



INFORMATICA

Sistema Operativo (fondamenti)

Dott. Franco Liberati
liberati@di.uniroma1.it

SISTEMA OPERATIVO

Argomenti della lezione



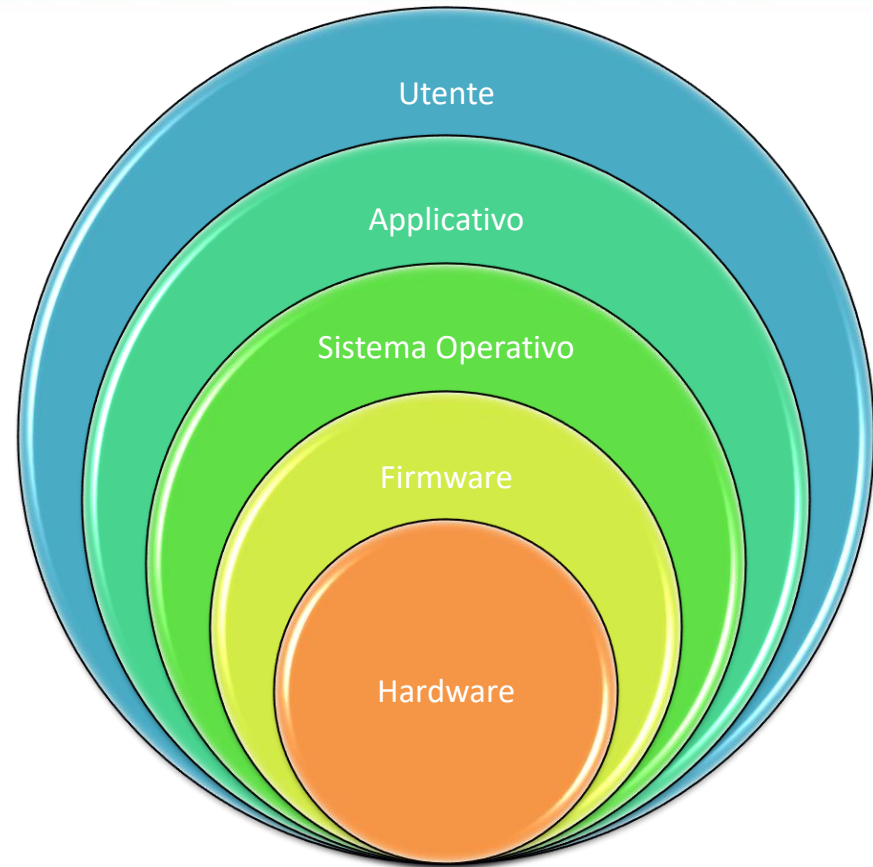


Sistema Operativo

SISTEMA OPERATIVO

Generalità

- ❑ Esistono diversi tipi di linguaggi di programmazione e le loro differenze dipendono dal loro impiego rispetto ai due estremi: l'hardware e l'utente
- ❑ I **linguaggi di basso livello** sono quelli *machine oriented*, tali cioè che ogni istruzione (una combinazione di codici binari) corrisponde a una specifica azione compiuta dal processore (es.: assembler, codice binario)
- ❑ I **linguaggi di alto livello** sono comprensibili all'uomo e dispongono di istruzioni simili al linguaggio naturale (es.: C, R, Java, Python, C++,...)
- ❑ In generale, si ha il **firmware** (linguaggio a basso livello), **sistema operativo** e **applicativo** (linguaggio alto livello)



FIRMWARE

Generalità

- ❑ Il **firmware** è un programma eseguibile che si avvia automaticamente dopo l'accensione della macchina (fase di *bootstrap*)
- ❑ Risiede su una memoria preservante di tipo EEPROM (programmabile) ed è raramente aggiornato



FIRMWARE

Operazioni

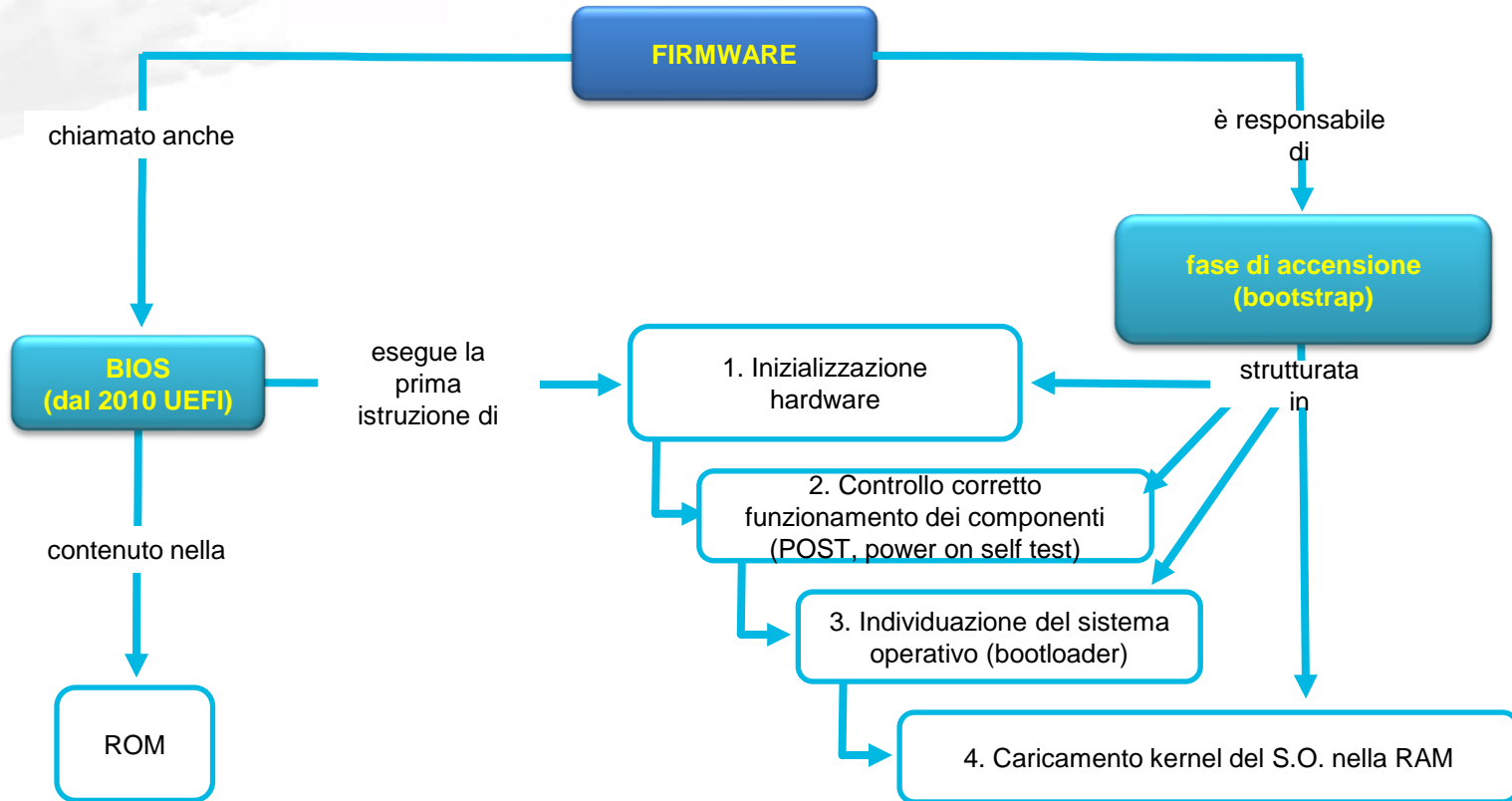
- ❑ Il firmware, come primo passo, esegue il **controllo del funzionamento dei componenti** (esistenza della tastiera e del videoterminale, quantità di RAM disponibile e funzionante, se il disco fisso magnetico è presente ed è operativo)
- ❑ In seguito il firmware esegue il **bootloader** cioè individua il Sistema Operativo (ce ne potrebbero essere più di uno installato) e carica nella Memoria Centrale la parte del Sistema Operativo che conserva le sub-routine fondamentali, cioè il **nucleo** (o kernel)
- ❑ Da questo momento, il Sistema Operativo rimane sempre in esecuzione e il kernel è sempre presente in Memoria Centrale



Dal 2010 il BIOS è stato progressivamente sostituito da UEFI

FIRMWARE

Schema



SISTEMA OPERATIVO

Definizione

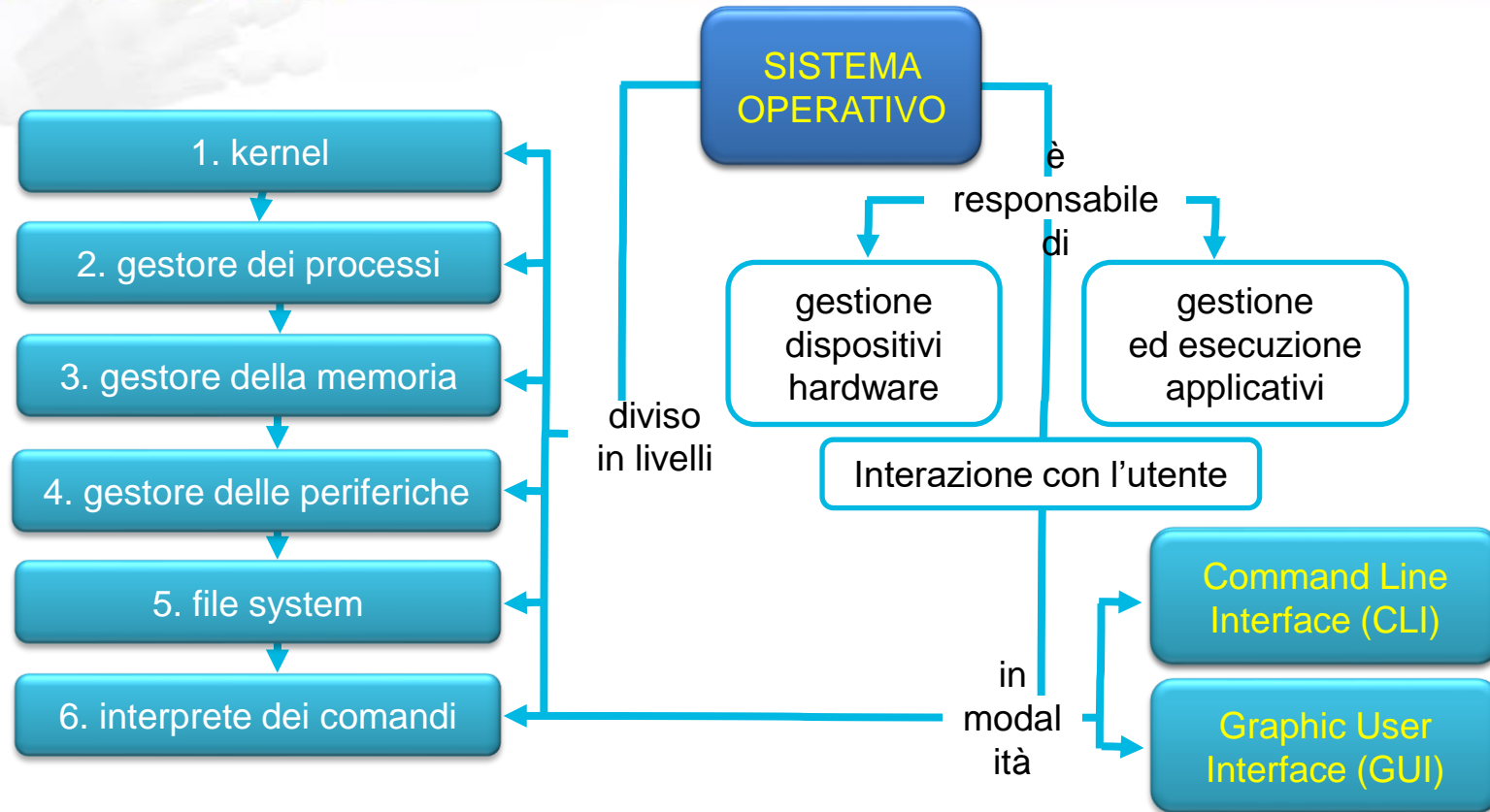
Definizione. Il Sistema Operativo è un insieme di programmi che:

- ❑ gestisce e coordina le componenti fisiche dell'elaboratore (**allocatore e gestore di risorse**)
- ❑ mette a disposizione un ambiente nel quale è possibile installare, gestire ed eseguire i programmi applicativi (**programma di controllo**)
- ❑ permette l'interazione utente-elaboratore senza conoscere tutti i dettagli dell'architettura (**fornitore di macchina estesa**)



SISTEMA OPERATIVO

Schema



SISTEMA OPERATIVO

Strutture del sistema operativo



SISTEMA OPERATIVO

Gestione dei Processi

Definizione. Un **processo** è un programma in esecuzione

- ❑ Un processo necessita di alcune **risorse** (il tempo di utilizzo di CPU, lo spazio in Memoria Centrale, dei file, interazione con dispositivi I/O)
- ❑ Le risorse possono essere attribuite quando si crea il processo o allocate durante l'esecuzione
- ❑ All'interno di un elaboratore elettronico sono eseguiti più processi (alcuni del Sistema Operativo altri dell'utente) in maniera concorrente, cioè alternando un processo ad un altro, nei mono processori e mono core (e in parallelo negli altri casi)

GESTIONE PROCESSO

CREARE E
CANCELLARE
PROCESSI UTENTI E DI
SISTEMA

SOSPENDERE E
RIPRISTINARE I
PROCESSI

SINCRONIZZAZIONE
DEI PROCESSI

COMUNICAZIONE
TRA PROCESSI

GESTIONE DEGLI
STALLI

SISTEMA OPERATIVO

Gestione della Memoria Centrale

- ❑ Il Sistema Operativo deve garantire l'**amministrazione della Memoria Centrale**
 - ❑ Ogni processo e documento digitale deve essere archiviato in Memoria Centrale

GESTIONE MEMORIA CENTRALE

TENERE TRACCIA
DELE PARTI DI
MEMORIA
UTILIZZATE (E DA
CHI)

DECIDERE QUALI
PROCESSI METTERE
IN MEMORIA
QUANDO SI HA
SPAZIO
DISPONIBILE

ASSEGNARE O
RILASCIARE SPAZIO
IN ACCORDO ALLE
NECESSITÀ

SISTEMA OPERATIVO

Gestione della Memoria Secondaria

- ❑ Il Sistema Operativo deve consentire **l'interazione e la gestione con la Memoria Secondaria** (hard disk, memoria a stato solido) nella quale risiedono in maniera permanente sia il Sistema Operativo sia i programmi sia i documenti digitali

GESTIONE MEMORIA SECONDARIA

GESTIONE SPAZIO
LIBERO

ALLOCAZIONE
DELLO SPAZIO DI
MEMORIZZAZIONE

SCHEDULING DEL
DISCO (CIOÈ
RECUPERO DEI
PROCESSI E DEI
FILE)



SISTEMA OPERATIVO

Gestione delle periferiche

- ❑ Il Sistema Operativo **astrae le caratteristiche fisiche dei diversi dispositivi di I/O** per consentire all'utente di usarle senza conoscerne a fondo le singole architetture e componenti

GESTIONE
PERIFERICHE

INDIVIDUAZIONE
ED USO

INSTALLAZIONE
E RICHIAMO DEI
DRIVER

SISTEMA OPERATIVO

Gestione dei documenti digitali

- ❑ Il Sistema Operativo ha una serie di funzioni utili per la **gestione dei documenti digitali** (*file system*) e del modo in cui essi sono organizzati sulla memoria di massa

GESTIONE FILE

CREAZIONE E
CANCELLAZIONE
FILE

CREAZIONE
CANCELLAZIONI
CARTELLE

PROTEZIONE SU
FILE E CARTELLE

SVOLGERE
BACKUP

SISTEMA OPERATIVO

Sistema di protezione

- ❑ Il Sistema Operativo si occupa anche di effettuare operazioni che consentano di assegnare ai soli processi autorizzati le risorse (es.: i file, le porzioni di memoria, l'uso della CPU)
- ❑ Il Sistema Operativo interagisce con questi e consente la coerenza e la salvaguardia dello svolgimento del processo e dell'elaboratore

ESEMPI DI PROTEZIONE

TIMER ASSICURA
CHE UN PROCESSO
NON OCCUPI PER
UN TEMPO
INDEFINITO LA CPU

INIBIRE L'USO
DIRETTO DEI
DISPOSITIVI DI I/O
AGLI UTENTI

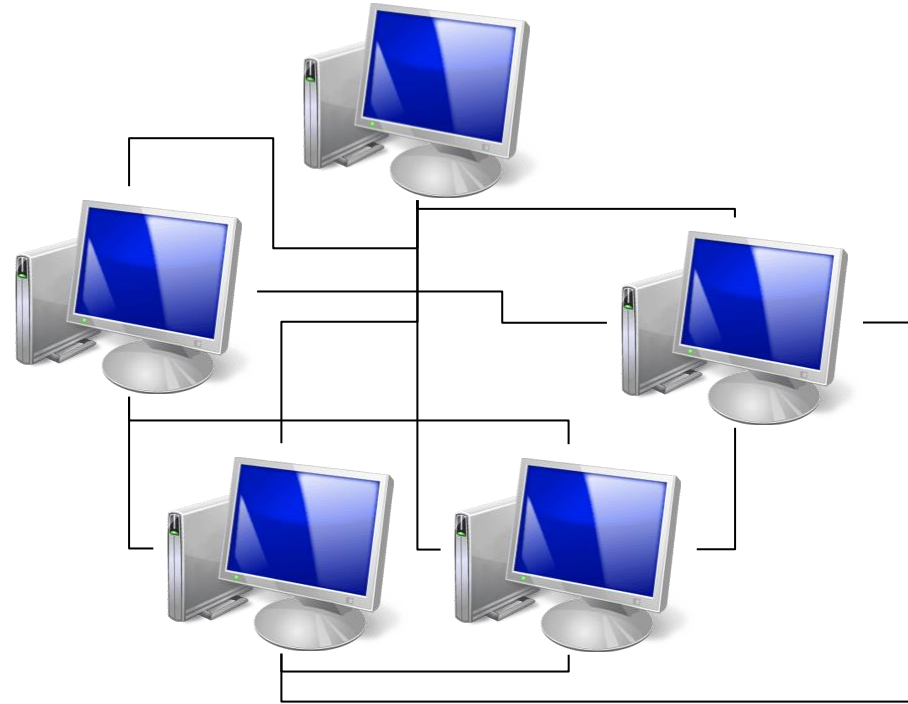
HARDWARE CHE
CONSENTE DI FAR
ESEGUIRE UN
PROCESSO NEL
PROPRIO SPAZIO DI
MEMORIA

SISTEMA OPERATIVO

Rete

Definizione. Un **Sistema distribuito** è un insieme di processori che non condividono lo stesso clock e memoria ma possono scambiarsi dati attraverso una struttura di interconnessione (linee telefoniche, fibra)

- ❑ Un **Sistema Operativo per un ambiente distribuito** deve tener conto dell'attivazione dei percorsi e delle connessioni ed evitare problemi di conflittualità e sicurezza
- ❑ Tra i vantaggi ci sono il miglioramento dell'affidabilità, l'aumento delle risorse e il miglioramento delle prestazioni di calcolo



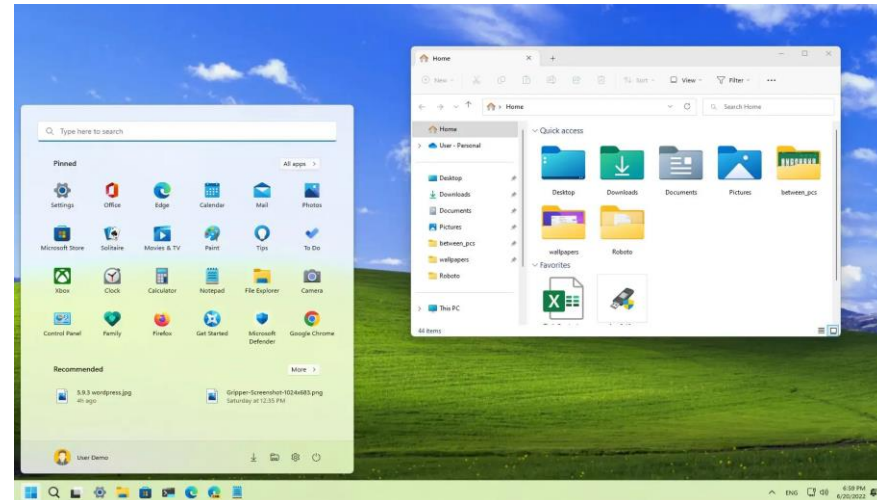
Oggi, con alcune differenze, si parla di rete di elaboratori

SISTEMA OPERATIVO

Interprete dei comandi

- ❑ Nel Kernel del Sistema Operativo è presente l'**interprete dei comandi**
- ❑ In questo modo l'utente può attivare processi generici (si avvia un eseguibile, *Running*, o lo si elimina, *Killing*), processi per interagire con i dispositivi di I/O (mount/unmount), con la memoria ausiliaria, con i file (file system), nonché processi per offrire la protezione e la comunicazione tra sistemi distribuiti
- ❑ L'interprete dei comandi può avere un aspetto grafico semplice (*shell*) o complesso (*metafora del desktop*, Windows e Macintosh)

```
vivek@nixcraft-asus:~$ echo "My current shell is $SHELL ($?)"  
My current shell is /bin/bash (bash)  
vivek@nixcraft-asus:~$  
vivek@nixcraft-asus:~$ ksh  
$ echo "My current shell is $SHELL ($?)"  
My current shell is /bin/bash (ksh)  
$ echo $SHELL  
/bin/bash  
$ tcsh  
nixcraft-asus:~>  
nixcraft-asus:~> echo $SHELL  
/bin/bash  
nixcraft-asus:~> echo $0  
tcsh  
nixcraft-asus:~> exit  
exit  
$ exit  
vivek@nixcraft-asus:~$
```





Gestione Processi

GESTIONE PROCESSI

Definizione

- ❑ Un programma presente in memoria principale, ed in corso di esecuzione, è detto **processo**
- ❑ Ogni processo ha un blocco di informazioni associate che lo descrivono il PCB (*process control block*) costituito da:
 - ❑ Stato del processo (nuovo, *new*; pronto, *ready*; in esecuzione, *running*; in attesa, *wait*; fermato, *halted*)
 - ❑ Program Counter: l'indirizzo in cui punta la prossima istruzione eseguire
 - ❑ Registri CPU
 - ❑ Informazioni di contabilità (es.: il tempo utilizzato, lo slot temporale assegnato,...)
 - ❑ Informazione I/O (es.: l'elenco dei file aperti, le risorse I/O richieste)
 - ❑ Altro...

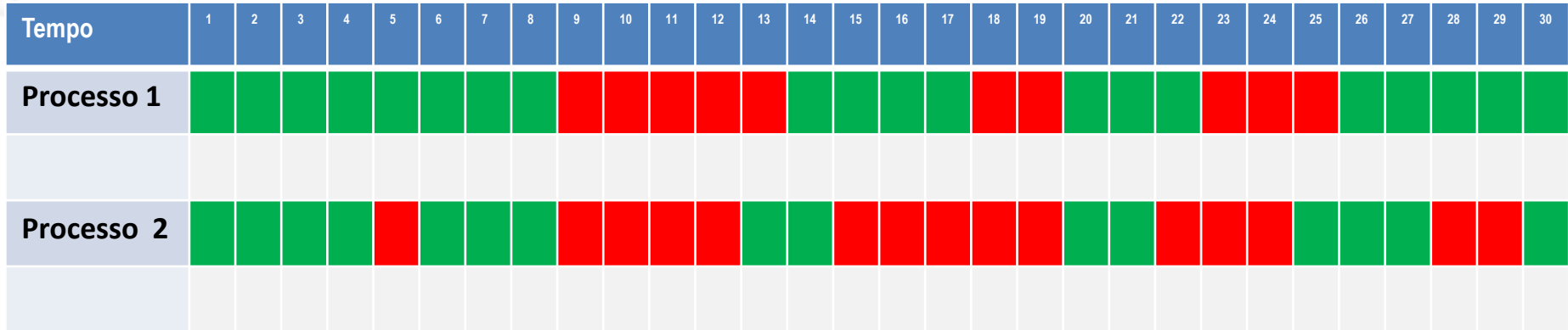


GESTIONE PROCESSI

Nozione di multiprogrammazione

Definizione. La multiprogrammazione è una tecnica basata su interruzioni che consente l'alternanza di processi in modo da massimizzare l'uso della CPU

ESEMPIO SENZA MULTIPROGRAMMAZIONE



Per portare a termine i due processi, uno dopo l'altro, si impiegherebbero $60\Delta t$ di cui $25\Delta t$ tempi di latenza

GESTIONE PROCESSI

Nozione di multiprogrammazione

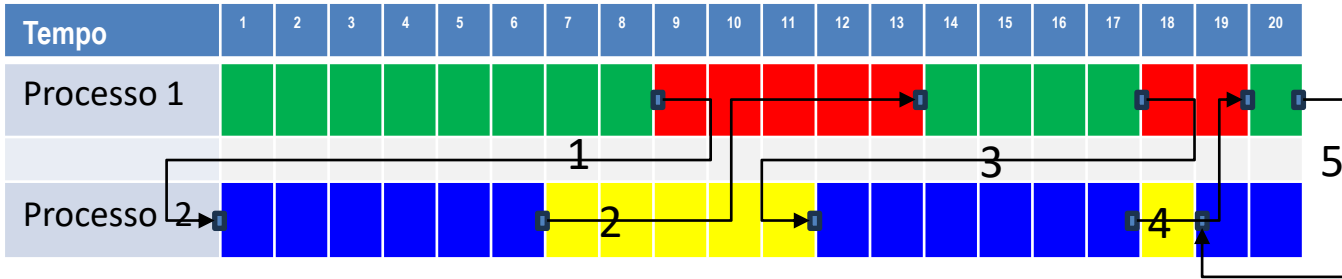
- ❑ L'idea che sta alla base della multiprogrammazione è la seguente:
 - ❑ i processi sono contemporaneamente nella RAM ed in modo concorrenziale ordinato condividono le risorse del sistema (tra cui la CPU)
 - ❑ finché un processo è in attesa di "qualcosa", un altro è in esecuzione nella CPU

Obiettivo: la CPU non deve mai essere inattiva

GESTIONE PROCESSI

Nozione di multiprogrammazione

- ❑ Alla fine di una istruzione di un programma, se ci si trova in un I/O burst, il Sistema Operativo analizza lo stato del processo e se è in WAIT lancia una interruzione che attiva un altro processo nello stato READY (diventa così *RUNNING*)

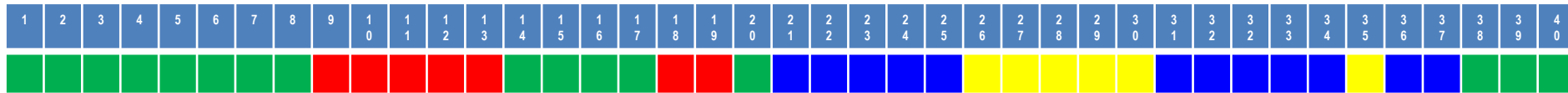


GESTIONE PROCESSI

Nozione di multiprogrammazione: esempio

- Considerati i due processi residenti in memoria il Processo 1 ha CPU/Burst verdi e IO/burst rossi mentre il Processo 2 ha CPU/Burst blu e IO/burst gialli

Tempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Processo 1	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red	Red	Red	Green	Green	Green	Green	Red	Red	Green	
Processo 2	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Yellow	Yellow	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Blue	Blue



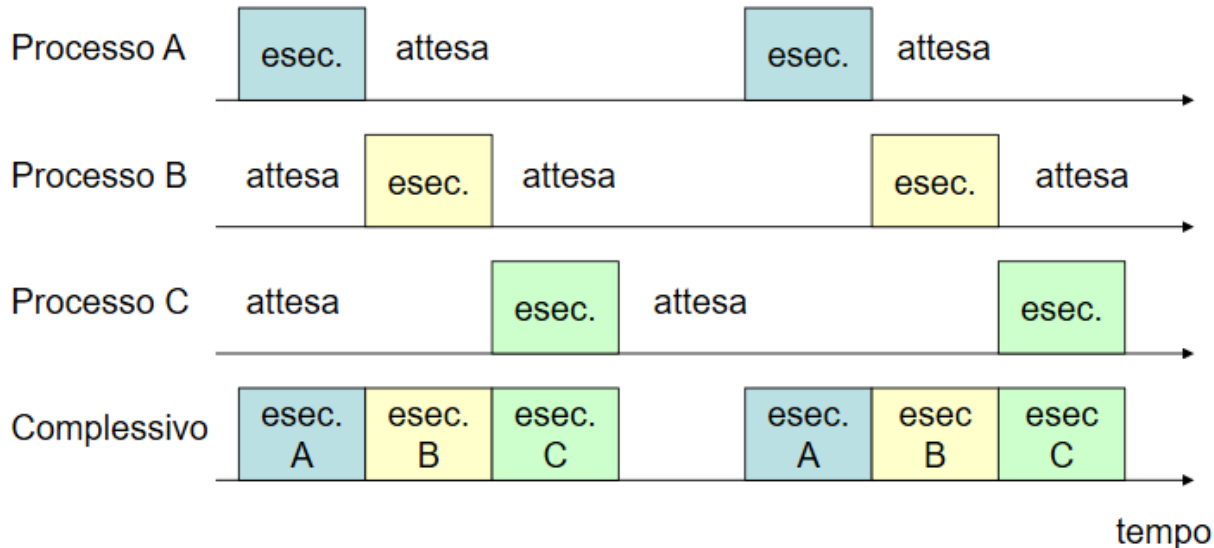
- L'applicazione della multiprogrammazione ci consente di avere un solo tempo di attesa (a fronte del tempo speso per la gestione delle interruzioni)



GESTIONE PROCESSI

Multiprogrammazione

- ❑ Con la **multiprogrammazione** si hanno più programmi eseguiti in memoria ad intervalli regolari e separati che danno un effetto di parallelismo (in realtà si tratta di pseudo-parallelismo)





GESTIONE PROCESSI

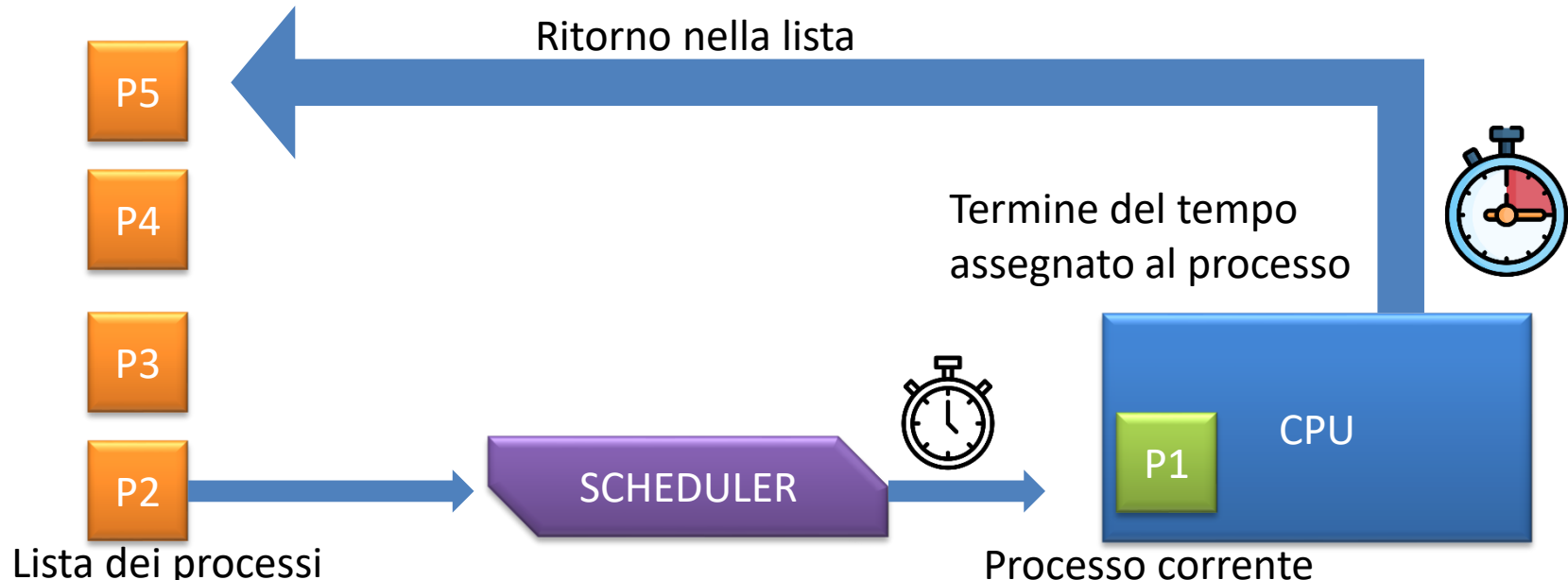
Time-sharing

- È possibile condividere la CPU tra più processi suddividendo il tempo di esecuzione del processore
- Ogni processo utilizza periodicamente un intervallo di tempo prestabilito (il **quanto** o **time slice**; di solito 10-100ms) scandito da un **Timer**
- Durante il quanto di esecuzione di un processo, tutti gli altri processi sono sospesi (wait, in attesa)
- Al termine di ogni quanto, il processo in esecuzione viene sospeso e si assegna la CPU ad un altro processo (context switch).
- Obiettivo: un solo processo alla volta è in esecuzione nella CPU, ma l'utente ha l'impressione che tutti i suoi programmi funzionino contemporaneamente

GESTIONE PROCESSI

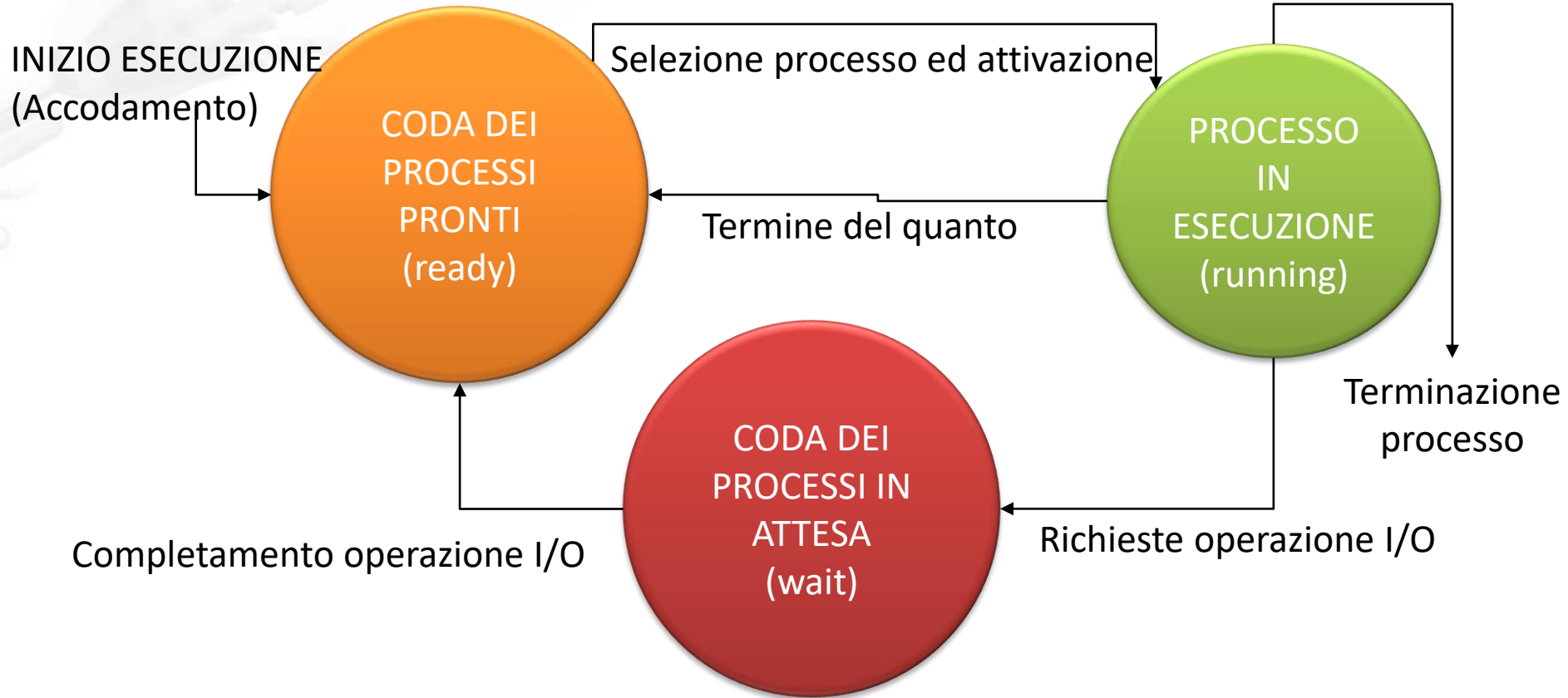
Time-sharing

- ❑ Le interruzioni generate dal Sistema Operativo a tempi prestabiliti garantiscono il **time sharing**: cioè più programmi sono eseguiti in memoria ad intervalli regolari e separati che danno un effetto di pseudo parallelismo
- ❑ Il programma che si occupa del time sharing è ancora lo **scheduler**



GESTIONE PROCESSI

Time-sharing

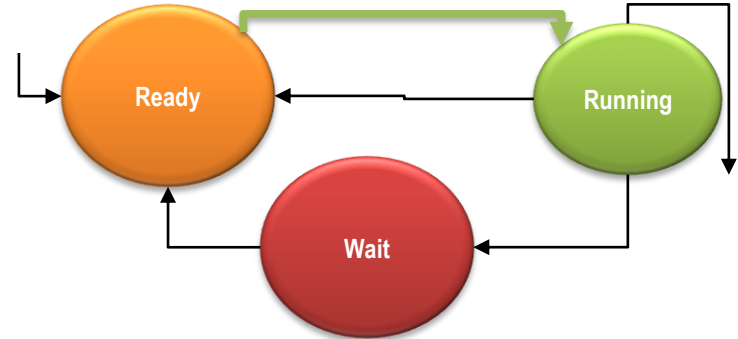


GESTIONE PROCESSI

Time-sharing

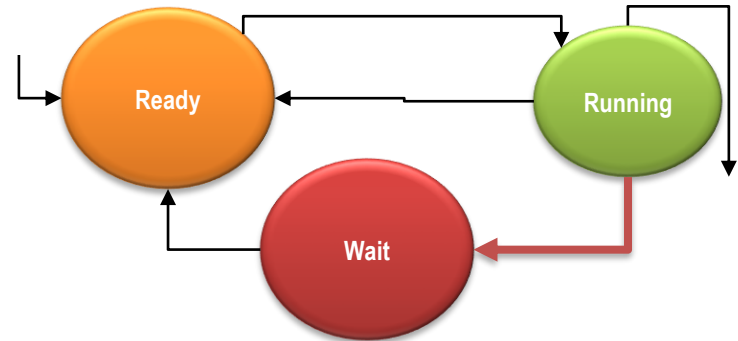
Ready → Running

Il SO stabilisce quale dei processi “pronti” debba essere mandato in “esecuzione”.
Al massimo c'è un processo in esecuzione



Running → Wait

il processo chiede delle risorse che non sono disponibili o attende un evento
Possono esserci molti processo in attesa



GESTIONE PROCESSI

Time-sharing

Wait → Ready

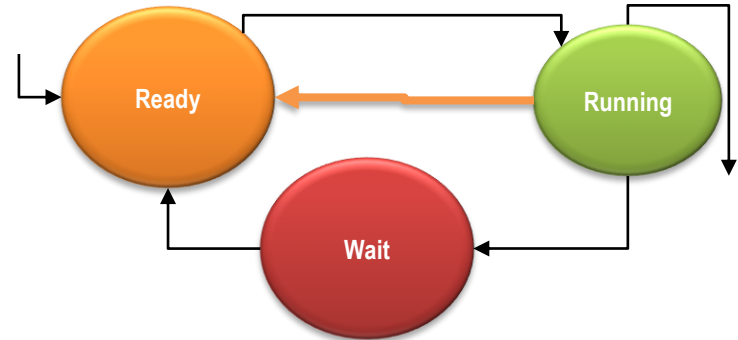
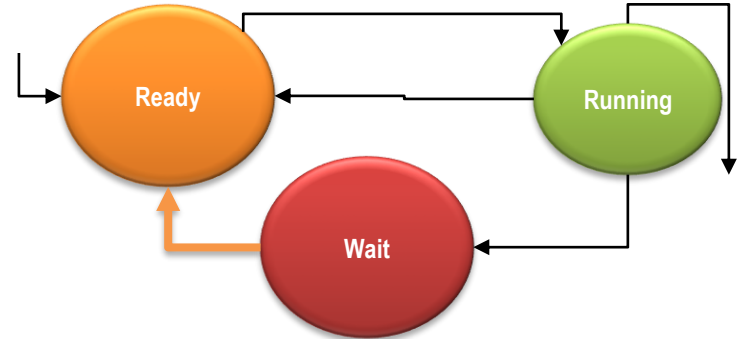
Si verifica l'evento atteso dal processo e il SO sposta quel processo nella coda dei processi pronti.

Possono esserci molti processi nella coda dei processi pronti

Running → Ready

Termina il quanto di tempo e il processo in "esecuzione" lascia spazio a un altro processo "pronto".

Contemporaneamente un altro processo passa da "pronto" a "esecuzione".



GESTIONE PROCESSI

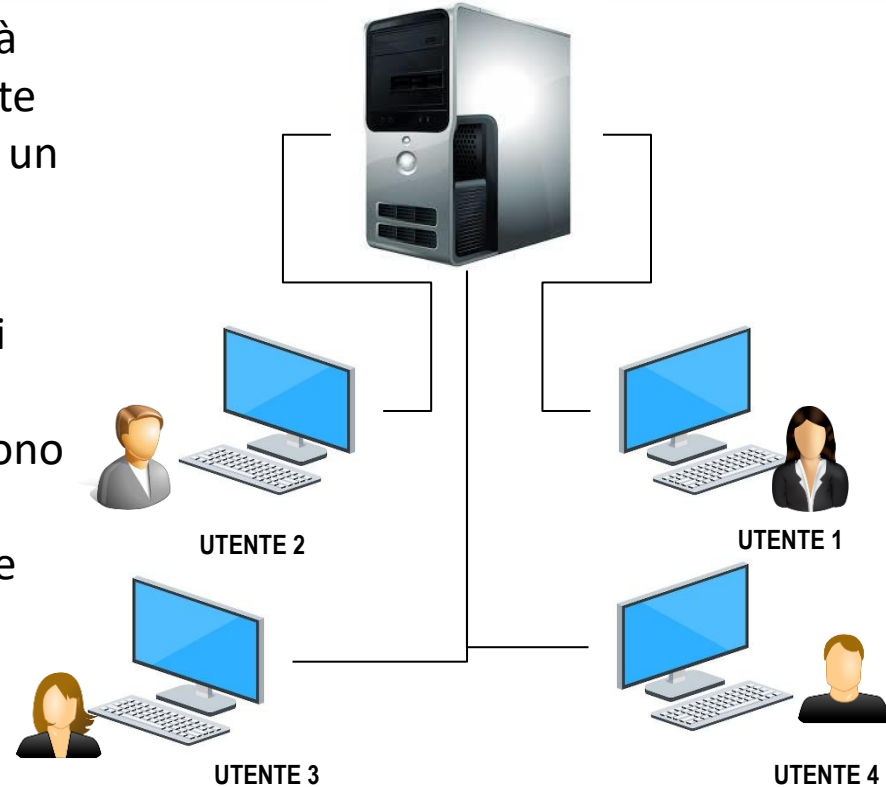
Multi-utenza

Definizione. La multi-utenza è una funzionalità offerta da alcuni Sistemi Operativi che consente a più utenti di accedere a un elaboratore (con un solo Sistema Operativo operante su di esso)

Esempio: Linux, Unix, Windows 2000, Ubuntu, Mac OS,...

- ☐ Nel Sistema Operativo multiutente, diversi utenti si collegano ad un elaboratore principale utilizzando dei terminali e possono eseguire programmi differenti (è una estensione della multi-programmazione) e accedono a risorse condivise (stampanti, file,..)

Esempi: sistemi per prenotazioni voli aerei, biblioteche





Gestore Memoria



GESTORE MEMORIA

Generalità

- ❑ Il gestore della memoria si deve occupare di:
 - ❑ tenere traccia delle **aree di memoria correntemente utilizzate** e dei rispettivi utenti
 - ❑ decidere **quale processo caricare in memoria** quando dello spazio si rende disponibile
 - ❑ **allocare o liberare (deallocare) spazio in memoria**, a seconda delle richieste
 - ❑ permettere, in modo controllato, la parziale **sovrapposizione degli spazi di memoria associati ai vari programmi** (condivisione dati, istruzioni e/o scambio messaggi)
 - ❑ **proteggere programmi e relativi dati** caricati nella memoria di lavoro mettendo in atto dei meccanismi per tutelare la privacy dello spazio di lavoro assegnato a ogni processo (nessun processo deve poter leggere o modificare quanto contenuto nello spazio di indirizzamento virtuale di un altro processo)
 - ❑ utilizza tecniche per gestire il conflitto fra dimensione della memoria fisica e spazio complessivo richiesto dai programmi che devono essere eseguiti in modo concorrente e dai relativi dati (memoria virtuale)
 - ❑ trasferimento del contenuto di un'area della memoria centrale in un'area della memoria di massa (paging)



Gestore delle periferiche



GESTORE PERIFERICHE

Generalità

- ❑ Il gestore delle periferiche si deve occupare di:
 - ❑ **garantire l'accesso ai dispositivi di I/O, mascherando dettagli di basso livello** (problemi di indirizzamento, di sincronizzazione delle periferiche, ...), tramite un insieme di comandi di alto livello per leggere e scrivere dati
 - ❑ **gestire conflitti** che possono insorgere nel caso che diverse richieste arrivino contemporaneamente a uno stesso dispositivo
 - ❑ **amministrare il controller**, i dispositivi hardware per effettuare le operazioni di trasferimento dati a livello fisico (DMA)
 - ❑ **evocare i driver**, software per la gestione delle periferiche



GESTORE PERIFERICHE

Driver

- ❑ I driver sono dei programmi usati dal sistema operativo per effettuare la comunicazione fra l'elaboratore e le varie periferiche. Per poter funzionare correttamente, ogni dispositivo deve avere il suo particolare driver registrato dal sistema operativo. Esistono perciò i driver di stampante, modem, lettore CD, scheda video, scheda audio, scheda di rete, ...
- ❑ I driver sono installati attraverso un software di installazione automatica che viene fornito assieme ai dispositivi stessi.
 1. all'attivazione il sistema operativo scandisce ed esamina tutte le periferiche collegate al sistema
 2. le periferiche si fanno riconoscere specificando di quali driver necessitano
 3. il sistema operativo installa gli opportuni driver per la loro gestione
- ❑ I driver
 - ❑ mascherano le caratteristiche specifiche dei controller
 - ❑ forniscono un insieme di primitive ad alto livello per la gestione delle operazioni di ingresso/uscita utilizzabili dai programmi applicativi e dagli utenti



Gestore Memoria Secondaria

GESTORE MEMORIA SECONDARIA

Generalità

- ❑ Il gestore della memoria secondaria:
 - ❑ amministra lo spazio libero su memorie di massa-associa a un nome lo spazio fisico sulla memoria di massa
 - ❑ fornisce meccanismi per la protezione dei dati-rende trasparente la struttura fisica del supporto di memorizzazione
 - ❑ offre un insieme di operazioni per lavorare sui file: creare o rimuovere un file, copiarlo, cambiargli nome, inserire informazioni in un file

FILE SYSTEM

Generalità

- Il **file system** è quella parte del Sistema Operativo che consente di memorizzare e gestire i documenti digitali su un supporto di massa permanente (es.: HDD, MSS,...)

Principali file system

APFS (Apple File System)	Ext (Extended file system)	Ext4 (Extended File System 4)	FAT	FAT32	ExFAT	HFS (Hierarchal File System)	HFS+ (Hierarchal File System Plus)	ISO 9660	NTFS - NT File System.	UDF	UFS (Unix File System)	UFS2 (Unix File System)	XFS (eXtended FileSystem)
Introdotta da Apple nel 2016	primo file system per GNU/Linux	Ultima versione	Usato su DOS, Microsoft Windows (tabelle a 12 e 16 bit)	versione con tabelle a 32 bit di FAT	creato da Microsoft per memorie flash	usato su Mac OS	usato su Mac OS (nuova versione)	Usato su dischi CD-ROM e DVD-ROM	Usato su sistemi basati su Windows NT	usato su supporti WORM/RW, CD-RW e DVD	usato su vecchi sistemi BSD	usato su nuovi sistemi BSD	indicato per HDD e SSD di nuova generazione



FILE SYSTEM

Generalità

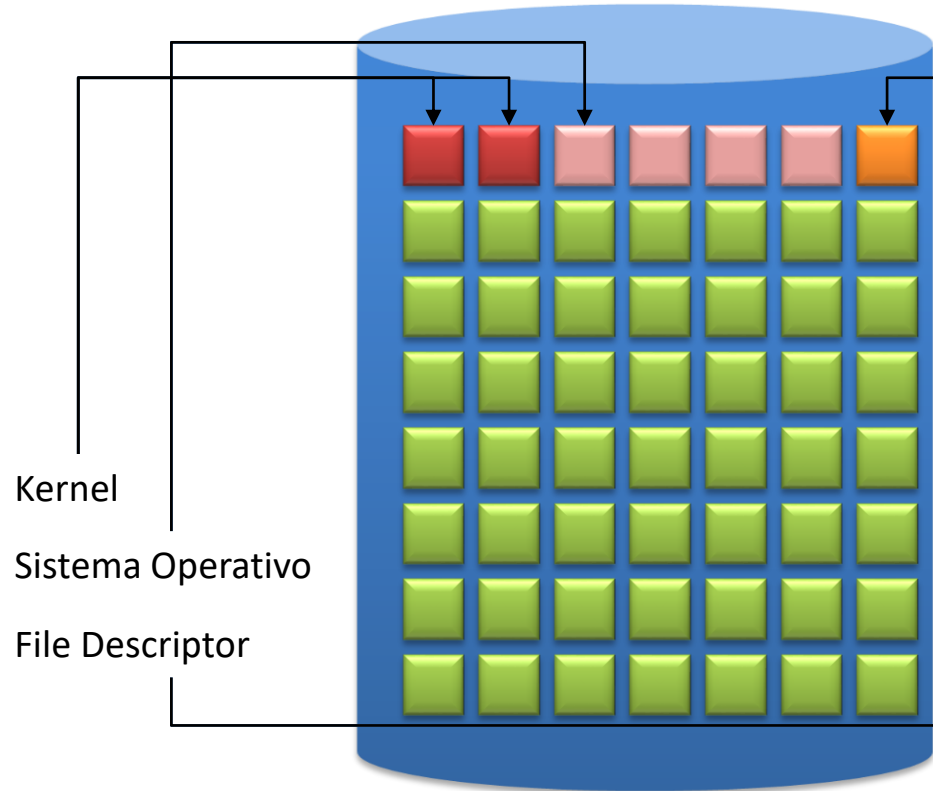
- ❑ Il file system fornisce una visione astratta dei documenti digitali su memoria di massa permanente e permette all'utente di:
 - ❑ **Identificare** ogni file mediante il suo nome
 - ❑ **Operare** sui file mediante opportune operazioni (apertura, lettura, scrittura, chiusura, esecuzione)
 - ❑ **Effettuare** l'accesso alle informazioni grazie ad operazioni ad alto livello (es.: fopen/fread/fwrite) che non richiedono la conoscenza del tipo di memorizzazione (si accede allo stesso modo ad un file memorizzato su HDD oppure su un CD)

- ❑ Sfruttando il file system, l'utente ha la possibilità di:
 - ❑ **Strutturare** file, organizzandoli secondo le loro caratteristiche
 - ❑ **Proteggere** ciascun documento digitale, se è in un sistema multi-utente; ossia impedire a utenti non autorizzati di leggerli, copiarli o cancellarli

FILE SYSTEM

Generalità

- ❑ Il file system individua e specifica la posizione di:
 - ❖ File/documento digitale
 - ❖ Directory/cartella (una struttura in cui risiedono i file)
- ❑ La posizione di file e cartelle è sita in una **tabella** (detta anche **file descriptor**) residente in memoria di massa permanente e letta e gestita dal file system nel momento in cui si attiva il kernel del Sistema Operativo



FILE SYSTEM

File

- Un **file** è una collezione di record caratterizzati da un nome ed una estensione nella forma **nome.estensione**
(es.: luna.jpg divinacomedia.txt calcolo_somma.exe)

L'estensione, di solito, indica il tipo del file (es: doc è un file di testo formattato; xls è un foglio di lavoro a celle; bmp è un file di immagine,...)

Indirizzo dall'inizio del file	Dati esadecimali	Valore ASCII
00000000	01 00 FF FF 00 00 00 00
00000010	04 00 00 00 00 00 EB 007.....I
00000020	6E 00 66 00 6F 00 72 00	n.f.o.r.m.a.z.i.
00000030	6F 00 6E 00 69 00 20 00	o.n.i.s.u.S.
00000040	63 00 72 00 69 00 62 00	c.r.i.b.b.l.e.
00000050	08 00 00 00 00 01 4D 00M.S.Sh.
00000060	65 00 6C 00 6C 00 20 00	e.l.l.D.l.g.
00000070	00 00 00 00 00 00 00 00P
00000080	14 00 14 00 FF FF FF FF
00000090	00 00 00 00 00 00 00 00P
000000a0	28 00 0A 00 77 00 08 00	(.....
000000b0	53 00 63 00 72 00 69 00	S.c.r.i.b.b.l.e.
000000c0	20 00 56 00 65 00 72 00	V.e.r.s.i.o.n.
000000d0	65 00 20 00 31 00 2E 00	e.l.l.0
000000e0	00 00 00 00 00 00 00 00P
000000f0	77 00 08 00 FF FF FF FFC.o.
00000100	70 00 79 00 72 00 69 00	p.y.r.i.g.h.t.
00000110	28 00 43 00 29 00 20 00	(C).2.0.0.1
00000120	00 00 0C 00 00 00 00 00P
00000130	B2 00 07 00 32 00 0E 002
00000140	4F 00 4E 00 00 00 00 00	O.K.

FILE SYSTEM

File: attributi

- ❑ Un file ha **attributi** che possono variare in relazione al Sistema Operativo utilizzato
- ❑ Gli attributi risiedono nella struttura del file system
- ❑ Tra gli attributi più comuni ci sono:
 - ❑ Nome (*unica informazione in formato "leggibile"*)
 - ❑ Tipo
 - ❑ Posizione (puntatore allo spazio fisico sul dispositivo)
 - ❑ Dimensione
 - ❑ Protezione (controllo su chi può leggere, scrivere, eseguire)
 - ❑ Tempo, data e identificazione dell'utente

Nome

Tipo

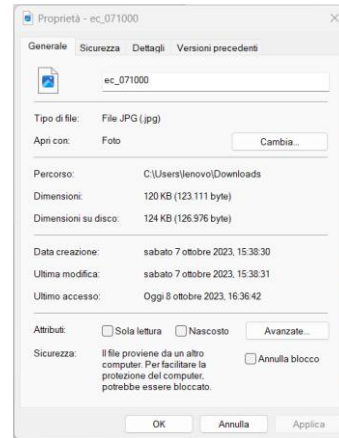
Locazione
(path)

Ora e data
ultima
modifica

Dimensione

Protezione

Ora e data
creazione



FILE SYSTEM

File: tipologie

- ❑ Il **tipo** di file specifica il documento
 - ❑ In prima battuta se è testuale o binario e poi se ha testo formattato e viceversa se è una immagine, un video, un suono o un contenuto audiovisivo
- ❑ Il tipo di file indica anche la struttura interna del file
 - ❑ Nessuna (*sequenza di parole, bytes, ...*)
 - ❑ Struttura a “record” (*database*)
 - ❑ Struttura complessa (*documenti formattati*)

Formato	Nome	Tipo di parola
Immagine compressa	JPEG	Binario
Audio compresso	MP3	Binario
Video	MP4	Binario
Testo	TXT	Testuale
Testo con marcatori	HTML	Testuale
Immagine	BMP	Binario
Audio non compresso	WAV	Binario
Immagine non compressa	TIFF	Binario
Codice programmazione C	C	Testuale
Codice eseguibile	EXE	Binario
Librerie informatiche	LIB	Binario
Informazione priva di metadati	RAW	Binario

FILE SYSTEM

Operazioni sui file

- Un insieme minimale di **operazioni**, garantito da tutti i file system è il seguente:
 - creazione** di un file (*create*)
 - cancellazione** di un file (*delete*)
 - copia o spostamento** di un file (*copy*)
 - visualizzazione del contenuto** di un file (*view*)
 - stampa** di un file (*print*)
 - modifica del contenuto** di un file (*write*)
 - ridenominazione** di un file (*rename*)
 - visualizzazione delle proprietà** di un file (*view attr*)

Creazione

Cancellazione

Lettura

Scrittura

Scrittura in
fondo

Lettura e
scrittura

Rinomina

FILE SYSTEM

Cartella/Directory

- ❑ Una **cartella** (directory) è un contenitore logico di un insieme di file
- ❑ L'uso di cartelle migliora
 - ❑ L'operazione di **ricerca** (*searching*)
 - ❑ la **nomenclatura** (*naming*): i nomi dei file sono locali alle directory; in altre parole: si possono avere due file con lo stesso nome in cartelle diverse
 - ❑ il **raggruppamento logico** (*grouping*): classificazione logica dei file per criterio (tipologia, protezione,...)

Nome

Dimensione

Locazione
(path)

Protezione

Ora e data
creazione

Ora e data
ultima
modifica

FILE SYSTEM

Operazioni sulla cartella

- Su una directory il file system agisce con una serie di **operazioni**:
 - Aggiungere un file
 - Cancellare un file
 - Visualizzare il contenuto della directory
 - Rinominare un file
 - Ricercare un file/cartella secondo un criterio
 - Es.: cercare tutti i file/directory il cui nome inizia per «Doc»
 - Attraversare il file system

Aggiunta file

Cancellazione
file

Visualizzazione
lista file

Rinomina

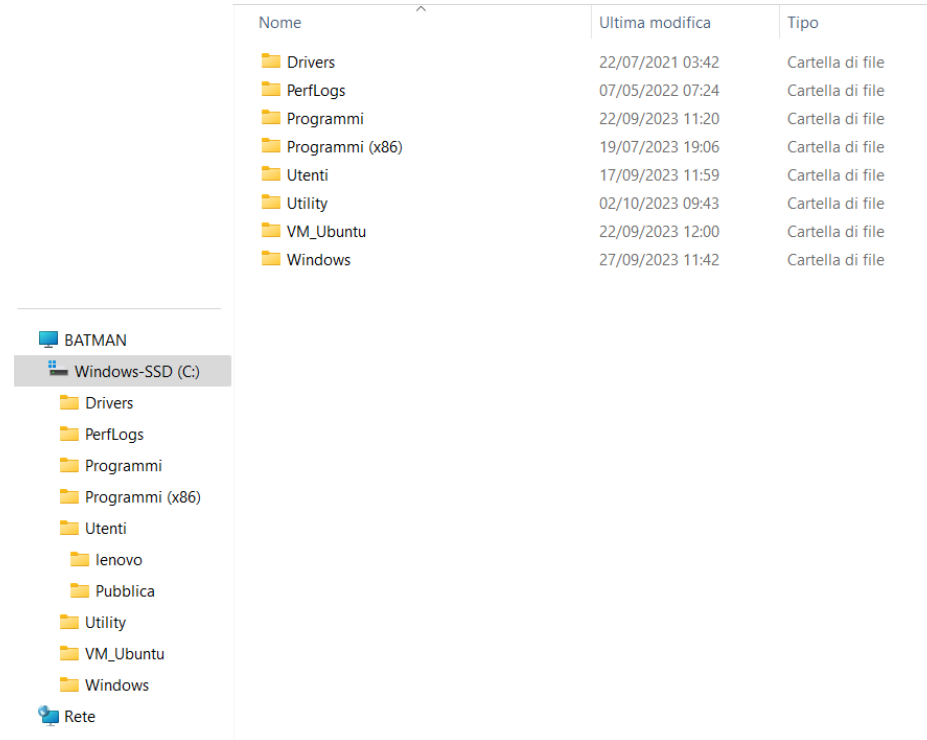
Ricerca di un
file/cartella

Analisi del file
system

FILE SYSTEM

Organizzazione gerarchica

- ❑ Una **organizzazione gerarchica** prevede una cartella principale (**root**) nella quale risiedono uno (o più) file e una (o più) cartella
- ❑ Nelle cartelle può esistere una (o più) cartella, la **sotto cartella (sub – directory)**, contenenti file e altre cartelle
 - ❑ Nel Sistema Operativo DOS e Windows la **root** corrisponde al nome della memoria di massa (C:, A:,...) nella quale risiedono i file
 - ❑ In Unix/Linux la gestione è trasparente all'utente che conosce solo il nome del file, e non si interessa dell'unità dove esso è memorizzato



The image shows a Windows File Explorer window with a list of folders and their details. The left pane shows the navigation tree with 'Windows-SSD (C:)' selected. The right pane shows a table of folders with columns for 'Nome', 'Ultima modifica', and 'Tipo'.

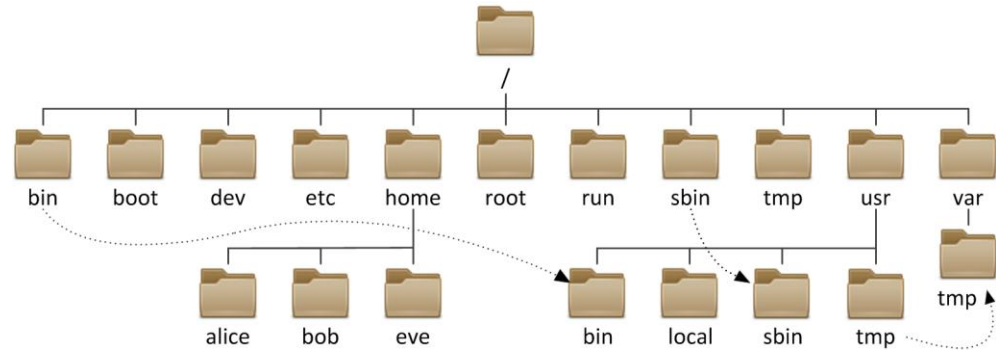
Nome	Ultima modifica	Tipo
Drivers	22/07/2021 03:42	Cartella di file
PerfLogs	07/05/2022 07:24	Cartella di file
Programmi	22/09/2023 11:20	Cartella di file
Programmi (x86)	19/07/2023 19:06	Cartella di file
Utenti	17/09/2023 11:59	Cartella di file
Utility	02/10/2023 09:43	Cartella di file
VM_Ubuntu	22/09/2023 12:00	Cartella di file
Windows	27/09/2023 11:42	Cartella di file

FILE SYSTEM

Organizzazione gerarchica: ad albero

❑ L'organizzazione gerarchica che ha una root iniziale è detta anche **gerarchia ad albero**:

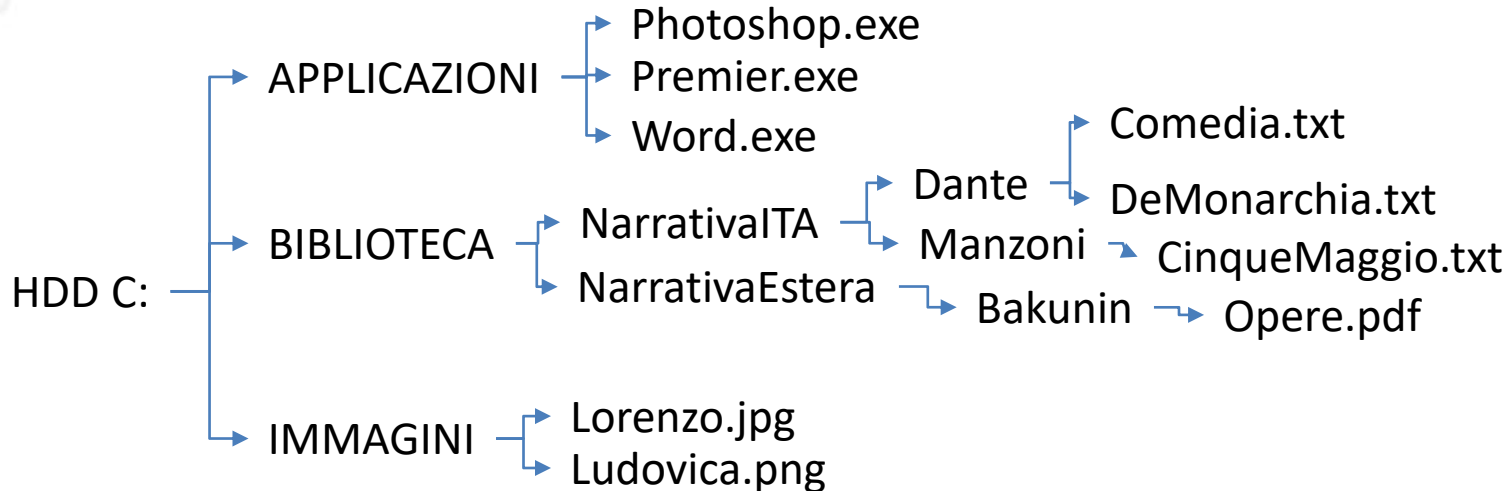
- ❑ Ricerca efficiente
- ❑ Possibilità di raggruppamento
- ❑ Concetto di directory corrente (*working directory*)
 - ❖ `cd /spell/`
 - ❖ `cd /spell/mail/prog`



FILE SYSTEM

Organizzazione gerarchica: individuazione di un file

- ❑ Nel caso di un'organizzazione ad albero per individuare un file o una directory in modo univoco si deve specificare l'intera sequenza di directory che lo lo contengono a partire dalla radice (**pathname**); cioè si esplicita il **percorso assoluto**
- ❑ Il carattere “\” (slash) viene usato come separatore di directory; nei sistemi Unix si usa il carattere “/”

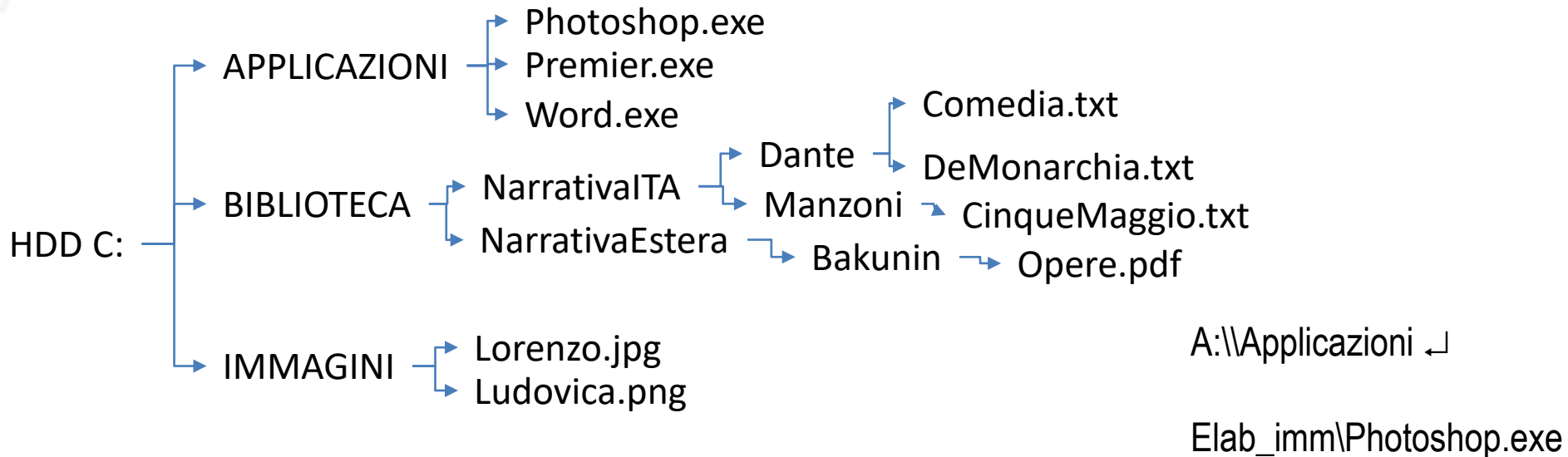


C:\\Biblioteca\\NarrativaITA\\Dante\\Comedia.txt

FILE SYSTEM

Organizzazione gerarchica: individuazione di un file

- ❑ Un altro modo di individuare un file (o una directory) è basato sul concetto di posizione relativa (relative pathname); cioè si esplicita il **percorso relativo**
- ❑ Per individuare un file o una directory in modo univoco si deve specificare l'intera sequenza di directory che lo contengono a partire dalla posizione corrente



FILE SYSTEM

Protezione

- Le informazioni contenute in un elaboratore devono essere **protette** da accessi impropri (protezione). Questo aspetto è molto sensibile nei sistemi condivisi (es.: multiutente o distribuiti)
- Il possessore di un file deve:
 - poter controllare
 - Cosa è possibile fare su un file
 - Da parte di chi
 - svolgere le operazioni di controllo
 - Lettura
 - Scrittura
 - Esecuzione
 - Append (aggiunta in coda)
 - Cancellazione

PROTEZIONE IN LINUX

In Linux un file può essere assegnato a

- 1) Proprietario
- 2) Gruppo - Utenti (appartenenti dello stesso gruppo del proprietario)
- 3) Altri

Le operazioni/permessi sono:

Letture: r (read)

Scrittura: w (write)

Esecuzione: x (execute)

`chmod 754 angeliedemoni.txt`

I campi 754 in binario sono

111 **101** **100**

rwX rwX rwX

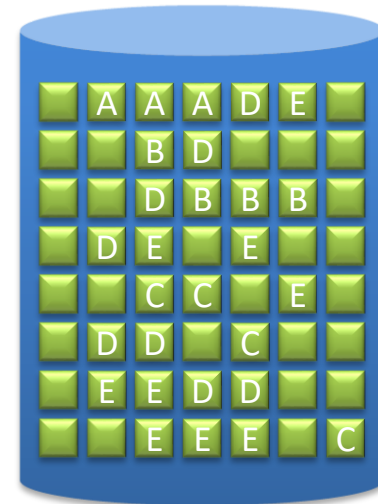
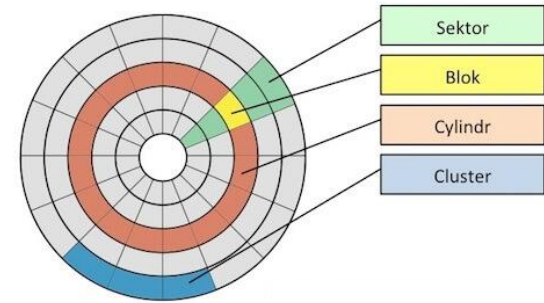
Il comando CHMOD imposta la protezione del file angeliedemoni.txt in questo modo:

*il **proprietario** ha il pieno controllo del file, gli appartenenti al **gruppo** non possono modificare il file, gli **altri** possono solo leggere*

FILE SYSTEM

Memorizzazione file

- ❑ Il file è suddiviso in parti ciascuna occupante un **blocco** del disco magnetico (HDD) o nella memoria a stato solido (SSD)
- ❑ Un blocco è una quantità di spazio del supporto in grado di memorizzare informazione digitale (di solito un blocco è di 4KB o 8KB)
- ❑ Per questo motivo c'è bisogno di una struttura che ne descrive la posizione e che nel contempo minimizza i tempi di accesso e massimizza l'utilizzo dello spazio:
 - ❑ **Allocazione contigua**
 - ❑ **Allocazione a lista concatenata (linked)**
 - ❑ **Allocazione indicizzata**





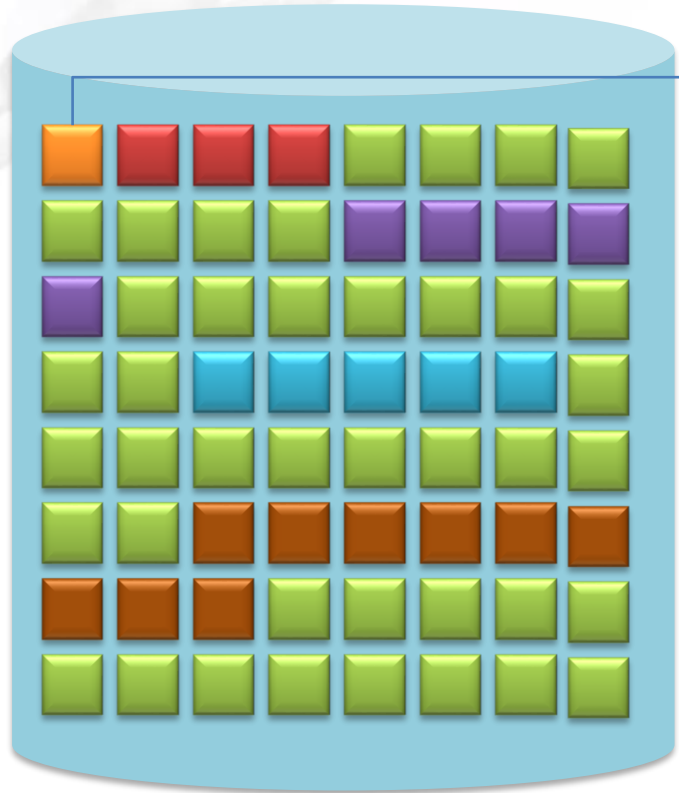
FILE SYSTEM

Allocazione contigua

- Nell'**Allocazione Contigua** ogni file occupa un insieme di blocchi contigui su memoria di massa
- Nel **file descriptor** si riporta l'indirizzo del blocco di partenza e la lunghezza (numero di blocchi)
- Vantaggi
 - Accesso semplice
 - Accesso veloce (i blocchi sono contigui)
- Svantaggi
 - perdita di tempo nel trovare lo spazio dove allocare il file
 - Spreco di spazio (frammentazione)

FILE SYSTEM

Allocazione contigua



FILE DESCRIPTOR

File	Inizio	Lunghezza
Lorenzo.jpg	1	3
Pinocchio.txt	12	5
Ludovica.jpg	26	5
RichardBenson.mp3	42	9



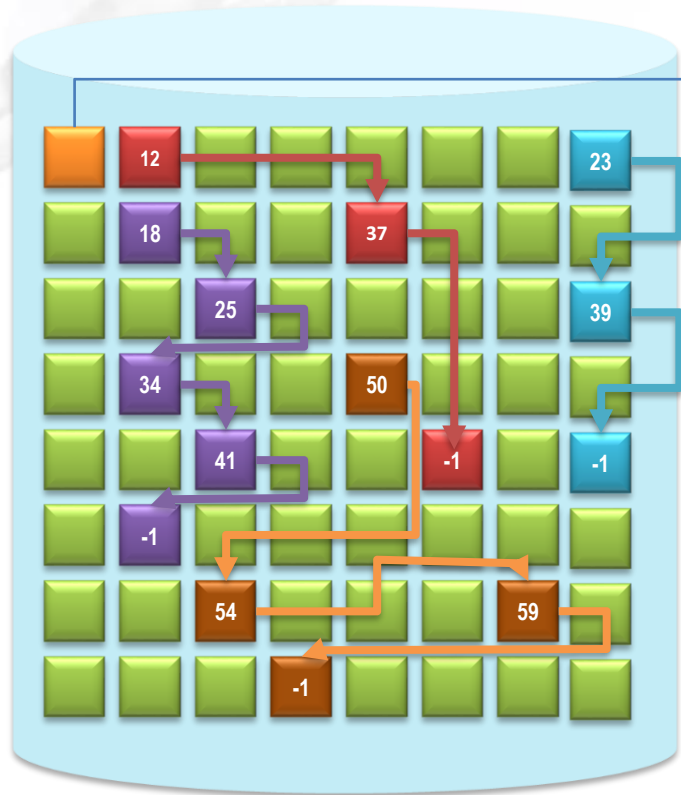
FILE SYSTEM

Allocazione a lista

- Nella **Allocazione a Lista Concatenata** ogni file è una lista di blocchi
- I blocchi possono essere sparsi ovunque
 - Il **file descriptor** contiene puntatori al primo e all'ultimo blocco
 - Ogni blocco contiene un puntatore al blocco successivo (l'ultimo ha un puntatore NULL)
- Vantaggi
 - Creazione nuovo file semplice
 - Possibilità di incrementare il file semplice
 - Nessuno spreco (eccetto lo spazio per il puntatore)
- Svantaggi
 - No accesso casuale (bisogna scorrere tutti i blocchi a partire dal primo)
 - Scarsa efficienza (blocchi in posizioni distanti)
 - Scarsa affidabilità (se si perde un puntatore si perde l'intero file!)

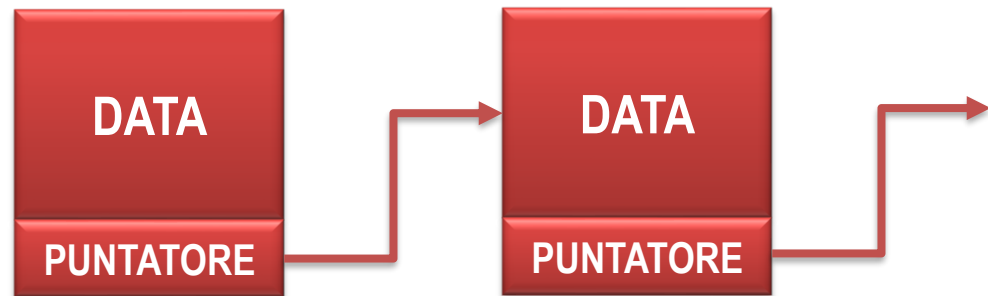
FILE SYSTEM

Allocazione a lista



FILE DESCRIPTOR

File	Inizio	fine
Lorenzo.jpg	1	37
Pinocchio.txt	9	41
Ludovica2.jpg	7	39
RichardBenson2.mp3	28	59





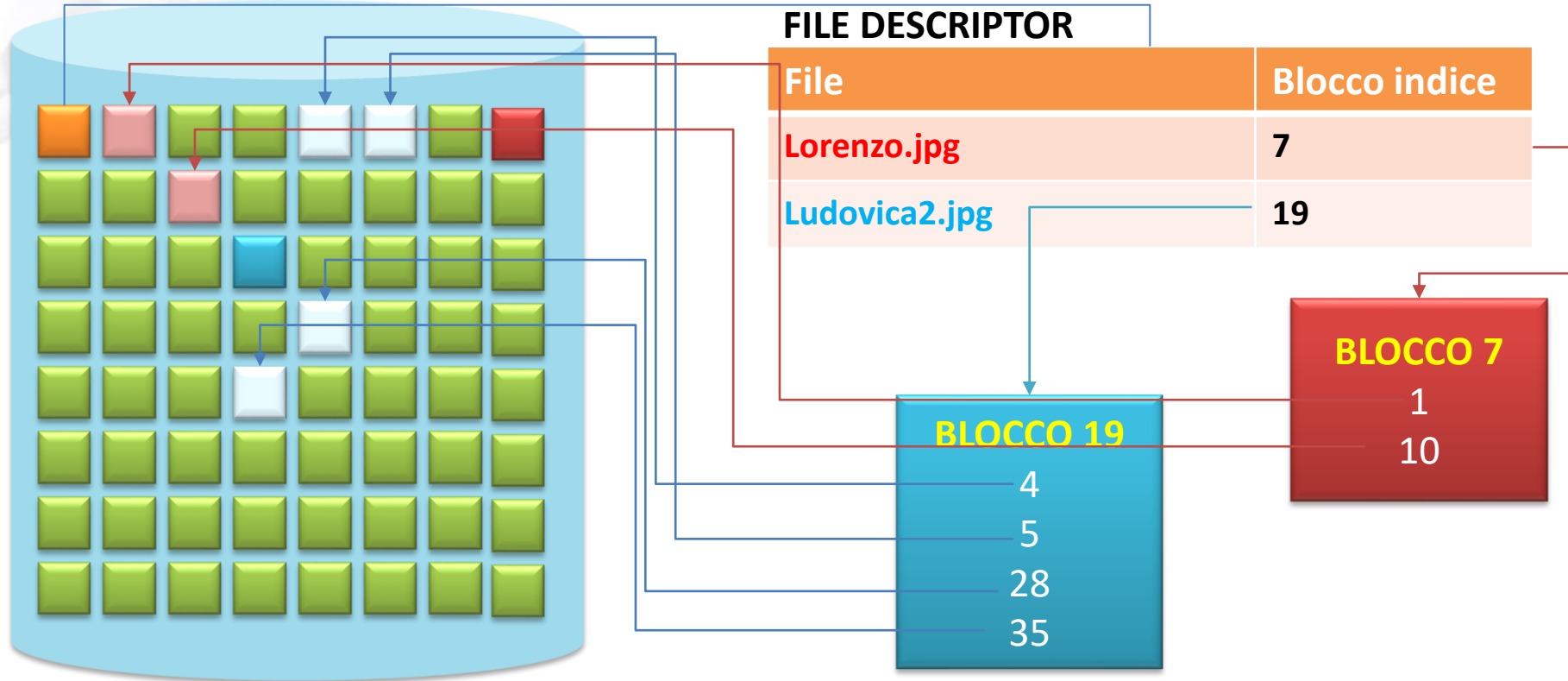
FILE SYSTEM

Allocazione indicizzata

- Nella **Allocazione indicizzata** ogni file ha un blocco indice (*index block*) contenente la tabella degli indirizzi (*index table*) dei blocchi fisici
- Il **file descriptor** contiene l'indirizzo del blocco indice
- Vantaggi
 - Accesso casuale efficiente
 - Accesso dinamico senza frammentazione esterna
- Svantaggi
 - La dimensione del blocco limita la dimensione del file
 - soluzione: per file di dimensione senza limiti si usa uno schema a più livelli (es.: Indici multilivello o Schema concatenato)

FILE SYSTEM

Allocazione indicizzata





Fine