

# La Shell di UNIX

## 1 – Ambiente di Lavoro

Dipartimento di Informatica  
Università di Pisa  
Largo B. Pontecorvo 3  
56127 Pisa



# Temi

- Ambiente di Lavoro (2h)
- Comandi di UNIX (2h)
- Script di Shell (4h)
- ( Emacs Beginner's HOWTO )

# Overview

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi

## Storia breve di UNIX (1)

**MULTICS** - Alla fine degli anni'60, General Electric, MIT e Bell Labs hanno insieme lanciato un progetto per sviluppare un sistema operativo **multi-utente** e **multi-tasking** chiamato Multiplexed Information and Computing System **MULTICS**.  
- abbandonando in 1969.

**UNICS** - Ken Thompson dei Bell Labs ha sviluppato un sistema operativo più semplice, Uniplexed Information and Computing System **UNICS** nel 1969.

## Storia breve di UNIX (2)

**UNICS** (poi UNIX) è stato progettato con le seguenti caratteristiche:

- l'ambiente di programmazione;
- interfaccia utente che renda facile;
- design modulare;
- gerarchia del File System (ad albero);
- semplice interfacciamento con i dispositivi;
- sistema *multi-utente* e *multi-processo*: molti utenti possono collegarsi al sistema ed eseguire istanze di programmi (processi) simultaneamente;
- architettura indipendente e trasparente all'utente.
- comandi efficienti in termini di spazio, ad es. `ls`, `cp`, `mv`, ecc.

## Storia breve di UNIX (3)

- 1973** - Unix riscritto prevalentemente in **C**, un linguaggio di prog. ad alto livello da **Dennis Ritchie**.
- 1974** - Unix si diffonde prevalentemente in campo accademico (licenza educativa).
- 1978** - divisa in due distinti linguaggi, Bell Labs' System V (**SYS V**) e Berkeley Software Distribution (**BSD**).
- per l'università a costo contenuto
  - senza alcun tipo di supporto tecnico, né alcuna garanzia.
- 1992** BSD "ripulito" dal codice proprietario e rilasciato come **386BSD**.
- 1995** fu nato **freeBSD** (solo per un anno) .
- La versione final fu la **4.4BSD-lite** con restrizioni piccoli.

## Storia breve di UNIX (4)

**1985** Richard Stallman (NY, USA) ha fondò la Free Software foundation (FSF)

- **La filosofia** - Gli utenti possono distribuire e modificare il software a seconda delle proprie esigenze e distribuire anche le modifiche fatte.
- **La pratica** - Contratto di licenza d'uso, la [General Public License \(GPL\)](#)
- **Gnu's Not Unix (GNU)** - Sistema operativo completo alla base di C.

**1990** Lo sviluppo del kernel di GNU, **The Hurd** fu iniziato.



## Storia breve di UNIX (5)

**1991** - Studente finlandese Linus Torvalds creò un kernel *unix-like* (conforme alla Single Unix Specification) per PC e lo chiamò **Linux**.

- Interfaccia utente simile al BSD e ad altri processi di avvio in stile SYS V-style
- conforme a POSIX (Portable Operating System Interface) una famiglia di standard, specificata dall'IEEE.
- open source (GNU-GPL) - in quanto chiunque può trovare e correggere bug.
- Tante distribuzioni: Debian, Ubuntu; Redhat, Fedora, CentOS; Slackware, SUSE.

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi

# Il sistema operativo (1)

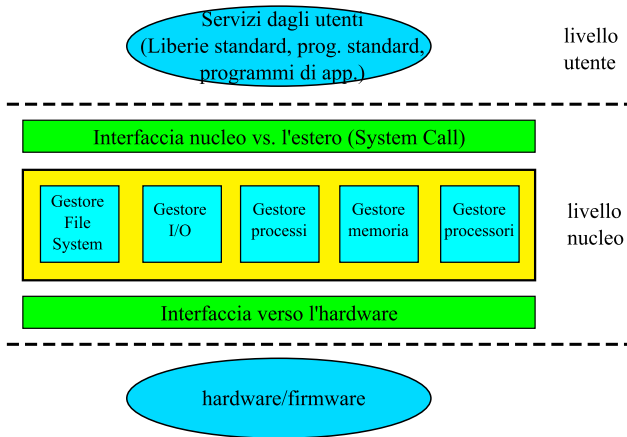
## Definizione

- Un SO è un programma che **controlla l'esecuzione di programmi applicativi** e agisce come **interfaccia tra le applicazioni e l'hardware** del sistema.
- Elemento principale del software di sistema
- Organizzato logicamente a strati ... ogni 'strato' risolve uno specifico problema

## Obiettivi

- Un SO cerca di utilizzare in modo **efficiente** le risorse
- Un sistema operativo dovrebbe **semplificare** l'utilizzazione dell'hardware.

## Il sistema operativo UNIX (2)



# Concetto di Processo

## Programma

- entità **statica** composta da codice eseguibile dal processore, tipicamente codificato in una serie di linee di codice scritte in un certo linguaggio di programmazione

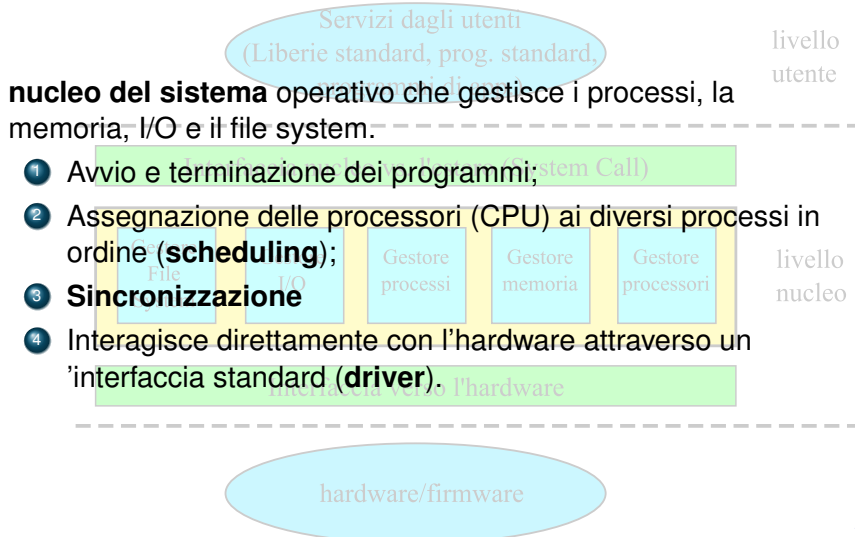
## Processo

- entità **dinamica** caricata su memoria RAM che corrisponde al programma in esecuzione, composto da:
  - codice** (text) del programma eseguito
  - dati** (variabili globali)
  - program counter** (un registro della CPU che punta all'indirizzo di memoria della prossima istruzione)
  - stato dei **registri** di CPU
  - lo **stack**, un tipo di dato astratto che funge da raccolta di elementi, con due operazioni principali: push/pop.

N.B:

ad un programma possono corrispondere diversi processi

# Il Kernel



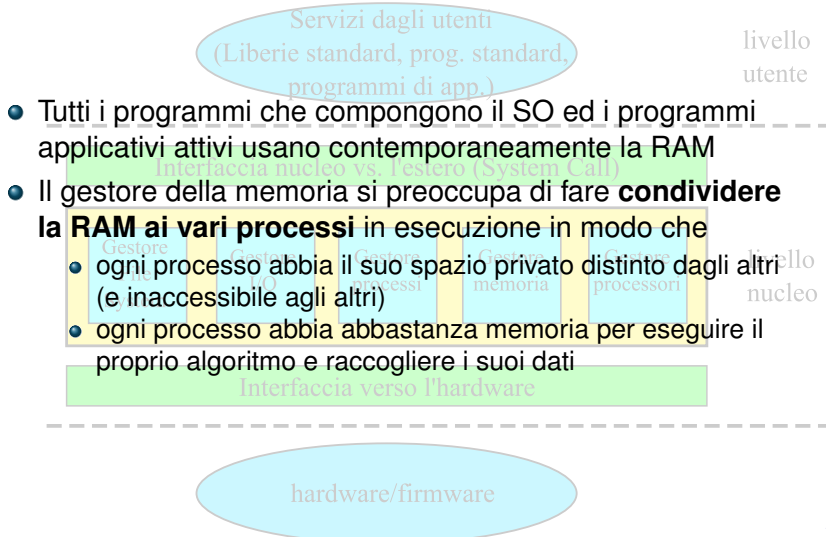
# Gestore dei processori e dei processi (Scheduler)

- controlla la **sincronizzazione, interruzione e riattivazione** dei programmi in esecuzione che viene assegnato ad un processore; livello utente
- Obiettivi:
  - massimizzare l'uso della CPU
  - massimizzare il numero di elaborazioni in un certo intervallo di tempo
- Distinzione di parallelismo in:
  - **Multithreading**: i singoli thread condividono lo stesso spazio d'indirizzamento, la stessa cache. livello nucleo
  - **Multiprocessing**: i processi hanno spazio d'indirizzamento individuale sullo stesso processore o processori diversi.

N.B:

possono esserci più thread in un processo.

# Gestore della Memoria





# Protezione HW: Livello kernel / Livello utente

I sistemi multiprogrammati e multiutente richiedono la presenza di **meccanismi di protezione**.

## Modalità kernel/supervisore/privilegiata

- i processi in questa modalità hanno accesso a tutte le istruzioni, incluse quelle **privilegiate**, che permettono di gestire totalmente il sistema

## Modalità utente

- i processi in modalità utente non hanno accesso alle istruzioni privilegiate

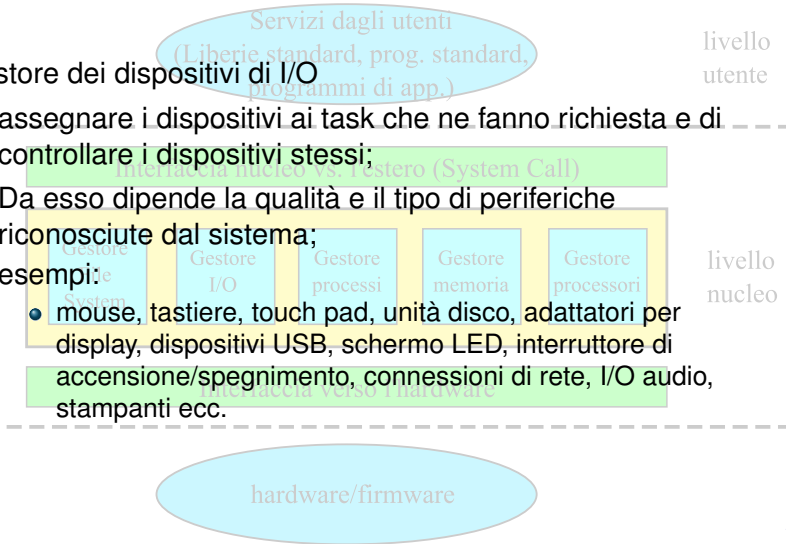
**"Mode bit" nello status register per distinguere tra i due**



# Gestore I/O

## Il gestore dei dispositivi di I/O

- assegnare i dispositivi ai task che ne fanno richiesta e di controllare i dispositivi stessi;
- Da esso dipende la qualità e il tipo di periferiche riconosciute dal sistema;
- esempi:
  - mouse, tastiere, touch pad, unità disco, adattatori per display, dispositivi USB, schermo LED, interruttore di accensione/spegnimento, connessioni di rete, I/O audio, stampanti ecc.



# Gestore File

## Il gestore del file system

- gestisce le informazioni memorizzate sui dispositivi di memoria di massa;
- garantisce la **correttezza e la coerenza** delle informazioni;
- Nei sistemi multi-utente, deve **proteggere i propri dati dall'accesso** da parte di altri utenti non autorizzati.

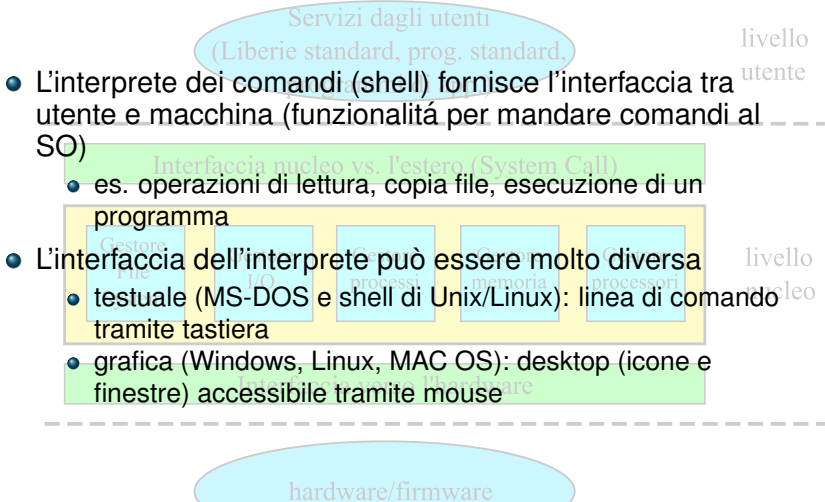
## Le funzioni tipiche:

- Fornire un meccanismo per l'identificazione dei File;
- Fornire opportuni metodi per accedere ai dati;
- Rendere trasparente la struttura fisica del supporto di memorizzazione;
- Implementare meccanismi di protezione dei dati.

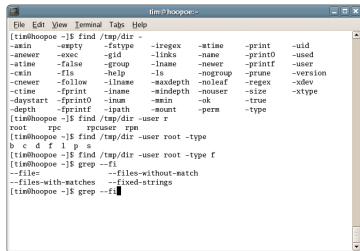
livello  
utente

livello  
nucleo

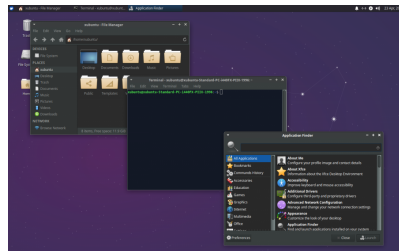
# Interfaccia Utente



# La Shell UNIX



```
tim@hoopoe:~$ find /tmp/dir -
-amin      -empty      -fstype     -iregex     -atime      -print      -uid
-amin      -exec         -gid        -links      -name       -print0     -used
-atime     -false       -group      -lname      -nouser     -printf     -user
-cmin      -fls         -help       -ls         -nogroup    -prune      -version
-cnewer    -follow      -ilname     -maxdepth   -noleaf     -regex      -xdev
-ctime     -fprint      -iname      -sindepth   -nouser     -size       -xtype
-daystart  -fprint0     -inum       -smmin      -ok         -true       -perm
-depth     -fprintf     -ipath      -sount      -type
[ti@hoopoe ~]$ find /tmp/dir -user r
root      rpc      rpcuser  rpm
[ti@hoopoe ~]$ find /tmp/dir -user root -type
b c d f l p s
[ti@hoopoe ~]$ find /tmp/dir -user root -type f
[ti@hoopoe ~]$ grep --fl
--file=
--files-with-matches
--files-without-match
--fixed-strings
[ti@hoopoe ~]$ grep --fl
```

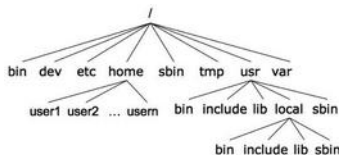


- Un programma che fornisce un'interfaccia alle funzionalità del sistema operativo;
- fornisce un **linguaggio di programmazione**, con variabili, funzioni e costrutti per il controllo del flusso.
- è un interfaccia grafica **Graphical User Interface - GUI** o **testuale**.

- 1 Legge l'input (dalla tastiera del terminale utente, da una stringa passata come argomento (opzione **-c**) o da un file di **script**).
- 2 Scompone l'input in parole ed operatori, rispettando le regole di quoting(). In questo passo viene anche effettuata l'espansione degli **alias**.
- 3 Esegue vari tipi di **espansioni** (ad es. espansione di percorso o espansione della storia).
- 4 Esegue ogni **ridirezione necessaria** e rimuove gli operatori di ridirezione ed i loro operandi dalla lista degli argomenti.
- 5 Esegue il comando.
- 6 Aspetta che l'esecuzione termini e quindi ne rileva lo stato

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi

## Il file system di UNIX (1)



- è il meccanismo fornito dal sistema operativo che regola l'organizzazione delle informazioni sui dispositivi (disco, cd-rom, dvd, ...);
- è paragonabile alla struttura rovesciata di un **albero** (In realtà è un grafo).
- è **omogeneo**: tutto è un file (documenti, sorgenti di programmi, applicazioni, immagini...). Quattro categorie di file: *file ordinari*, *direttori*, *dispositivi* e *link*.



## Il file system di UNIX (2)

### Il file & la directory

Ogni nodo dell'albero è un file o una directory di file. L'ultima può contenere altri file e directory.

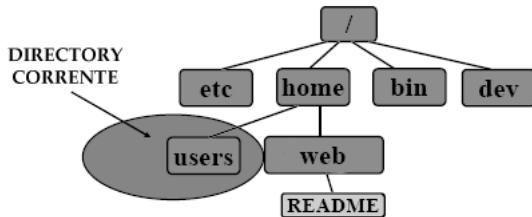
- Un **file** è una sequenza non strutturata di bytes (unità logica di memorizzazione);
- Una **directory** è un file che indicizza altri file.

Un file, identificato da un **path name**, ha i seguenti attributi: tipo, permessi (diritti di accesso), nome utente proprietario, nome gruppo proprietario, dimensione, data di creazione, ultima modifica, ultimo accesso. Lo può essere

- **assoluto**, riferito alla radice della gerarchia ( / ), oppure
- **relativo**, riferito alla posizione dell'utente nel file system.

# Il file system di UNIX (3)

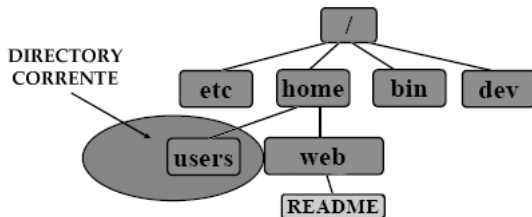
## Il file & la directory



- Il livello più alto è la **radice (root)** ' / '
- La directory corrente di lavoro è ' . '
- La directory padre è ' . . '
- La directory home è ' ~ '

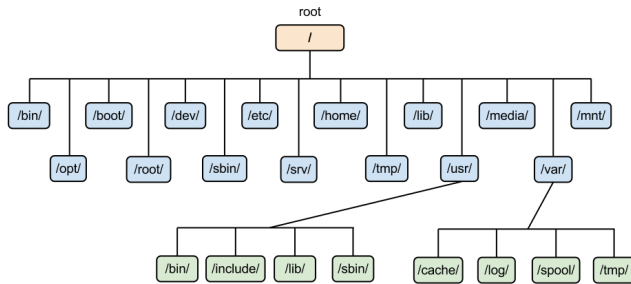
# Il file system di UNIX (4)

## Il file & la directory



- **Nome assoluto:** `/home/web/README`
- **Nome relativo:** (basata sulla `users`) `../web/README`
- (basata sulla `home`) `../web/README` or `~/web/README`

# Le Directory (1)



- Esiste uno standard (FHS) per l'organizzazione dell'albero delle directory per il FS UNIX:  
[https://en.wikipedia.org/wiki/Filesystem\\_Hierarchy\\_Standard](https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard)
- Diverse distribuzioni non sono del tutto conformi a FHS, ogni distribuzione fa proprie personalizzazioni
- Per alcune directory non c'è una distinzione netta per i loro contenuti (es. /lib, /usr/lib, oppure /bin, /usr/bin)
- Manual entry: `man 7 hier` e `man 7 file-hierarchy`

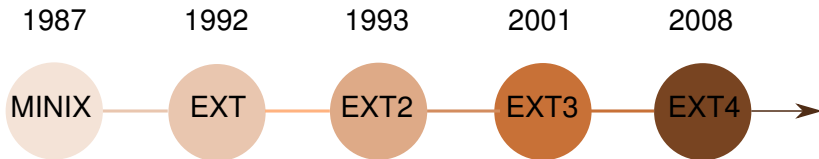
# Le Directory (2)

## Directory principale

cartella	Cosa contiene
/	Cartella root dell'albero di directory del FS
/home	Contiene le home directory degli utenti (es. /home/riccardo /home/marta )
/etc	Contiene i file (e directory) di configurazione del sistema
/usr	Contiene eseguibili e librerie associate a SW installati dall'ammin. (tipicamente in /usr/local) o dalla distribuzione SW (Debian/Ubuntu, CentOS, RedHat, ...)
/var	Contiene file variabili come i log di sistema, spool, cache,...
/dev	Contiene i file associati alle periferiche (devices) e pseudo-devices (file speciali)
/bin	Contiene gli eseguibili di uso comune per gli utenti. Non c'è una distinzione netta tra /bin, /usr/bin e /sbin
/sbin	Contiene gli eseguibili utilizzati prevalentemente dall'amministratore del sistema
/opt	Cartella opzionale utilizzata per installare pacchetti e software di terze parti
/lib*	Contiene le librerie di sistema (a volte si distingue tra /lib, /lib32 e /lib64)
/proc	Contiene una gerarchia di file speciali che rappresentano lo stato attuale del sistema. Contiene informazioni sui dispositivi e sull'HW della macchina (cpu, memoria, cores, interrupts, ...)
/mnt	Directory usata per il mount point di FS temporanei (es. quando si monta una chiavetta USB)
/tmp	Directory contenente file temporanei del sistema e degli utenti

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi

# L'evoluzione del file system Linux



## Blocchi (1)

- **Struttura logica**
  - I dati sono organizzati in unità logiche di lunghezza fissa ma arbitraria dette blocchi logici [byte];
- **Struttura fisica**
  - I dati sono organizzati in unità fisiche di lunghezza fissa e dipendente dal dispositivo dette blocchi fisici
  - Dimensioni tipiche dei blocchi di unità a disco rigido variano da 32 a 4096 bytes, tipicamente 512);

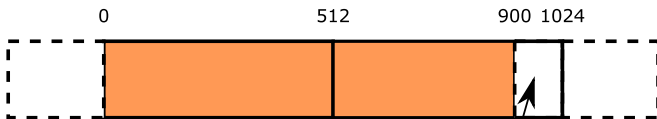
N.B:

La struttura logica e fisica sono differenti.



## Blocchi (2)

- Si consideri un file lungo 1000 byte ed un disco con blocchi da 512 bytes:



Questa parte del blocco (124 byte) viene sprecata  
e non è utilizzabile da altri file

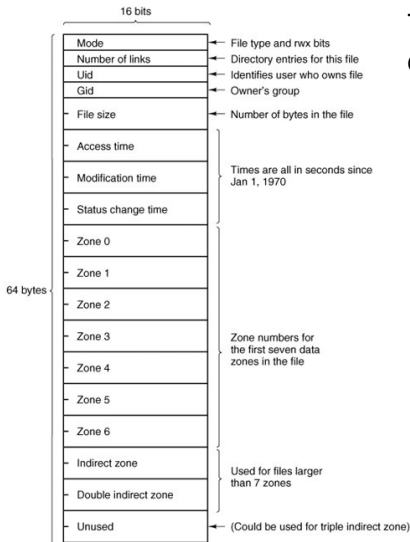
N.B:

Tale inconveniente è detto di "frammentazione interna".

## Index-nodes (I-nodes)

- Il file system UNIX è basato sugli **i-node**.
- ad ogni file è associato uno ed un solo descrittore (i-node), univocamente identificato da un intero (i-number).
- I-node sono **metadati** su un file.
- Usiamo un inode per trovare i blocchi associati a un file, per vedere quanto è grande il file, per dare a un file i permessi e il tipo, etc.
- Un i-node nel file system **Minix 3** [A. Tanenbaum-1987] ha la seguente struttura e richiede **64 byte**.

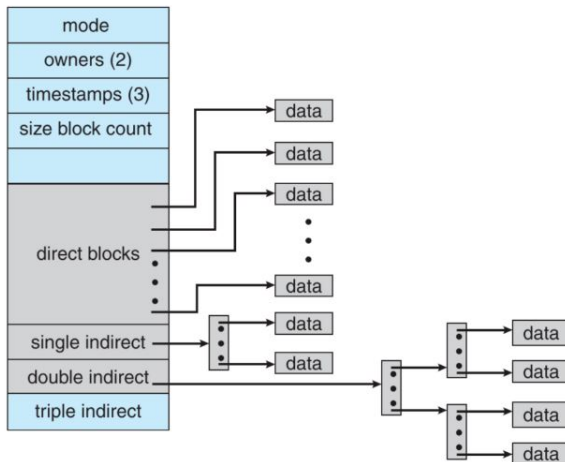
# Indirizzamento (1)



Tra gli **attributi** (64 Byte) contenuti nell' i-node ci sono

- **tipo** di file
- Numero di (hard) **link** al file
- Proprietario (**UID**)
- il suo gruppo (**GID**)
- Lunghezza del file [Byte]
- Data e ora in cui il file è stato **accesso**, **modificato** e **creato** dopo il 1/1/1970.
- 10 **indirizzi** di blocchi.

## Indirizzamento (2)



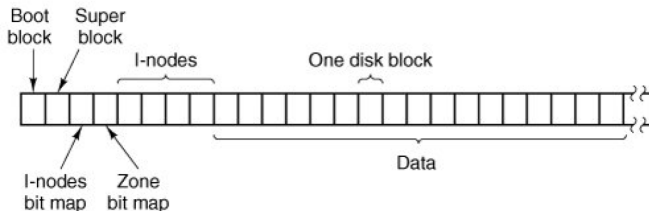
## Indirizzamento (3)

### MINIX 3

dimensione del blocco = 1 kB  
indirizzi di 16 bit (2 byte)

- 7 blocchi di dati sono accessibili direttamente  
file di dimensioni minori di  $7 \times 1 \text{ kB} = 7 \text{ kB}$  sono accessibili direttamente
- 1 blocco è accessibile con indirettezza semplice  
(Puntatore da 4 Byte)  
il numero di puntatore è la dimensione del blocco (1024 byte) diviso dim. del puntatore (4 byte) = 256 blocchi.
- 1 blocco è accessibile con indirettezza doppia  
Quindi,  $L = 7 + 256 + 256^2$  blocchi da 1 kB fa  $> 64 \text{ MB}$  (al massimo per filesystem).

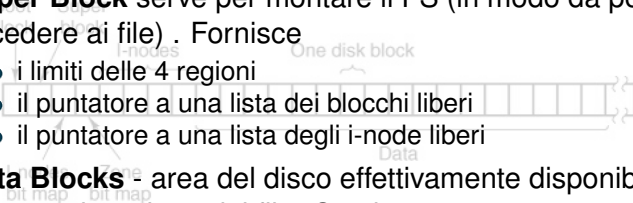
# Struttura fisica del file system (1)



- Superficie del disco partizionata in **4 regioni**
  - boot block
  - super block
  - i-list
  - data block (dim: 1024 Byte)

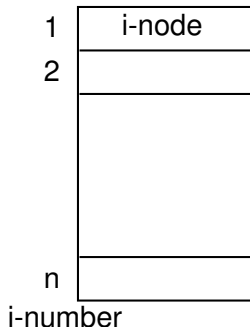
## Struttura fisica del file system (2)

- **Boot Block** contiene le procedure di inizializzazione del sistema (da eseguire al bootstrap)
- **Super Block** serve per montare il FS (in modo da poter accedere ai file) . Fornisce
  - i limiti delle 4 regioni
  - il puntatore a una lista dei blocchi liberi
  - il puntatore a una lista degli i-node liberi
- **Data Blocks** - area del disco effettivamente disponibile per la memorizzazione dei file. Contiene:
  - i blocchi allocati
  - i blocchi liberi (organizzati in una lista collegata)



## Struttura fisica del file system (3)

- **i-List** contiene la lista di tutti i descrittori (i-node) dei file, direttori e dispositivi presenti nel file system del file system (accesso con l'indice i-number)





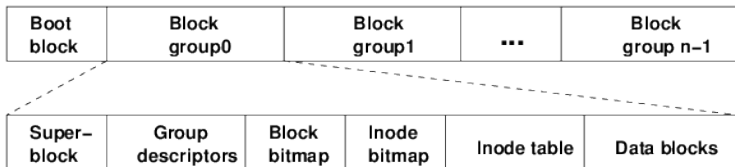
## Linux Ext2fs (1)

- Gli extended file systems sono i file system di default di Linux
- Ext2fs usa default block size di 1 kB, ma è anche in grado di supportare blocchi da 2kB, 4kB e 8kB (Linux 2.6.28).
- Nel i-node sono **12** blocchi accessibili direttamente invece di 7.
- Quindi, Ext2fs supporta 16GB / 256 GB / 2 TB / 2 TB (perché?) dati

# Linux Ext2fs (2)

## Organizzazione fisica

- Il disco è diviso in gruppi di blocchi
- Ogni gruppo di blocchi contiene i **metadati** e i dati dei file. Ha una **copia di backup** del super block (per affidabilità)
- **Bitmap**
  - Block bitmap (1 bit per blocco [libero/occupato])
  - Inode bitmap (1 bit per i-node [libero/occupato])



# Linux Ext2fs (3)

## Limitazioni

Mentre facevo gli aggiornamenti è saltata la corrente. Cosa succede?

- Esempio.
  - sto creando molti file in una directory
  - Il sistema si è arrestato in modo anomalo durante l'aggiornamento della voce di directory
  - Tutti i nuovi file sono ora "persi".
- Recupero (fsck)
  - Potrebbe non essere possibile.

# Linux Ext3fs (1)

Una versione journaling di Ext2fs

- Largamente basato su codice Ext2
- un fs Ext3 "cleanly unmounted" può essere montato come fs Ext2
- Inserisce un journal (.journal file) in Ext2

# Linux Ext3fs (2)

## Journaling

- L'idea
  - Al inizio un registro (**journal**) è scritto che descrive tutte le modifiche al filesystem, poi il filesystem effettivo è aggiornato in un secondo momento.
- Procedure
  - Inizia la transazione;
  - Memorizza i tipi di cambiamenti ai file in un **Journal**;
  - Termina la transazione (**commit**);
  - A un certo punto di controllo (**checkpoint**) sincronizza il registro con il filesystem.

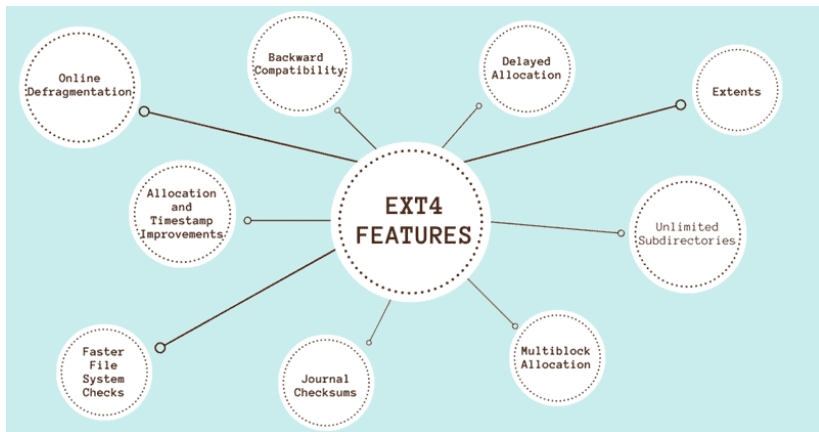
# Linux Ext3fs (3)

## Recovery

Al momento del **recovery** (da una partizione `ext3` danneggiata)

- Controlla i log dall'ultimo **checkpoint**;
- Se è stato eseguito il **commit** di un log delle transazioni, applica le modifiche al filesystem;
- Se un log delle transazioni **non** è stato eseguito, ignora semplicemente la transazione.

# Linux Ext4fs



- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 **LINUX dentro il browser**
- 3 I primi passi



# LINUX dentro il browser

- Emulare Linux via web, tramite un browser e JavaScript.
- Sviluppato di un celebre hacker francese, Fabrice Bellard
- ispirato dal suo QEMU (Quick Emulator) emulatore/virtualiser
- Architetture supportate sono **x86** (Intel) e **RISC-V** (Reduced Instructen Set Computer - open source)
- Impostazioni predefinite:

<https://bellard.org/jslinux/index.html>

# JSLinux (1)

## Alte opzioni di avvio

- **url** imposta l'URL del filesystem della macchina virtuale
- **mem** imposta la dimensione della memoria in MB
- **cpu** seleziona il tipo di cpu: x86, riscv64 o riscv32 (default = x86)
- **cols** numero di colonne del terminale (default = 80)
- **rows** numero di righe del terminale (default = 30)
- **font\_size** dimensione del carattere in pixel (predefinito = 15)
- **graphic** 0 = TTY, 1 = GUI (default = 0)
- **w** larghezza in pixel del frame buffer (se `graphic = 1`)
- **h** altezza in pixel del frame buffer (se `graphic = 1`)

## JSLinux (2)

```
Loading...

Welcome to JS/Linux (x86)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup.
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

[root@localhost ~]# uname -a
Linux localhost 4.12.0-rc6-g48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586
GNU/Linux
[root@localhost ~]#
```



# JSLinux (3)

[VfSync](#) [Home](#) [Login](#) [Sign up](#) [Delete account](#)

## Signup Form

Please enter your user name and password to create your account

User name:

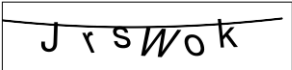
HandsOn

Password:

••••••••

Confirm password:

••••••••



Type the text:

Jrswok|

Sign up

© 2017-2018 Fabrice Bellard

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi

# La shell

- Fondamentalmente una shell è un **interprete di comandi**, che fornisce all'utente un'interfaccia verso il S.O.
- Nel tempo, le shell sono state **arricchite con caratteristiche** orientate più verso l'uso interattivo che verso la programmazione. Alcune di tali caratteristiche sono il **controllo dei job**, **l'editor da linea di comando**, **history** ed **alias**.
- I sistemi Unix offrono diverse shell. Le varie shell presentano numerosi aspetti comuni. Differiscono per la sintassi e per alcune caratteristiche e funzionalità più sostanziali.

## Tipi di shell

- **sh:** **Bourne Shell**. La shell presente sui primi sistemi Unix
- **bash:** **Bourne-Again Shell**, la shell di default per gli utenti Linux. È la shell di riferimento in questo corso.
- **ash:** **Almquist Shell**. È più leggero di `bash`, il che lo rende popolare nei sistemi Linux embedded.
- **zsh:** **Shell Z**. Creato da Paul Falstad nel 1990, `zsh` è una shell in stile Bourne +
  - il controllo ortografico,
  - la capacità di guardare gli accessi / i logout,
  - alcune funzioni di programmazione incorporate come il bytecode,
  - il supporto per la notazione scientifica nella sintassi,
  - consente l'aritmetica a virgola mobile e altre funzionalità.

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

# Sintassi dei comandi UNIX

La sintassi tipica dei comandi UNIX è la seguente:

**comando** <opzioni> <argomenti>

- ogni comando può richiedere al kernel l'esecuzione di una particolare azione;
- i comandi esistono nel file system come file binari, generalmente eseguibili da tutti gli utenti.

<opzioni> sono facoltative e influiscono sul funzionamento del comando. Generalmente consistono nel simbolo del "-" seguito da una sola lettera.

<argomenti> si possono avere più argomenti o anche nessuno in base al comando.



# Login

Quando si accede alla macchina JSLinux, apre la finestra del prompt dei comandi

## Shell

```
localhost:~# vflogin HANDSON
```

```
localhost:~# vflogin handson
```

```
localhost:~# vflogin HandsOn
```

**Password:**

Your account is now synchronized with the server every 60 seconds. You can force a synchronization with the 'vfsync' shell command.

## N.B.

I file system dei sistemi unix-like sono **case-sensitive**

# Cambiare la password

In generale, la password può essere cambiata con

## Shell

```
$ passwd
```

```
Changing password for HandsOn.
```

```
(current) UNIX password:
```

```
Enter new UNIX password:
```

```
Retype new UNIX password:
```

N.B.

non è implementato nell' emulatore.

## Ulteriori informazioni e Logout

- funzione di autocompletamento (tasto TAB);
- history (freccia SU/GIU).
- per effettuare il logout

Shell

```
$ exit
```

- 1 Lavoro di base
  - Storia di UNIX
  - Il sistema operativo UNIX
  - Il File system di UNIX
  - Indirizzamento file e percorsi
- 2 LINUX dentro il browser
- 3 I primi passi