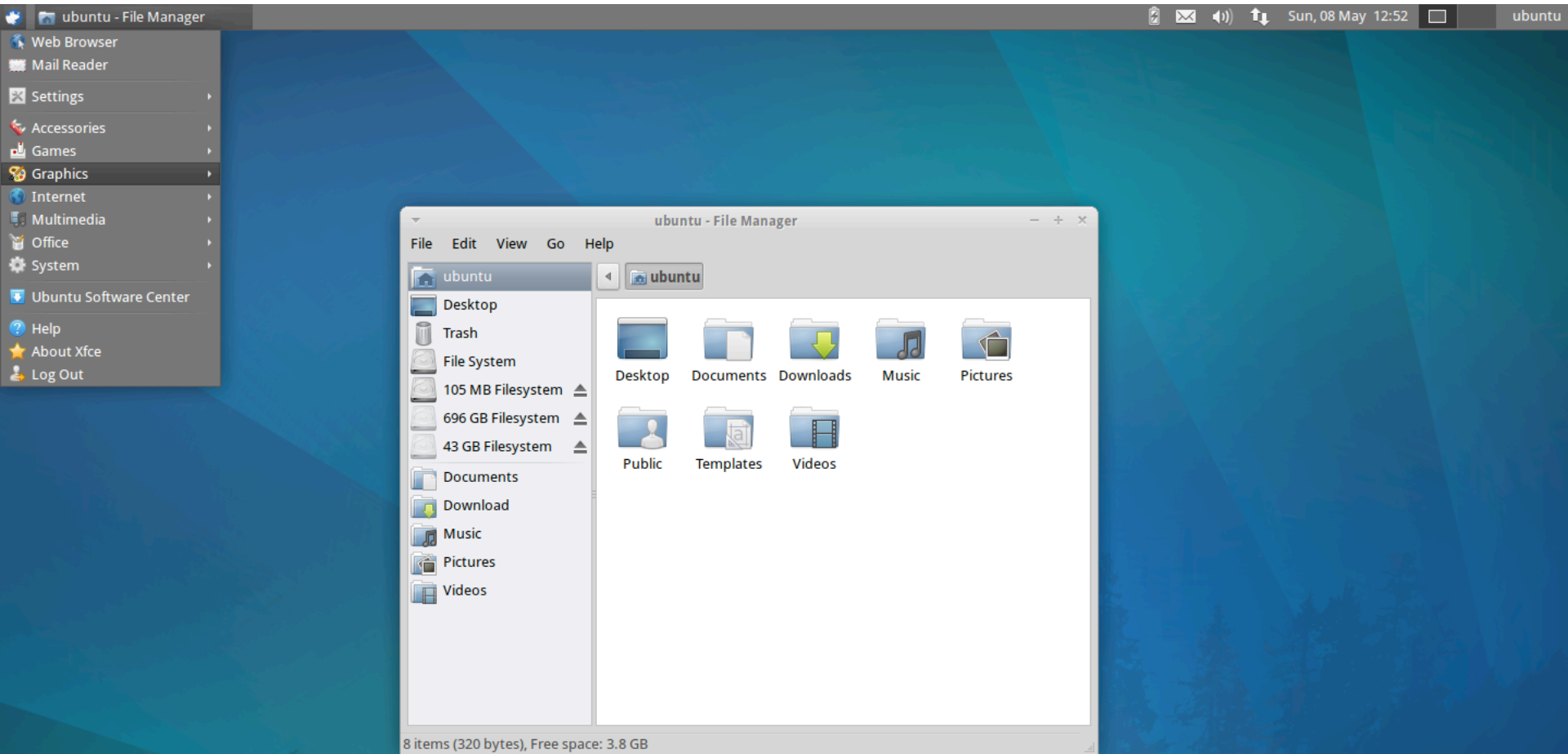
X Window System

Strumento per la gestione di GUI (Graphical User Interface) utilizzato da molti sistemi Unix-like.

- **Fornisce gli strumenti base per l'interazione grafica: disegno o spostamento di finestre, interazione con periferiche di input, ecc..**
- **Non si occupa della resa grafica delle applicazioni. Compito del desktop environment.**
- 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

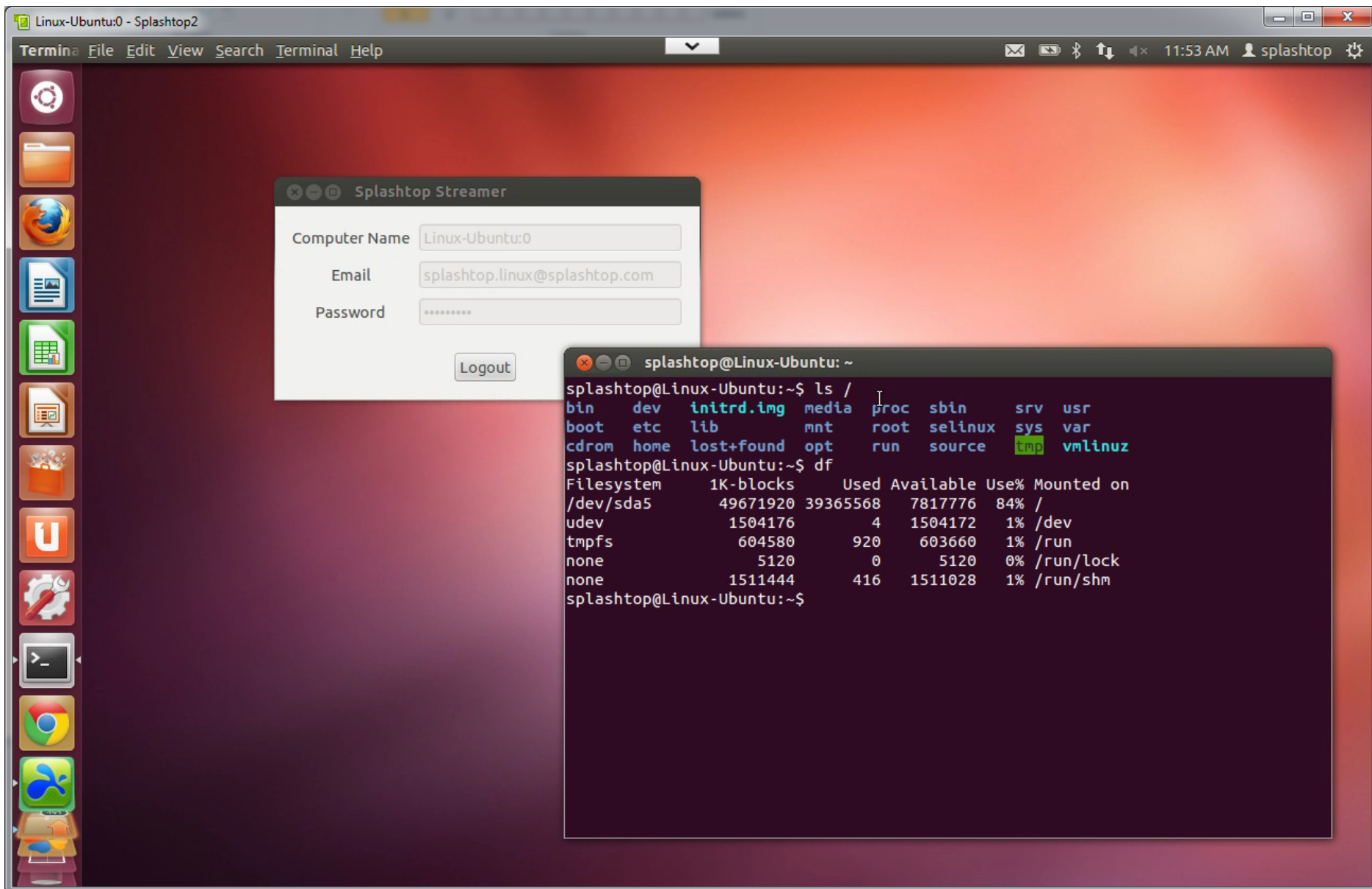


Ubuntu: creare un nuovo shell
(applicazioni -> accessori-> terminale)

Comandi della shell di Unix

standard input, output, error;
tipi di comandi

La Shell di Unix: Comandi

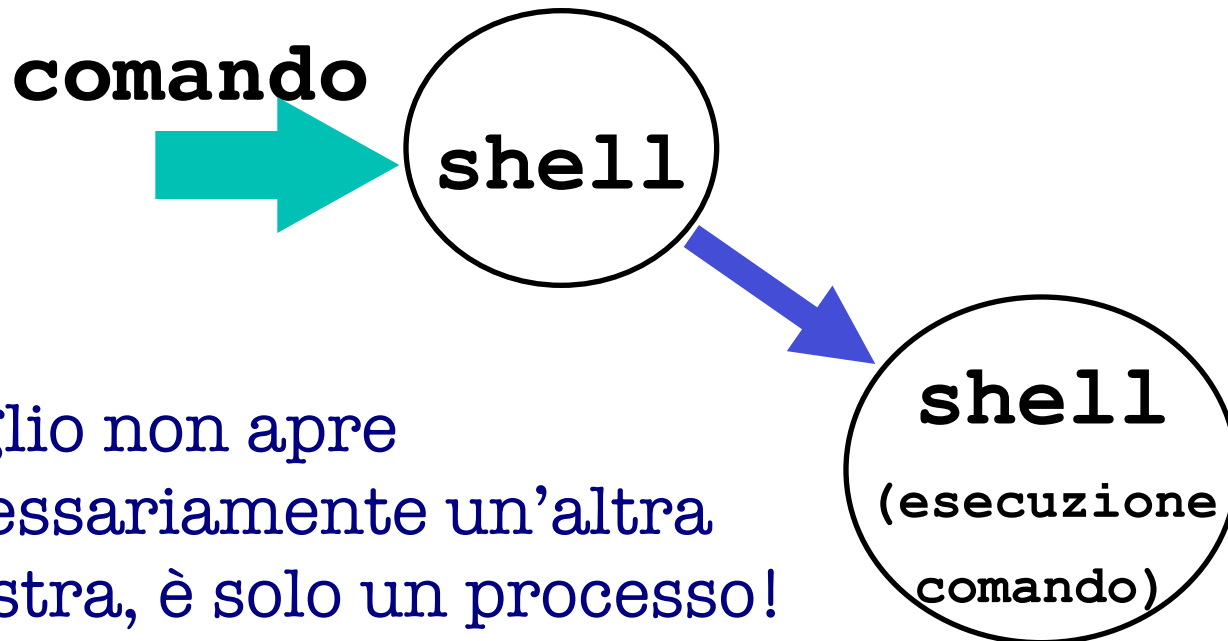


La Shell di Unix: Comandi

- ogni comando richiede al nucleo l'esecuzione di una 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

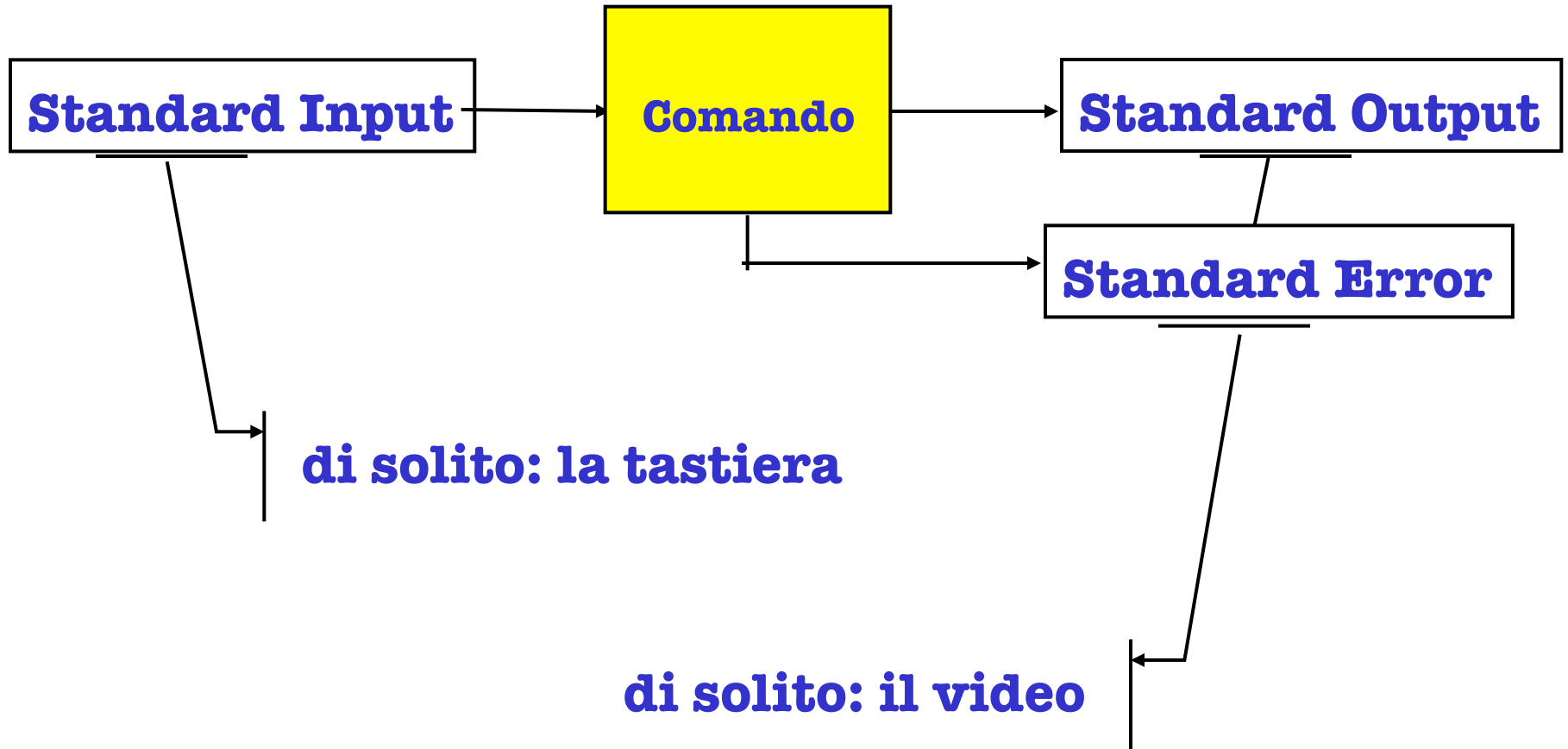
esecuzione di comandi

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



il figlio non apre necessariamente un'altra finestra, è solo un processo!

input / output di un comando



alcuni tipi di comandi

- Alcuni comandi si occupano di **interagire** con il file system:
 - gestione di file e direttori
- altri della **gestione** del sistema:
 - informazioni sulle risorse
 - modifica di dati di sistema

esempi di comandi

provare

```
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, percorsi 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

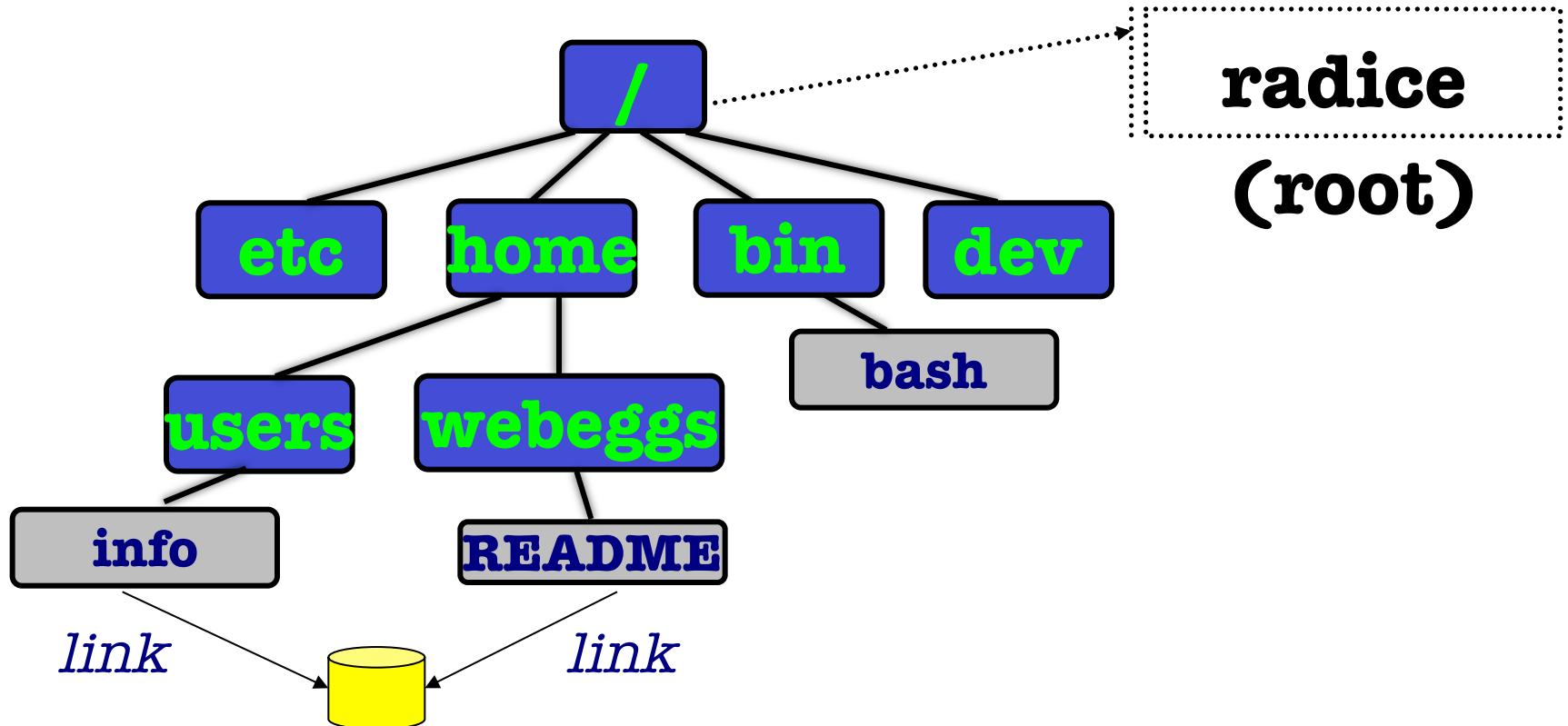
- È possibile nominare un file con una qualsiasi sequenza di caratteri (max. 255), a eccezione di '.' e '..' (sono nomi che hanno un significato particolare)
- È sconsigliabile utilizzare per il nome di file dei caratteri speciali, ad es. metacaratteri e segni di punteggiatura
- ad ogni file possono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

Filesystem hierarchy standard (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	:	<i>Mount point</i> per <i>media</i> rimovibili
/mnt	:	<i>Mount point</i> per operazioni di mount temporanee di FS
/opt	:	Software applicativi
/tmp	:	File temporanei

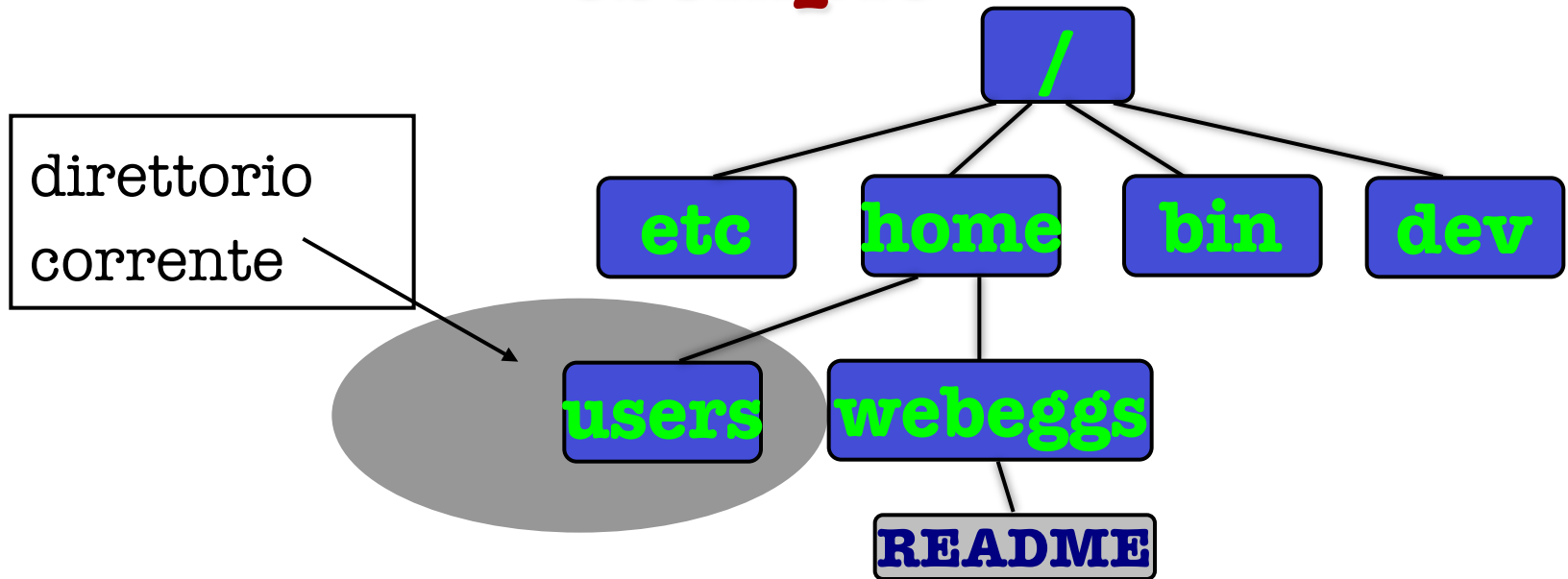
...

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

nomi relativi / nomi assoluti

- ogni utente può specificare un file attraverso:
 - **nome relativo**: è riferito alla posizione dell'utente nel file system (direttorio corrente stampato da **pwd**)
 - **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**

gerarchie di direttori

- Testare i comandi di shell:
 - **pwd**
 - **cd /var**
 - **cd .**
 - **cd ..** => Verificare con **pwd** dove sono ora?

file

concetto di file, comando **ls**,
metacaratteri

file

- file = insieme 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 (semplificata):

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)

provare

tot spazio occupato (blocchi)

```
dloret@lab3-linux:~$ ls -l
```

total 3

-rw-r--r--	1	dloret	staff	57	Apr 12 13:06	f1.txt
-rw-r--r--	1	dloret	staff	1424	Apr 11 10:32	f2.txt

gruppo

proprietario

numero di nomi (link)

diritti (user, group, others)

tipo di file

nome

data ultima modifica

dimensione (byte)

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)
- es: **man ls** , **man pwd**, ecc...
- 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

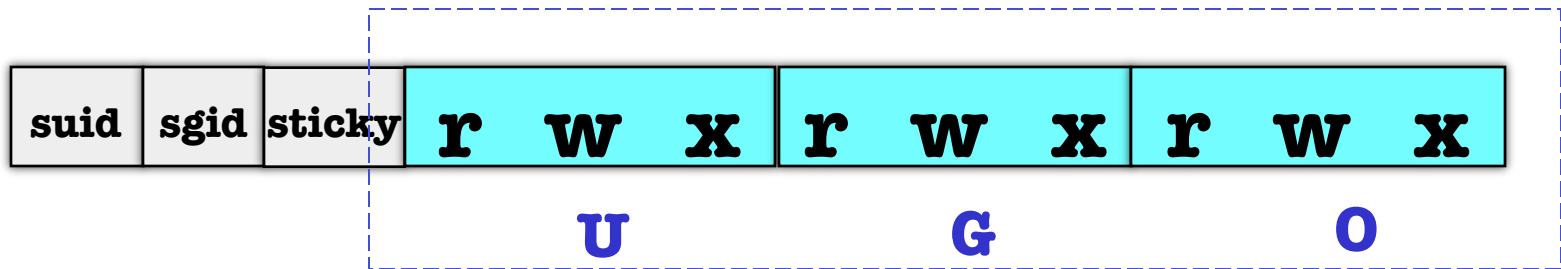
bit di protezione

- Ad ogni file sono associati 12 bit di protezione:

suid	sgid	sticky	r	w	x	r	w	x	r	w	x
			U			G			O		

Bit di Protezione:

lettura, scrittura, esecuzione



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

- utente proprietario (**U**ser)
- utenti del gruppo (**G**roup)
- tutti gli **a**ltri utenti (**O**thers)

bit di protezione:

lettura, scrittura, esecuzione

Ad esempio, il file:

	U	G	O
pip	1	1	1
ppo	0	0	1
o	0	0	0
	r	w	x
	-	-	x
	-	-	-

- 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