Laboratorio di Sistemi Operativi

Corso di Laurea in Informatica

A.A. 2020-2021

Alberto Finzi

Introduzione

Il Sistema Operativo (SO) è lo strato software che controlla le risorse Hardware e fornisce l'ambiente nel quale vengono eseguiti i programmi (kernel).

In senso più largo si considera parte dell'SO anche altro software: utility, shell, applicazioni, librerie, etc.

Tutti i SO forniscono servizi. Eseguire un nuovo programma, aprire un file, allocare la memoria sono esempi di servizi forniti da un sistema operativo.

Descriveremo servizi forniti da varie versioni del sistema operativo Unix.

Il Sistema Operativo Unix

1965 Bell Laboratory (con MIT e General Eletrics) lavora ad un nuovo sistema operativo: Multics. Principali caratteristiche: capacità multi-utente (multi-user), multi-processo (multi-processor) e un file system multi-livello (gerarchico) (multi-level file system).

1969 AT&T abbandona Multics. Ken Thompson, Dennis Ritchie, Rudd Canaday e Doug McIlroy, progettano e implementano la prima versione del file system Unix insieme ad alcune utility. Il nome Unix è di Brian Kernighan: gioco di parole su Multics.

1 Gennaio 1970 Inizio di Unix.

Nel 1973 Unix riscritto in C (Dennis Ritchie)

Caratteristiche:

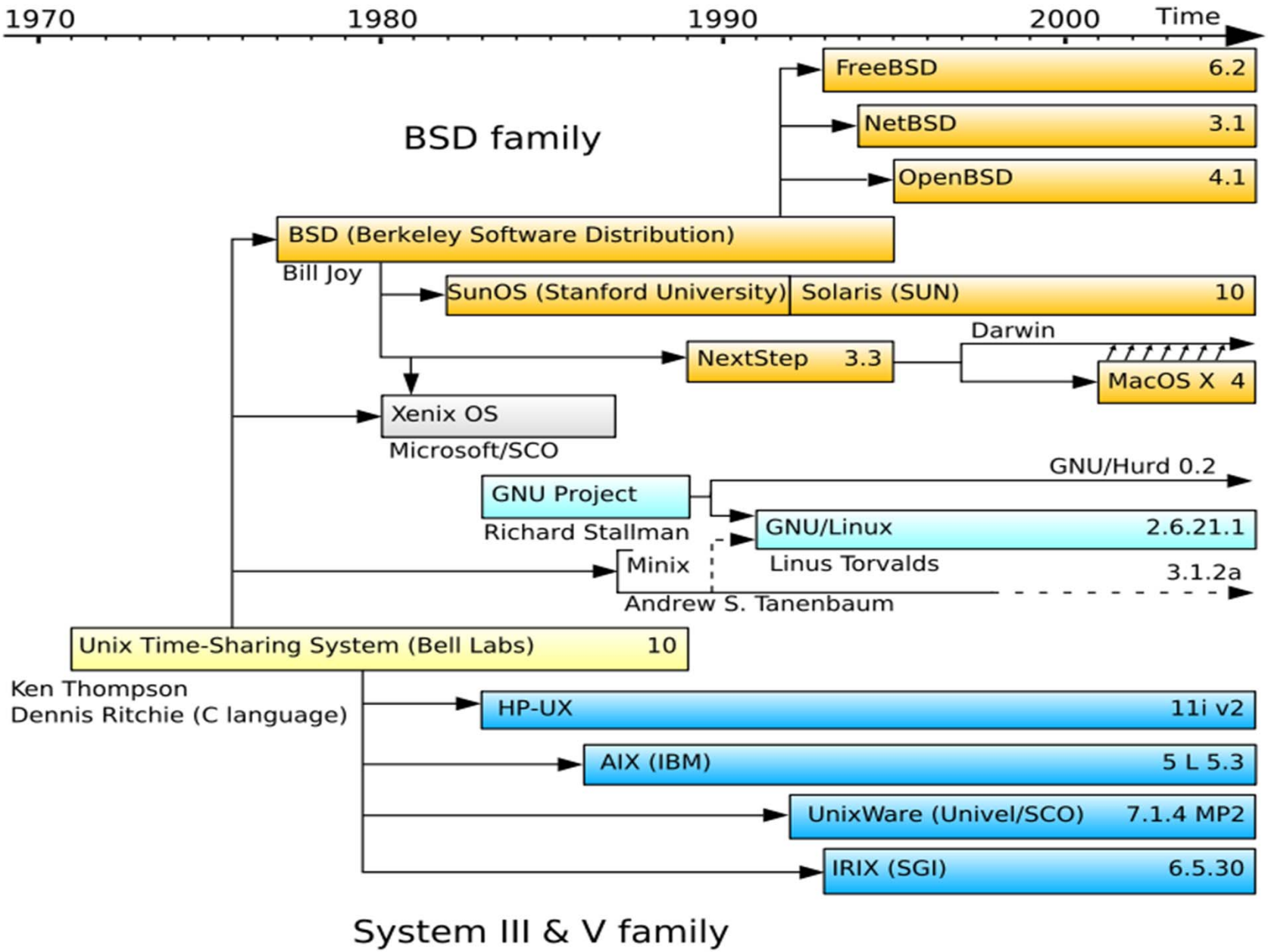
-Architettura semplice ed elegante,

-Ambiente di programmazione (Implementato in C),

-Indipendenza dall'Hardware.

In particolare, SO basato su **kernel**
-il nucleo del SO e l'unica porzione che deve essere adattata all'Hw;
-tutte le altre funzioni poggiano sul nucleo (indipendenti Hw).

Il Sistema Operativo Unix



Architettura Sistema Unix

- Il **nucleo** del sistema (**kernel**) gestisce le risorse essenziali: la CPU, la memoria, le periferiche
- Tutto il resto, anche l'interazione con l'utente, è ottenuto tramite programmi eseguiti dal kernel, che accedono alle risorse hardware tramite delle richieste a quest'ultimo.

Il kernel è il solo programma ad essere eseguito in modalità privilegiata, con il completo accesso all'hardware, gli altri programmi vengono eseguiti in modalità protetta.

Architettura di Sistema

Il kernel si occupa di:

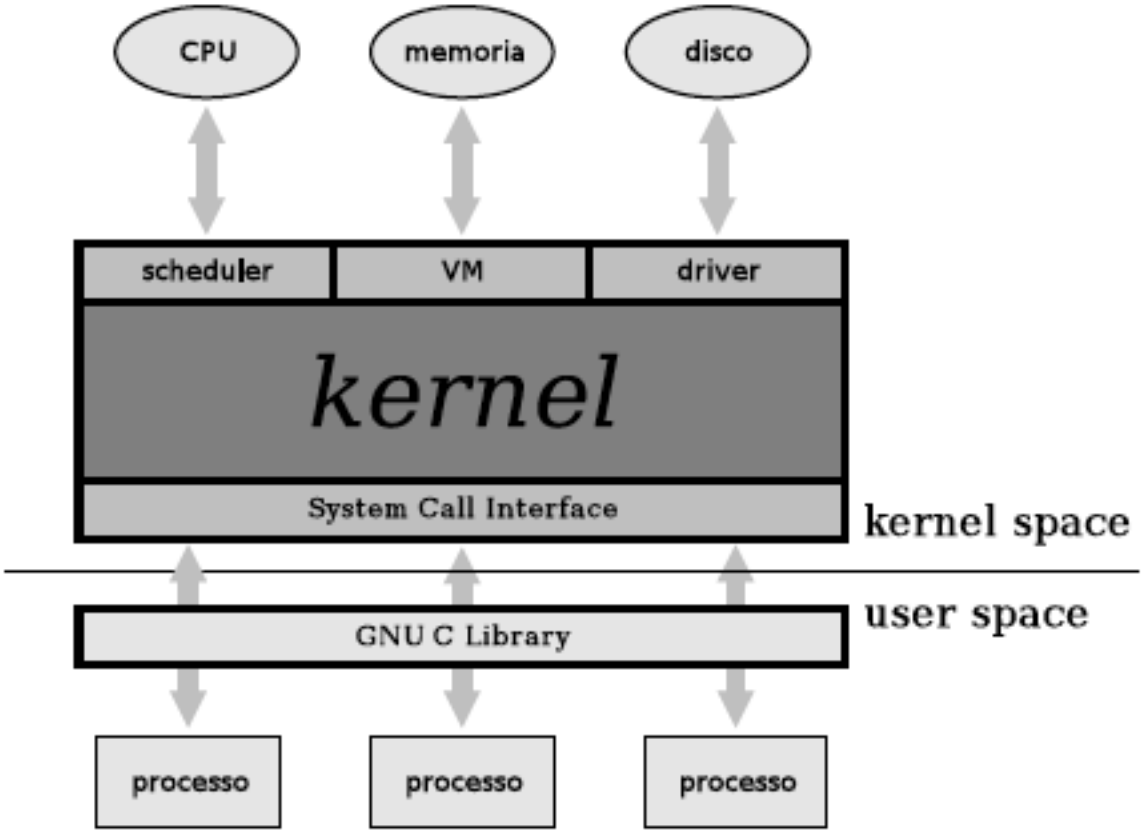
CPU: Una parte del kernel, lo scheduler , si occupa di stabilire, ad intervalli fissi e sulla base priorità, quale “processo” deve essere mandato in esecuzione

Memoria: La memoria viene gestita dal kernel attraverso il meccanismo della memoria virtuale, che consente di assegnare a ciascun processo uno spazio di indirizzi “virtuale” che il kernel, con l’ausilio della unità di gestione della memoria, rimappa automaticamente sulla memoria disponibile.

Periferiche: Le periferiche vengono viste attraverso un’interfaccia astratta che permette di trattarle come fossero file, secondo il concetto per cui “everything is a file” (le interfacce di rete fanno eccezione).

Architettura di Sistema

Le interfacce con cui i programmi possono accedere all'hardware vanno sotto il nome di chiamate al sistema (system call): un insieme di funzioni che un programma può chiamare, per le quali viene generata un'interruzione del processo passando il controllo dal programma al kernel.



Queste chiamate al sistema vengono rimappata in funzioni definite dentro opportune librerie (Libreria Standard del C).

Sistema Multi-utente

- Il kernel Unix/Linux nasce fin dall'inizio come sistema multiutente.
- Il concetto base è quello di utente (user) del sistema, sono previsti *meccanismi* per identificare gli utenti, *permessi* e *protezioni* per impedire che utenti diversi possano danneggiarsi a vicenda o danneggiare il sistema.
- in ogni Unix è presente un utente speciale privilegiato, superuser, il cui username è di norma root, ed il cui uid è zero che identifica l'amministratore del sistema.

Unix Shell

- La shell è un interprete di comandi
- Si interpone tra l'utente ed il sistema operativo
- In sistemi Unix qualsiasi operazione può essere eseguita da una sequenza di comandi shell

Tra le Shell più utilizzate ci sono:

• Bourne shell,	/bin/sh	(Bell Labs)
• Bourne-again shell,	/bin/bash	(Linux)
• Cshell,	/bin/csh	(Berkeley)
• Korn shell,	/bin/ksh	(Bell Labs)
• TENEX C shell	/bin/tcsh	(BBN tech)

Il file /etc/shells contiene l'elenco delle shell installate dall'amministratore e disponibili a tutti gli utenti.

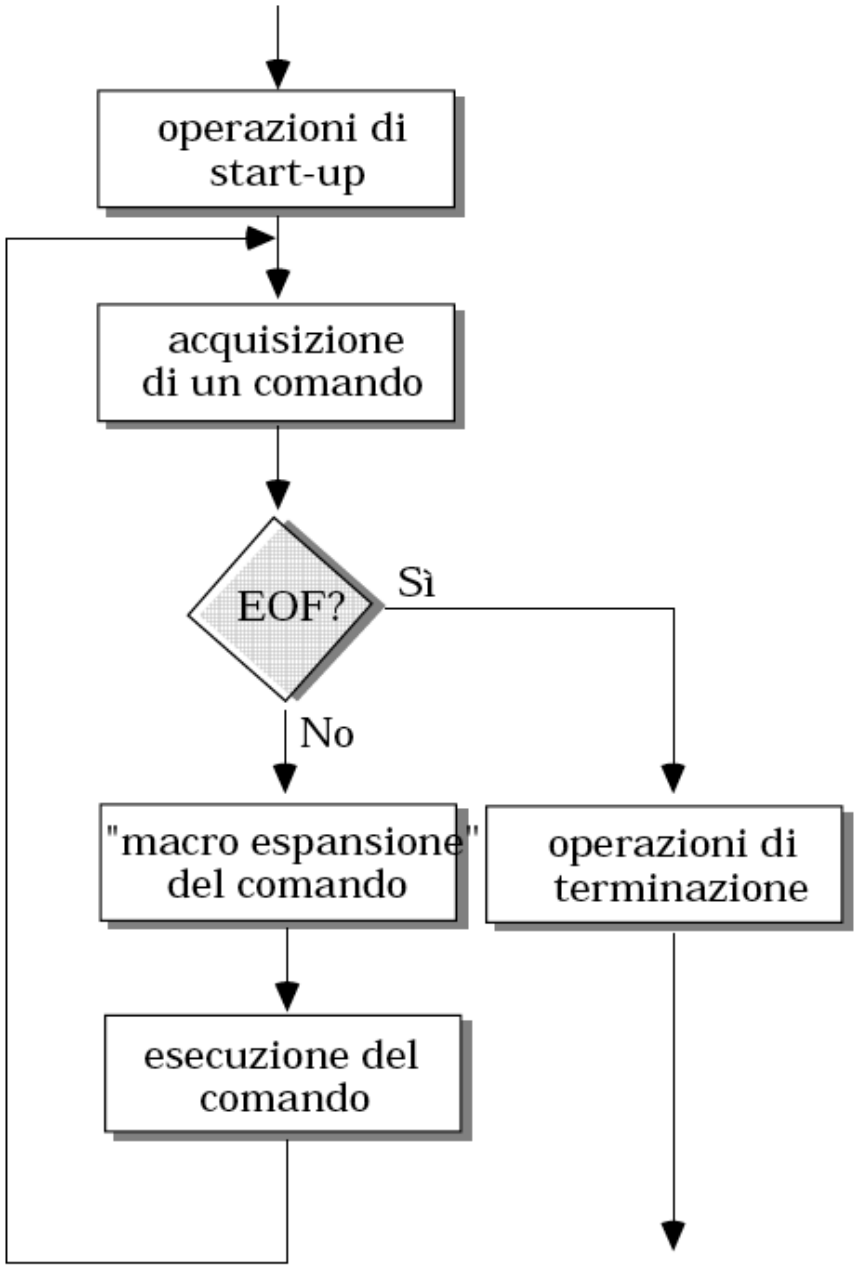
Shell

Programma che interpreta il linguaggio a linea di comando attraverso il quale l'utente utilizza le risorse del sistema. Permette la gestione di variabili e dispone di costrutti per il controllo del flusso delle operazioni.

Viene generalmente eseguito in modalità interattiva, all'atto del login, restando attivo per tutta la durata della sessione di lavoro ed effettuando le seguenti operazioni:

- Gestione del "main command loop";
- Analisi sintattica;
- Esecuzione di comandi ("built-in", file eseguibili) e programmi in linguaggio di shell (script);
- Gestione dello standard I/O e dello standard error ;
- Gestione dei processi da terminale.

Ciclo Esecuzione Shell



Shell Interattiva

- Comunicazione tra utente e shell avviene tramite comandi o script:
- Nome comando bult-in oppure
- Nome di un file eseguibile oppure
- Nome di Script, cioè file ASCII presente nel sistema dotato del premesso di esecuzione.

Classi di Comandi

- reperire informazioni su comandi, programmi, file....
- gestire filesystem
- operare su file e directory
- elaborare testi
- sviluppare software
- operare in remoto
- ....(comandi “di utilità” vari)
- amministrare il sistema
  - utenti e gruppi
  - dispositivi
  - software

Formato dei comandi

comando [argomento ...]

Gli argomenti possono essere:

- opzioni o flag (-)
- parametri

separati da almeno un separatore (di default il carattere spazio)

L'ordine delle **opzioni e', in genere, irrilevante**

L'ordine dei **parametri e', in genere, rilevante**

Attenzione: Unix e' CASE SENSITIVE

Formato dei comandi

Comandi Equivalenti:

ls -l -F file1 file2 file3

ls -lF file1 file2 file3

ls -F -l file1 file2 file3

Comandi NON equivalenti:

cat file1 > file2

cat file2 > file1

ls -l -F file1

ls -f -L file1

Comandi ERRATI:

LS -F -L

CAT file1 > file2

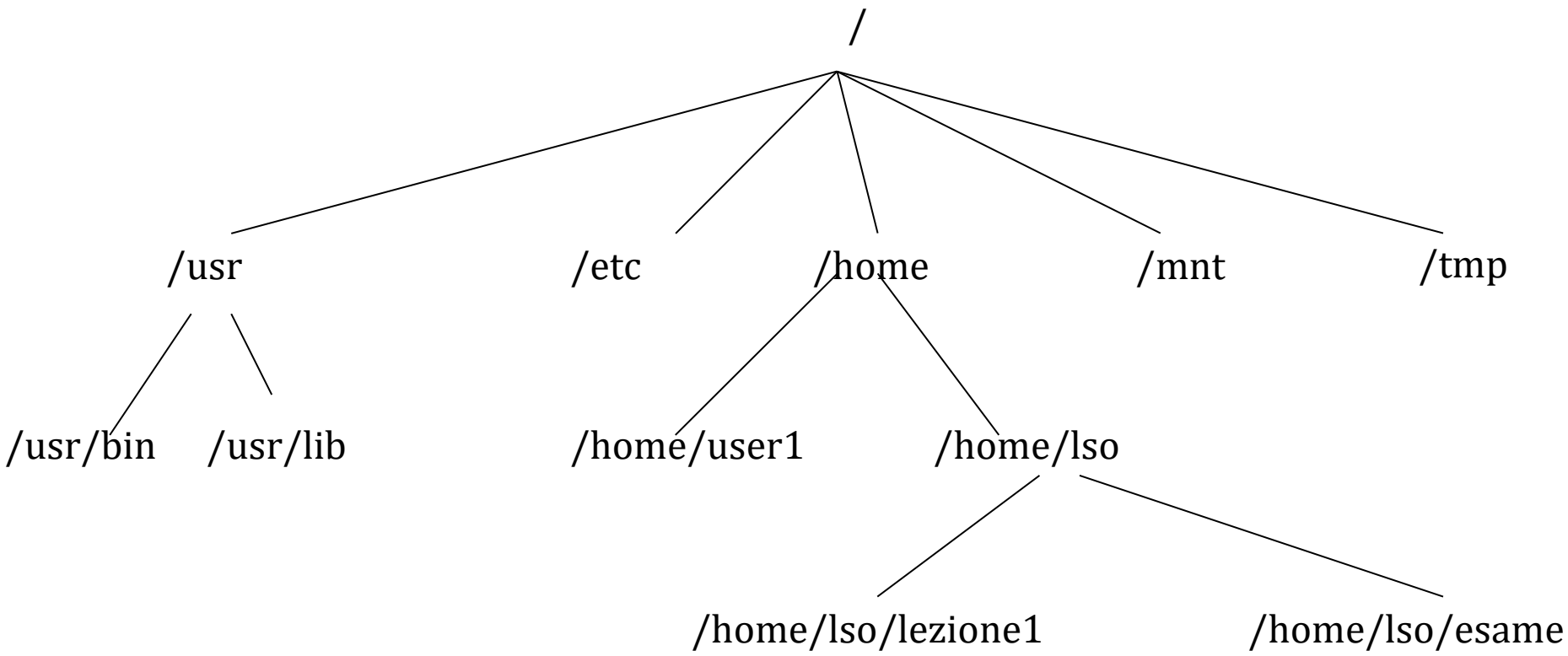
Esempi di Comandi

Comando	Significato
ls	visualizza la lista di file nella directory, come il comando **dir** in DOS
cd directory	cambia directory
passwd	cambia password
file filename	visualizza il tipo di file o il tipo di file con nome filename
cat textfile	riversa il contenuto di textfile sullo screen
pwd	visualizza la directory di lavoro corrente
exit or **logout**	lascia la sessione
man *command*	leggi pagine manuale su **command**
info *command*	leggi pagine Info su **command**
…	

Gestione delle Directory

mkdir	Crea una directory
rmdir	Elimina una directory vuota
pwd	Emette il percorso della directory corrente
ls	Elenca il contenuto di una o più directory
du	Calcola lo spazio utilizzato da una serie di directory e subdirectory

File System di UNIX



mkdir (make directory)

mkdir [options] directory

Crea una directory secondo le opzioni

Alcune opzioni:

-m mode Indica le protezioni per la directory da creare

-p Crea, se assente, l'intero percorso indicato

Esempi:

mkdir -m 600 tracce_esami

mkdir -p lso/11-12/lezione1/

rmdir (remove directory)

rmdir [options] directory

Rimuove la directory indicata.

Le directory possono essere rimosse se
(a) sono vuote e

(b) possibile scrivere nella directory padre.

Opzioni: -p Rimuove l'intero percorso indicato

Esempi:

Crea la directory lezione1 nella directory corrente

rmdir lezione1

Rimuove, dalla directory corrente, il path indicato

rmdir -p lso/11-12/lezione1/

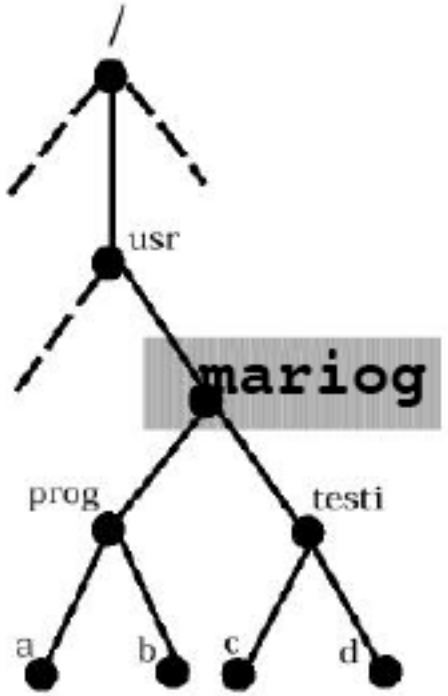
pwd (print working directory)

pwd "printworking directory"

stampa il path della directory corrente

Esempio:

```
% pwd
/usr/mariog
%
```



cd (change directory)

cd [directory]

Cambia la directory corrente a quella indicata.

Se non viene passato nessun argomento, la directory corrente diventa la home directory.

Esempio:

```
lso:~>cd

lso:~>pwd

/home/lso

lso:~>cd esempio/esempiocd

lso:~>pwd

/home/lso/esempio/esempiocd
```

ls (list)

`ls [options] [directory]`

Elenca i file contenuti nella directory specificata.

Se la directory non e' indicata, viene elencato il contenuto della directory corrente.

Alcune opzioni:

-a Elenca anche i file nascosti

-l Formato esteso con informazioni su modo, proprietario, dimensione, etc dei file

-s fornisce la dimensione in blocchi dei file

-t lista i file nell'ordine di modifica (prima il file modificato per ultimo)

-1 elenca i file in una singola colonna

-F aggiunge / al nome delle directory e * al nome dei file eseguibili

-R si chiama ricorsivamente su ogni subdirectory

ls – campi del formato esteso