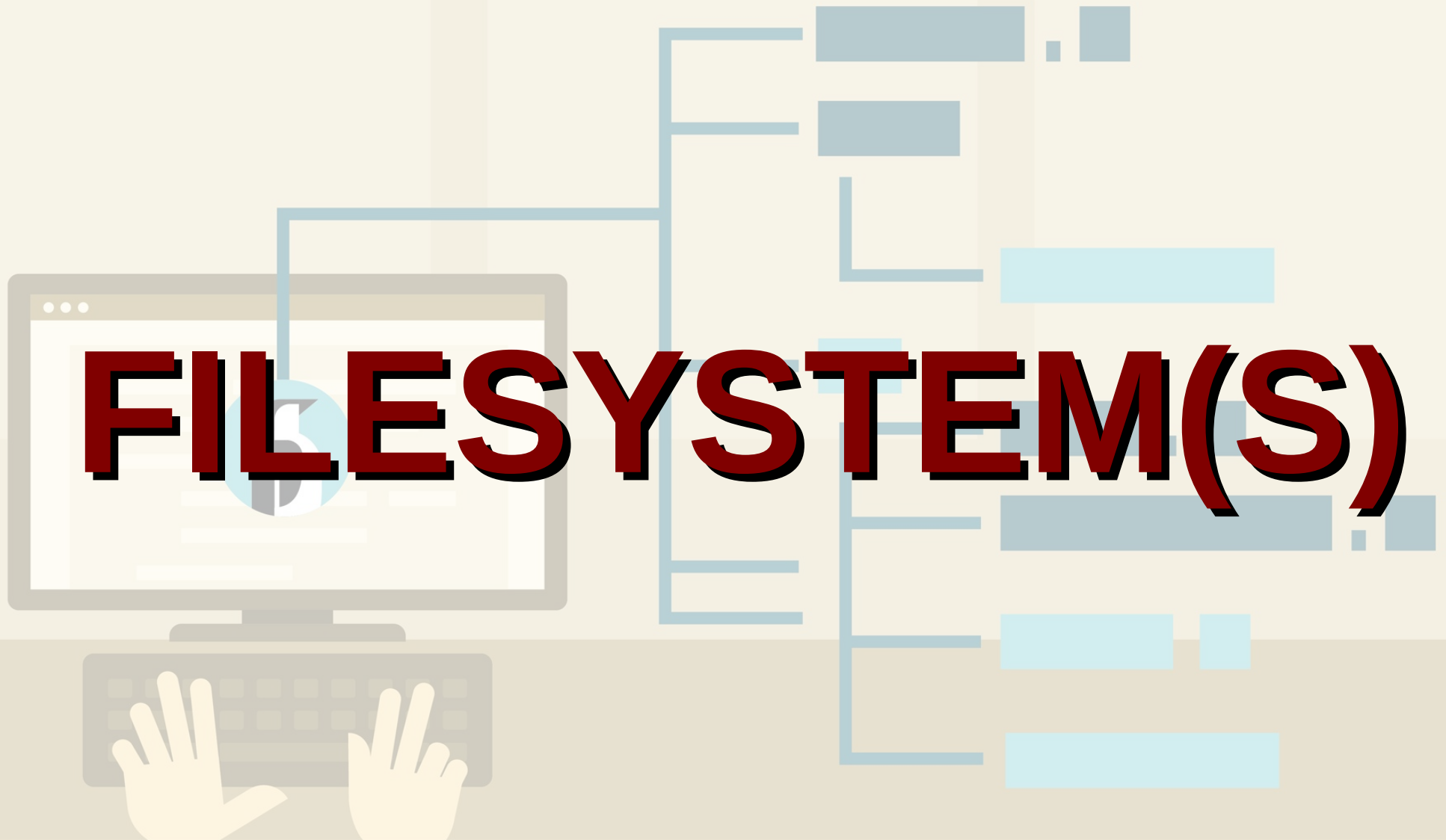


FILESYSTEM(S)



Filesystem

La parte del sistema operativo che si occupa della gestione dei file, formattando opportunamente le unità di memoria di massa, registrando e leggendo i file.

Filesystem(s)

- Si presenta tramite due viste differenti:
 - **Utente**: appare come un insieme di **file** e **directory** (o **archivi** e **cartelle**) ed è usato per memorizzare e organizzare i dati in modo che siano accessibili al sistema e ai suoi utenti.
 - **OS**: è un insieme di **strutture di controllo** e **blocchi di dati** che occupano uno spazio ben definito di una *partizione* (porzione del disco). Tale insieme permette di memorizzare e gestire i dati.

Vista Utente - File

- **Un file è un'astrazione del sistema operativo** che permette di usare in modo semplice ed efficiente i dispositivi di memoria secondaria.
- Ogni file possiede attributi (nome, tipo, dimensione, protezione, data, ora ecc).
- I file possono essere creati, letti, scritti, cancellati ecc..
- I nomi dei file possono avere, nei SO moderni una lunghezza compresa tra 1 e 255 caratteri (**percorso compreso**), in particolare:
 - Linux ha una lunghezza massima di 255 caratteri per la maggior parte dei filesystem (incluso EXT4) e un percorso massimo di 4096 caratteri.
 - Nelle edizioni di Windows < Windows 10 versione 1607, la lunghezza massima per un percorso è MAX _ PATH= 260 caratteri. Nelle versioni successive di Windows, è possibile modificare una chiave del registro di sistema o usare lo strumento Criteri di gruppo per rimuovere il limite.

Vista Utente – Tipi di File

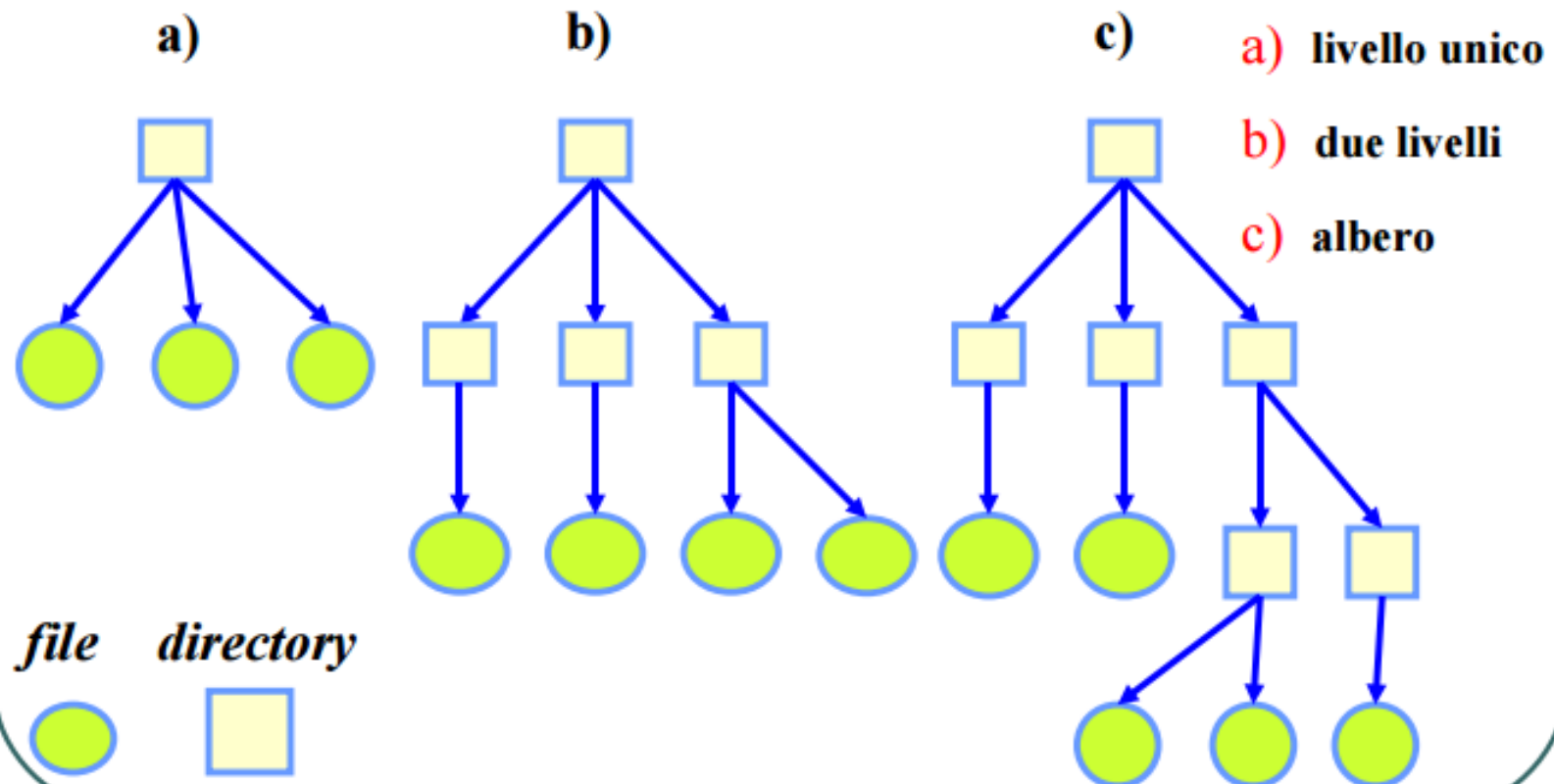
- **File regolari** (o file utente, sono ASCII o BIN).
- **Directory** (contengono file regolari e non e sono organizzate in una struttura gerarchica).
- **File speciali a blocchi** (unità di I/O a blocchi).
- **File speciali a caratteri** (unità di I/O a caratteri).
- I file si distinguono per le estensioni: exe, jpg, gif, mov, mkv ecc...
- In Windows un file speciale è quello di **Swap**...
Ora presente anche quasi tutte le distribuzioni Linux.

Vista Utente - Directory

- **È un elenco di nomi di file (o/e altre directory) a cui è associato un nome.**
- Fondamentale per dare una struttura gerarchica al filesystem.
- In tale struttura ad albero la radice è detta radice del filesystem (`/` o `c:\`).
- Sulle directory si possono effettuare le seguenti operazioni: creazione, cancellazione, link, unlink, rename ecc...

Directory – Esempio di struttura ad Albero.

Esempi di strutturazione



Struttura Filesystem in Linux/Unix

- **/etc**, contiene i file di configurazione del sistema.
- **/bin**, contiene alcuni programmi essenziali (ad esempio `ls`).
- **/sbin**, (**super user binaries**) contiene alcuni programmi essenziali che l'utente comune non può e non deve lanciare autonomamente.
- **/lib**, contiene le librerie dinamiche necessarie ai programmi delle due directory precedenti.
- **/usr**, contiene tutti i programmi che non sono necessari nelle prime fasi di avvio del sistema, nonché le relative librerie e file accessori, nelle directory **/usr/[s]bin**, **/usr/lib** e **/usr/share**, rispettivamente.
- **/usr/local**, dispone anche di sotto-directory simili a quelle di **/usr**, in cui vengono memorizzati programmi e librerie installati dall'amministratore e scelti al di fuori di quelli forniti dalla distribuzione.

Struttura storica del Filesystem in Linux/Unix

- **/home**, contiene gli spazi di memorizzazione personali (home directory) degli utenti.
- **/root**, è la home directory dell'utente root.
- **/var**, contiene i dati di tutte quelle applicazioni di uso generale del sistema, e non del singolo utente. Qui dentro vi sono, tipicamente, i dati relativi ai demoni (software) che forniscono i vari servizi.
- **/dev**, contiene file che rappresentano i dispositivi presenti nel sistema, come ad esempio dischi e terminali.
- **/tmp** e **/var/tmp**, servono a memorizzare file temporanei.
- **/opt**, simile a **/usr/local**, viene usato per quei programmi che non rispettano la convenzione di separazione dei binari dalle librerie e dai file accessori (es. `anaconda`).
- **/var/log**, vedi slide successiva...

/var/log

- *È di particolare importanza in quanto contiene i log (registri degli eventi) del sistema, gestiti direttamente dalle relative applicazioni o dal servizio syslog. (In Windows vedi **Event Viewer**)*
- Un server, ad esempio un server di posta, è un insieme di uno o più servizi, detti **demoni (daemons)**. Ogni demone ha la sua configurazione (che di solito trovate in `/etc/<nomedemone>`) e scrive tutto quello che gli succede in alcuni file, detti **file di log**, che vengono salvati nella directory `/var/log`.
- Quindi quando qualcosa non va nel vostro sistema, dovete sempre controllare in `/var/log` cosa scrivono i servizi nei rispettivi file di log per capire dove sono i problemi.
- Di solito per analizzare un file di log si utilizzano i comandi `less` o `tail`. Il comando `tail`, tramite l'opzione `-f` permette di vedere in tempo reale il contenuto di un file di log mentre viene modificato.

```
# tail -f /var/log/mail.log
```

Struttura Filesystem in Linux ora

- **/boot**, che contiene alcuni dei file necessari al gestore d'avvio (boot loader), nonché il kernel ed alcuni file di supporto.
- **/proc**, contiene dei file (o directory) che rappresentano i processi in esecuzione nel sistema: contiene una directory per ogni processo in esecuzione, e altri file che rappresentano lo stato del sistema. Ad esempio il file **/proc/cpuinfo** contiene informazioni sui processori presenti nel sistema, mentre la directory **/proc/sys** contiene directory e file con cui è possibile modificare alcuni parametri del sistema operativo a tempo d'esecuzione (equivalentemente al comando `sysctl`).
- **/sys**, si occupa di rappresentare altre configurazioni a tempo d'esecuzione di alcuni aspetti del sistema collegati all'hardware. Possiede una struttura più regolare rispetto a **/proc**.
- **/run**, si occupa di mantenere alcuni piccoli file di stato delle applicazioni di sistema che, sebbene siano necessari anche per la cooperazione tra queste, non è necessario mantenere tra i riavvii. Tradizionalmente questo compito era assolto dalla directory **/var/run**.

RIASSUMENDO...

© skill2die4@yahoo.com

/ "ROOT"

/BIN
"ESSENTIAL BINARIES"

CAT
CHGRP
CHMOD
CHOWN
CP
DATA
DD
DMESG
ECHO
FALSE
HOSTNAME
KILL
LN
LOGIN
LS
MKDIR
MKNOD
MORE
MOUNT
MV
PS
PWD
RM
RMDIF
SED
SH
STTY
SU
SYNCH
TRUE
UMOUNT
UNAME

/BOOT
"STATIC FILES OF
BOOT LOADER"

KERNEL
SYSTEMMAP
VMLINUZ
INITRD
GRUB
MODULEINFO
BOOT

/ETC
"HOST SPECIFIC
SYSTEM CONFIG"

CSH.LOGIN
EXPORTS
FSTAB
FTPUSERS
GATEWAYS
GETTYDEFS
GROUP
HOST.CONF
HOSTS
HOSTS.ALLOW
HOSTS.DENY
HOSTS.EQUIV
HOSTS.LPD
INETD.CONF
INITTAB
ISSUE
LS.SO.CONF
MOTD
MTAB
MTOOLS
NETWORKS
PASSWD
PRINTCAP
PROFILE
PROTOCOLS
RESOLV.CONF
RPC
SECURETTY
SERVICES
SHELLS
SYSLOG.CONF

/OPT
"CONFIG FILE
FOR ADD ON
APPLICATION
SOFTWARE"

/USR
"SHAREABLE AND
READ-ONLY DATA"

/LOCAL

"LOCAL
SOFTWARE"

/BIN
/GAMES
/INCLUDE
/LIB
/MAN
/SSIN
/SHARE
/SRC

/SHARE

"STATIC DATA
SHAREABLE
AMONG ALL
ARCHITECTURES"

/MAN

"MANUAL PAGES"

/MAN1 "user programs"
/MAN2 "system calls"
/MAN3 "lib functions"
/MAN4 "special file"
/MAN5 "file formats"
/MAN6 "games"
/MAN7 "misc."
/MAN8 "system admin."

/BIN

"MOST USER COMMANDS"

/INCLUDE

"STANDARD INCLUDE
FILES FOR 'C' PROG."

/LIB

"OBJ, BIN, LIB
FILES FOR PROG.
AND PACKAGES"

/SBIN

"NON ESSENTIAL
BINARIES"

/VAR

"VARIABLE DATA FILES"

/CACHE

"APPLICATION
CACHE DATA"

/LIB

"VARIABLE STATE
INFORMATION
REMAINS AFTER
REBOOT"

/YP

"DATA FOR
NIS SERVICES"

/LOCK

"LOCK FILES FOR
SHARED RESOURCES"

/OPT

"VARIABLE DATA OF
PACKAGES INSTALLED"

/RUN

"INFO OF SYSTEM
SINCE IT WAS BOOTED"

/TMP

"AVAILABLE FOR PROG."

/SPOOL

"DATA AWAITING
PROCESSING"

/LPD

/MQQUEUE

/NEWS

/RWWHO

/UUCP

/LOG

"LOG FILES
AND DIR"

LASTLOG
MESSAGES
WTMP

/SBIN

"SYSTEM BINARIES"

FASTBOOT
FASTHALT
FSCK
FSCK
GETTY
HALT
IFCONFIG
INIT
MKFS
MKSWAP
REBOOT
ROUTE
SWAPON
SWAPOFF
UPDATE

/TMP

"TEMPORARY FILES
DELETED ON BOOTUP"

/DEV

"LOCATION OF SPECIAL
OR DEVICE FILES
[CONTAINS MAKEDEV]"

/HOME

"USER HOME
DIRECTORIES"

/LIB

"LIBRARY AND
KERNEL MODULES"

/MNT

"MOUNT FILES
FOR TEMPORARY
FILESYSTEMS"

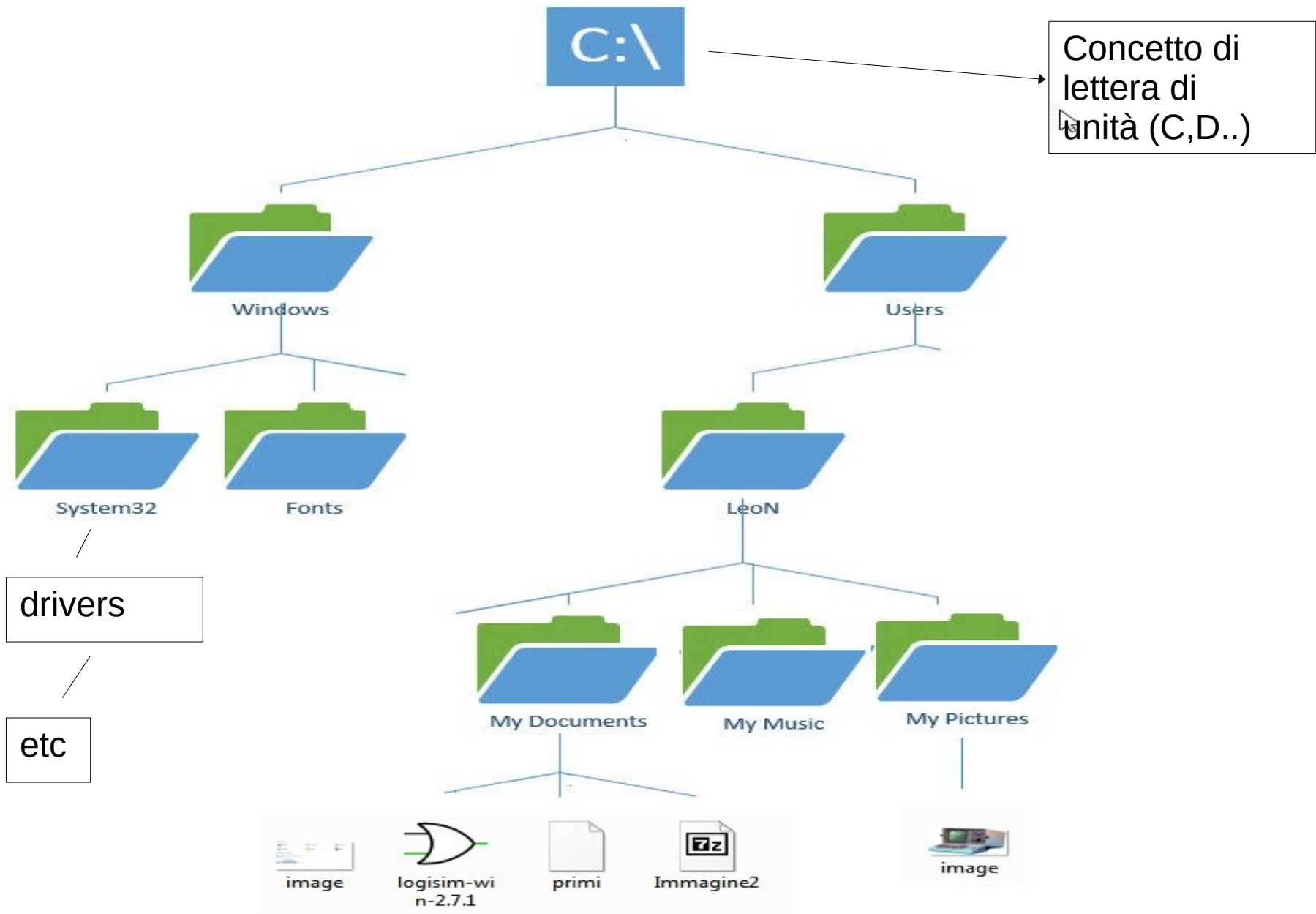
/OPT

"ADD-ON APPLICATION
SOFTWARE"

/ROOT

"HOME DIR. FOR
ROOT USER"

Struttura Filesystem Windows



SO e FileSystem

- Ogni sistema operativo si porta dietro il suo filesystem “personalizzato”:
 - **DOS e DOSLike (Windows 95/98/Me):** FAT(8), FAT16, VFAT, **FAT32**.
 - **Windows NT,2000,....2022,10,11:** **NTFS** e sue evoluzioni.
 - **MAC OS X:** HFS, **HFS+** e.... APFS.
 - **Novell Netware:** sviluppato da Novell per il suo sistema operativo Netware, uno dei primi sistemi orientati alla condivisione di risorse via rete per PC Intel.
 - **Linux/Unix-Like:** ext4, minix, ecc...

I FileSystem Unix-Like

- **Minix**: primo file system di Linux, limitato a soli 64Mb e 14 caratteri di lunghezza del nome (ecco perchè i comandi base hanno nomi corti: cp,mv,rm ecc).
- **UFS**: derivato dal Berkeley Fast File System, è uno dei filesystem più anziani nella storia di UNIX., usato da tutti i sistemi BSD (OpenBSD, FreeBSD, NetBSD).
- **ext2, ext3, ext4**: nati e sviluppati per Linux derivati da **UFS**.
- **ReiserFS**: primo filesystem journaled per Linux, dotato di utility per il ridimensionamento delle partizioni.
- **XFS**: sviluppato da SGI per Irix, journaled, con interessanti utility di backup di interi filesystem (`xfsdump/xfrestore`). .
- **ZFS**: sviluppato da Sun per il sistema operativo Solaris, journaled.
- **JFS**: sviluppato da IBM per i suoi vari sistemi (AS/400) , journaled.
- **BrtFS**: sviluppato da Intel e Nokia per il sistema operativo MeeGo.

Journaling

- **Tecnica utilizzata da molti file system moderni per preservare l'integrità dei dati da eventuali cadute di tensione o reset improvvisi.**
- Derivata dal mondo dei database, il **journaling** si basa infatti sul concetto di **transazione**: **ogni scrittura su disco è interpretata dal file system come una transazione.**
- Quando un applicativo invia dei dati al file system per memorizzarli su disco, questo prima memorizza le operazioni che intende fare su un file di log e in seguito provvede a effettuare le scritture su disco, quindi registra sul file di log le operazioni che sono state effettuate.

Journaling

- In caso di caduta di tensione durante la scrittura del disco rigido, al riavvio del sistema operativo il **file system** non dovrà far altro che analizzare il file di log per determinare quali sono le operazioni che non sono state terminate e quindi sarà in grado di correggere gli errori presenti nella struttura del file system.
- **transazione** indica una qualunque sequenza di operazioni lecite che, se eseguita in modo corretto, produce una variazione nello stato di una base di dati.
 - In caso di successo, il risultato delle operazioni deve essere **permanente** o **persistente**, mentre in caso di insuccesso si deve tornare allo **stato precedente** all'inizio della transazione.

FAT32 (FAT, FAT16, VFAT)

- Ultimo sviluppo della vecchia FAT, sviluppata inizialmente da IBM e Digital (1980) e poi anche da Microsoft.
- Si basa su una tabella, **FAT(File Allocation Table)**, da cui prende il nome.
- *Facilità di creare driver che lo ha portato ad essere l'unico filesystem che gira su tutti i sistemi (compresi i vecchi DOS).*
- Ormai lo trovate solo sulle chiavette.
- **Poco affidabile, lento e frammentario** (caro vecchio defrag...).
- Per sicurezza la **FAT** è replicata in più zone del disco.
- Non supporta file più grandi di 4GB.

NTFS

- File system sviluppato da Microsoft (e inizialmente IBM per OS/2) nel 1993, per tutti i **Windows serie NT** (NT, 2000, XP, VISTA, 7, 8, 10, 11,...).
- Permette di gestire 4 miliardi di file e si basa su una tabella chiamata **MFT** (**Master File (Allocation) Table**).
- Sicuro, stabile e flessibile: Per sicurezza la **MFT** è replicata in più zone del disco.
- **È journaled.**
- Doveva essere accantonato da WinFS...ma grazie all'avvento delle SSD perchè migliorare qualcosa che funziona?

exFAT

- **exFAT** (**Extended File Allocation Table**), o **FAT64**, è un file system, introdotto da Microsoft con Windows Embedded CE 6.0, **pensato appositamente per memorie flash e, in generale, per dispositivi di archiviazione dalle dimensioni contenute.**
- **exFAT** risulta particolarmente indicato in quei casi in cui **NTFS** risulta poco flessibile, e introduce notevoli miglioramenti rispetto alla **File Allocation Table (FAT)** precedente, tra cui:
 - il limite teorico delle dimensioni dei file passa da 4 gigabyte a 16 exabyte.
 - permette di memorizzare più di 1000 file per cartella.
 - il supporto di liste di controllo degli accessi (ACL).
 - ha prestazioni migliori, in particolare nell'allocazione di spazio libero.

Ext4

- Deriva da **ext2** (**S**econd **E**xtented Filesystem) a sua volta derivato da Berkeley Fast File System (**UFS**).
- **Filesystem di default su Linux**. Viene considerato stabile dal dicembre 2008.
- Versione migliorata di ext2 ed ext3: più veloce, più sicuro.
- **Journalled** (come tutti ormai e come lo era ext3).
- Veloce e performante anche per grandi quantità di piccoli file...
- Retrocompatibile con le versioni precedenti.
- Supporta file di dimensione fino a 16TB e partizioni grandi fino a 1 EB.

HFS+

- Una versione del vecchio **HFS** ma **Journalled**.
- *File system, vecchio e obsoleto, spesso poco performante ma perfettamente integrato nel SO che ne copre i difetti.*
- **Stabile e affidabile...** ma l'affidabilità si riduce con l'aumentare del peso del SO che supporta. Apple ha realizzato **APFS**...
- **APFS**: risulta ottimo sui tutti quei Mac dotati di SSD nativa, mentre per tutti gli altri modelli Apple forza l'uso di **HFS+**.

APFS

- è un file system sviluppato da Apple e introdotto a partire da **Mac OS Sierra** come alternativa a **HFS+**.
- A partire da **macOS 10.3** **APFS** sostituisce **HFS+** nei dispositivi Apple.
- Da **Mac OS High Sierra** è stata inclusa in Utility Disco la possibilità di convertire volumi **HFS+** in **APFS**, anche di avvio.
- Ottimizzato per le memorie flash e a stato solido, implementa un meccanismo di **copy-on-write**.
- **Copy-on-write**: è una tecnica di ottimizzazione che mira alla riduzione delle operazioni di duplicazione delle risorse del sistema (generalmente strutture dati o aree di memoria) attraverso l'eliminazione delle copie non necessarie.