



Corso di Informatica

I Concetti base dell'informatica

Concetti di base della IT

SOFTWARE

- Software:
 - Insieme di istruzioni, comandi che permettono al computer di compiere generiche funzioni
- Programma:
 - Una sequenza di istruzioni, scritte secondo un determinato linguaggio, con le quali si fa eseguire ad un computer il compito prefissato
- Software: "Insieme di Programmi"

Concetti di base della IT

SOFTWARE

- Il Software può essere classificato in SW:
 - di Base
 - Senza il quale l'elaboratore non può operare
 - Software fondamentale, che racchiude tutte le funzioni basilari per un computer
 - Es.: il *SISTEMA OPERATIVO*
 - Applicativo
 - Software non indispensabile
 - Espande le funzionalità fornite dal software di base
 - Es.: *Word, Excel, PowerPoint, Access, Outlook, ...*

Concetti di base della IT

SOFTWARE

Il software di base, conosciuto anche come *software di sistema*, è il software in grado di gestire tutte le risorse hardware di un qualunque computer. In parole povere, è la parte del software più vicina all'hardware della macchina.

i sistemi operativi, tipo Windows, Linux o macOS;

i compilatori e gli interpreti;

le librerie.

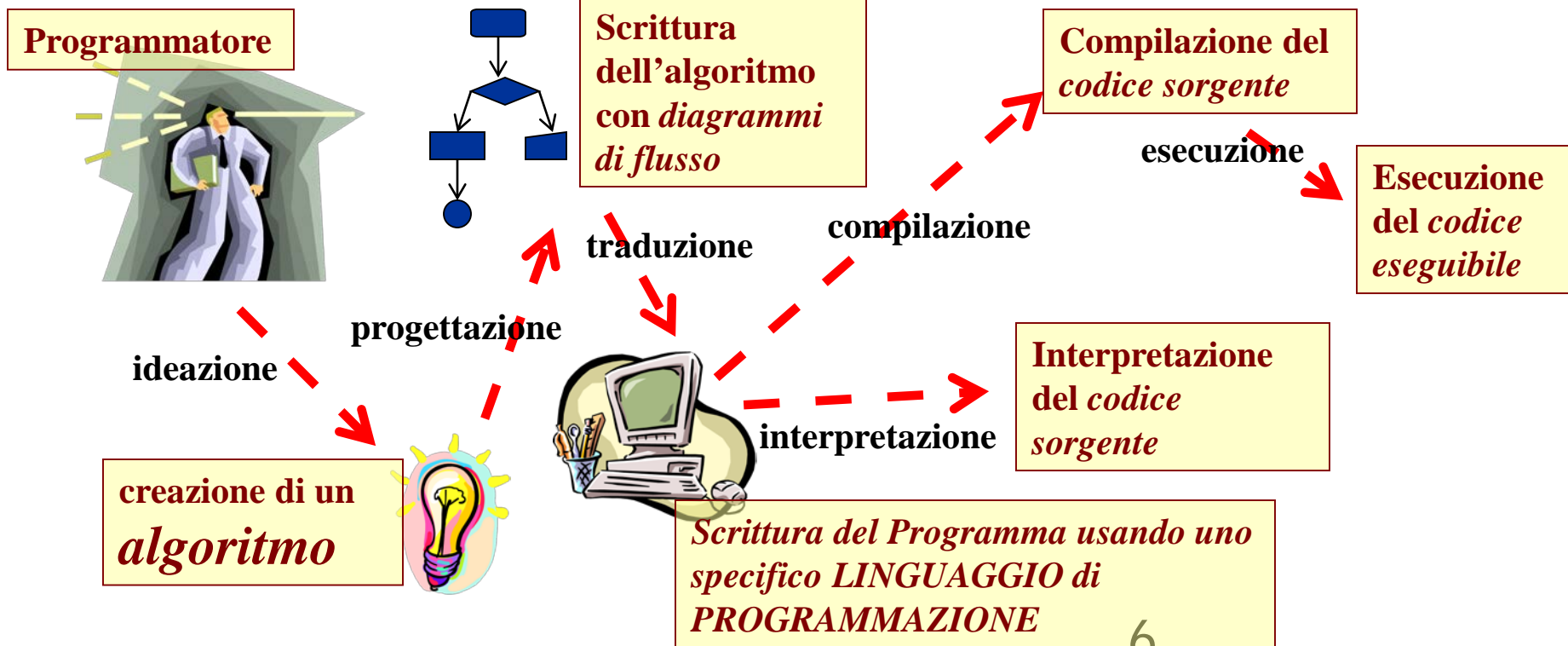
Concetti di base della IT

SOFTWARE

Il software applicativo, conosciuto anche con il termine di *applicazione*, è l'insieme dei programmi che aiutano l'utente a risolvere una vasta tipologia di problemi. In altre parole, si tratta di tutte quelle applicazioni che non fanno parte del sistema operativo ma che sono comunque necessarie all'utente per rendere possibile una o più determinate funzionalità.

Concetti di base della IT

Creazione di SOFTWARE



SISTEMA OPERATIVO 1

- Software di sistema che gestisce e controlla le risorse e le attività del computer
- Insieme dei programmi di base, ai quali spetta il compito di controllare e rendere possibile l'utilizzo di un computer

SISTEMA OPERATIVO 2

Principali funzioni:

- controllare il flusso dei dati tra l'unità centrale e le periferiche
- gestire e controllare le risorse e le attività del computer
 - regola il collegamento e lo scambio di dati e informazioni tra le diverse componenti hw e regola le azioni di queste componenti a fronte di un comando impartito dall'utente.

SISTEMA OPERATIVO 2

Principali funzioni:

- gestire i files: l'insieme dei dati o informazioni prodotte dal sistema di elaborazione, e i programmi sono contenuti in oggetti che prendono il nome di file. I file vengono memorizzati sulle memorie di massa e richiamati a seconda delle necessità dall'utente.
- Fornire un'interfaccia utente (mezzo attraverso cui l'utente interagisce con il computer)
 - a linea di comando (testo)
 - grafica (GUI)

SO: Interfaccia utente

L'**interfaccia utente**, anche conosciuta come UI (dall'inglese User Interface), è un'**interfaccia** uomo-macchina, ovvero ciò che si frappone tra una macchina e un utente, consentendone l'interazione reciproca.

- A linea di comando
- Grafica

INTERFACCIA UTENTE TESTUALE (a linea di comando)

Le operazioni vengono compiute scrivendo dei comandi su una riga dello schermo (**riga di comando**), individuata da una sequenza di caratteri detta **prompt**.

```
Welcome to FreeDOS

CuteMouse v1.9.1 alpha 1 [FreeDOS]
Installed at PS/2 port
C:\>ver

FreeCom version 0.82 pl 3 XMS_Swap [Dec 10 2003 06:49:21]

C:\>dir
Volume in drive C is FREEDOS_C95
Volume Serial Number is 0E4F-19EB
Directory of C:\

FDOS                <DIR>    08-26-04    6:23p
AUTOEXEC  BAT             435    08-26-04    6:24p
BOOTSECT  BIN             512    08-26-04    6:23p
COMMAND   COM          93,963    08-26-04    6:24p
CONFIG    SYS             801    08-26-04    6:24p
FDOSBOOT  BIN             512    08-26-04    6:24p
KERNEL    SYS          45,815    04-17-04    9:19p
        6 file(s)                142,038 bytes
        1 dir(s)           1,064,517,632 bytes free

C:\>_
```

GUI

(Graphical User Interface)₁

- Interfaccia utente grafica di un sistema operativo
- Definisce gli elementi grafici usati per interagire con il computer e usa simboli di facile comprensione

Comprende:

- finestre della scrivania virtuale e delle applicazioni
- icone, aspetto del puntatore
- menu a tendina

GUI

(Graphical User Interface)₂

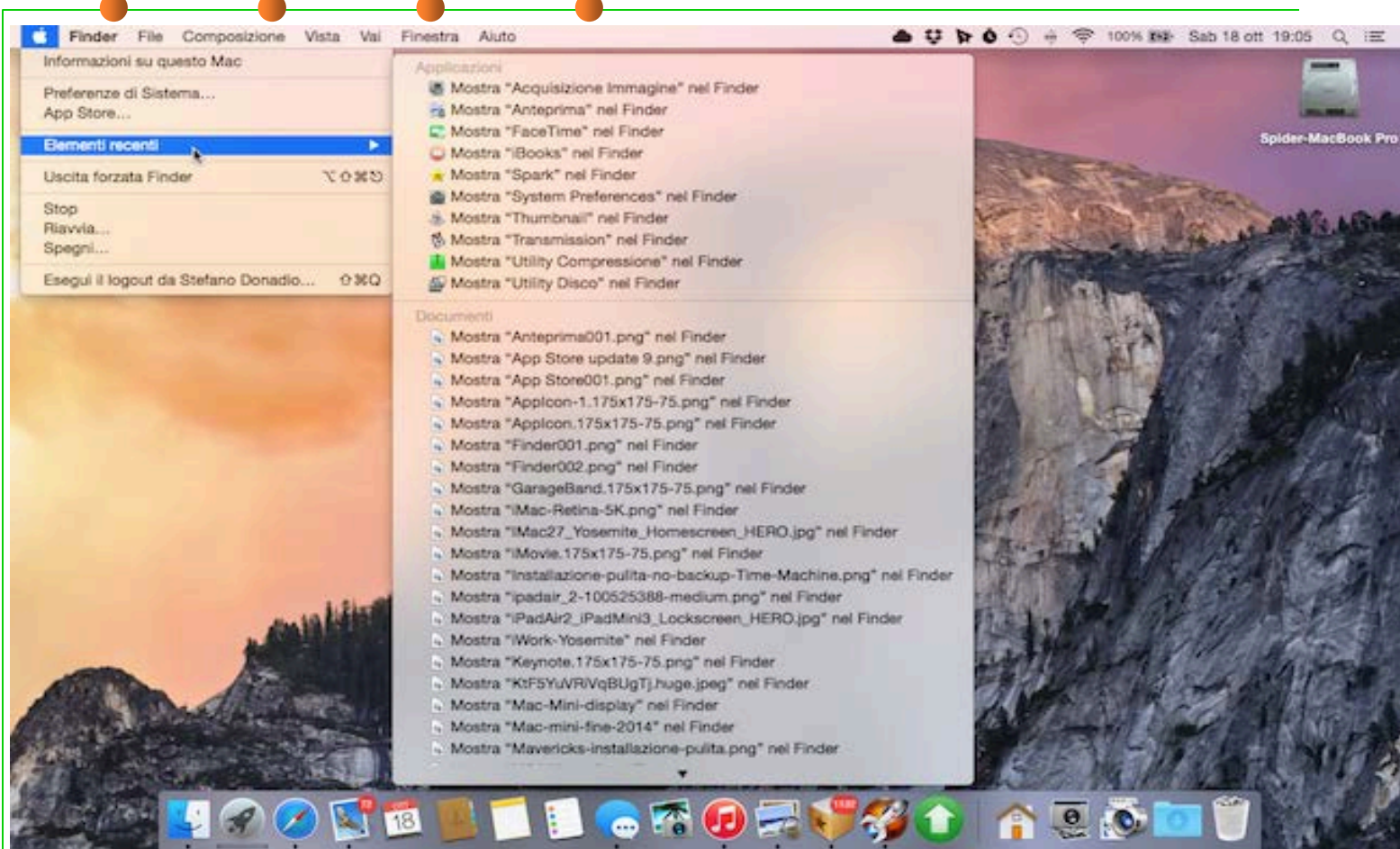
Vantaggi:

- Consente l'interazione uomo-macchina in modo visuale, semplice utilizzando rappresentazioni grafiche
- I programmi e i file vengono gestiti in modo più semplice e razionale
- Le istruzioni complesse sono generalmente più semplici da eseguire, perché basate su elementi grafici

GUI (Graphical User Interface)₃



GUI (Graphical User Interface)₄



SO: GESTIONE DEI PROCESSI

Programma e processo

Programma e processo: concetti ben distinti

Il **programma** è un insieme statico di istruzioni

Il **processo** è un'attività dinamica le cui proprietà mutano con il passare del tempo

Il processo è un programma in esecuzione

(*parte statica* [il programma] + *parte dinamica*

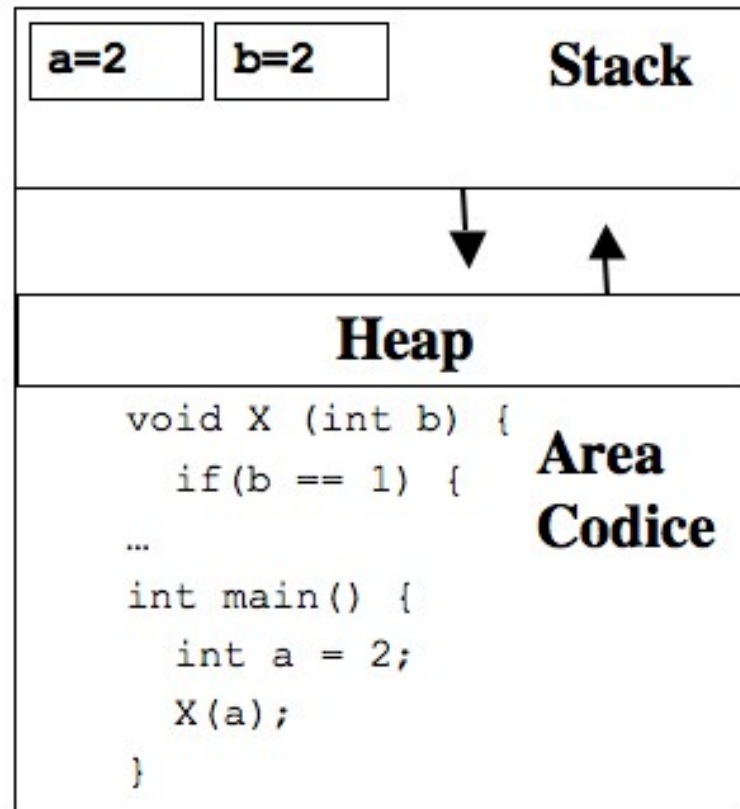
[informazioni sullo stato di esecuzione del programma])

SO: GESTIONE DEI PROCESSI

- .. Un programma

```
void X (int b) {  
    if(b == 1) {  
    ...  
int main() {  
    int a = 2;  
    X(a);  
}
```

- Stato di Esecuzione.



GESTIONE DEI PROCESSI

Definiamo **processo** l'attività di esecuzione di un **programma** sotto il controllo del sistema operativo

A ogni processo è associato lo stato corrente dell'attività che è detto **stato del processo**

GESTIONE DEI PROCESSI

Ogni processo ha un proprio task, ovvero una sua particolare e precisa mansione, e per eseguirla ha bisogno di determinate risorse. Una di queste, forse la più importante, è il **processore**.

Es. processo *"elaboratore di testo"* si serve del monitor, della memoria, della tastiera etc. (risorse fisiche) e una serie infinita di dati (risorse virtuali) per poter svolgere il proprio compito.

GESTIONE DEI PROCESSI

È un'istantanea del computer in un dato momento

Include i seguenti elementi:

- Il valore del contatore di programma (la posizione corrente del programma che viene eseguito)
- I valori dei registri della CPU
- Il valore delle celle di memoria assegnate al programma

Durante l'esecuzione di un programma (ovvero in momenti diversi del processo) **si osserveranno diversi stati del processo**

GESTIONE DEI PROCESSI

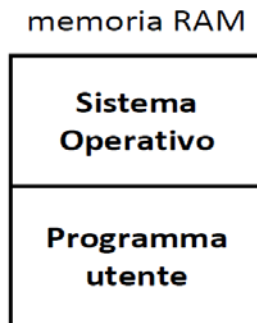
Senza l'intervento del sistema operativo un processo prosegue la sua esecuzione fino alla terminazione.

Alcuni processi – es. quelli che prevedono una continua interazione con l'utente – passano molto tempo nello stato di *Attesa* ad attendere come input un'azione (per esempio l'attivazione del mouse in una specifica posizione, oppure la digitazione di un testo) che viene intercettata dal sistema operativo e successivamente inoltrata al processo interessato.

S.O. monoprogrammato

S.O. in grado di gestire l'esecuzione di un programma utente alla volta, il che lo rende un software non molto complesso. Solo al termine dell'esecuzione di un programma, l'utente potrà lanciaerne un altro.

Un classico esempio di S.O. monoprogrammato è l'MS-DOS (Microsoft-Disk Operating System), divenuto oramai obsoleto.



SO multiprogrammato

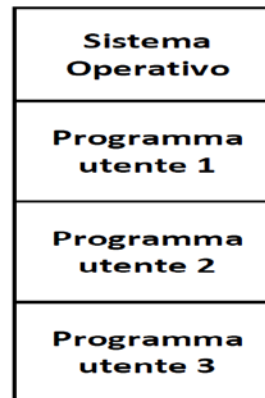
S.O. in grado di caricare in memoria più programmi utente contemporaneamente e di ripartire fra questi l'uso delle risorse.

Questi S.O. migliorano nettamente le prestazioni del sistema sfruttando i tempi morti della CPU dovuti alle operazioni di Input/Output, cioè in pratica quando un programma in esecuzione ha la necessità di svolgere una operazione di I/O, la CPU anziché restare senza far nulla in attesa del termine di questa operazione, viene assegnata ad un altro programma che deve essere eseguito.

SO multiprogrammato

Con una politica di gestione di questo tipo si cerca di tenere il più possibile la CPU al lavoro (tempo di attività di CPU molto elevato). La sensazione dell'utente è che il sistema stia eseguendo più programmi contemporaneamente, mentre in realtà è grazie alla velocità di esecuzione della CPU che passa da un programma all'altro, quindi in un dato istante è in esecuzione un solo programma.

memoria RAM



VANTAGGI e SVANTAGGI

I vantaggi dei S.O. multiprogrammati sono:

- Esecuzione di più programmi contemporaneamente;
- Migliore gestione delle risorse;

Gli svantaggi sono:

- Maggiore complessità del S.O.;
- Uso di algoritmi specifici per la gestione delle risorse: CPU, memoria e dispositivi di I/O;

In un S.O. multiprogrammato, quando un programma in esecuzione richiede di svolgere una operazione di I/O, la CPU viene liberata consentendole quindi di dedicarsi all'esecuzione di altri programmi.

S.O. Time-sharing

I S.O. multiprogrammati così come descritti sopra presentano un grosso problema: se un processo è in possesso della CPU e non fa operazioni di I/O, mantiene il possesso in modo esclusivo, non consentendo agli altri presenti in memoria di avanzare nell'esecuzione.

S.O. multiprogrammati in Time-Sharing, cioè a condivisione di tempo: S.O. assegna la CPU a un processo per un quanto di tempo predefinito chiamato time slice, terminato il quale il S.O. toglie la CPU al processo e la assegna al prossimo che ne ha diritto. Si ha l'impressione che tutti i processi vengano eseguiti contemporaneamente. Windows è un S.O. time-sharing.

Il coordinamento dei processi

Spetta al sistema operativo, che deve garantire:

- La **non interferenza** dei processi indipendenti
- La possibilità di **scambiare le informazioni** tra i processi che ne hanno bisogno
- La **disponibilità delle risorse** necessarie a ogni processo

Le attività di coordinamento dei processi sono svolte dallo scheduler e dal dispatcher.

Il dispatcher

È il componente del kernel che coordina l'esecuzione dei processi pianificati

Opera in questo modo:

1. Suddivide il tempo della CPU in brevi segmenti, chiamati **timeslice** o **quanto**
2. Assegna la CPU per il quanto di tempo **timeslice**, prefissato ad ogni processo.
3. Commuta l'assegnazione della CPU tra i processi.

Time sharing

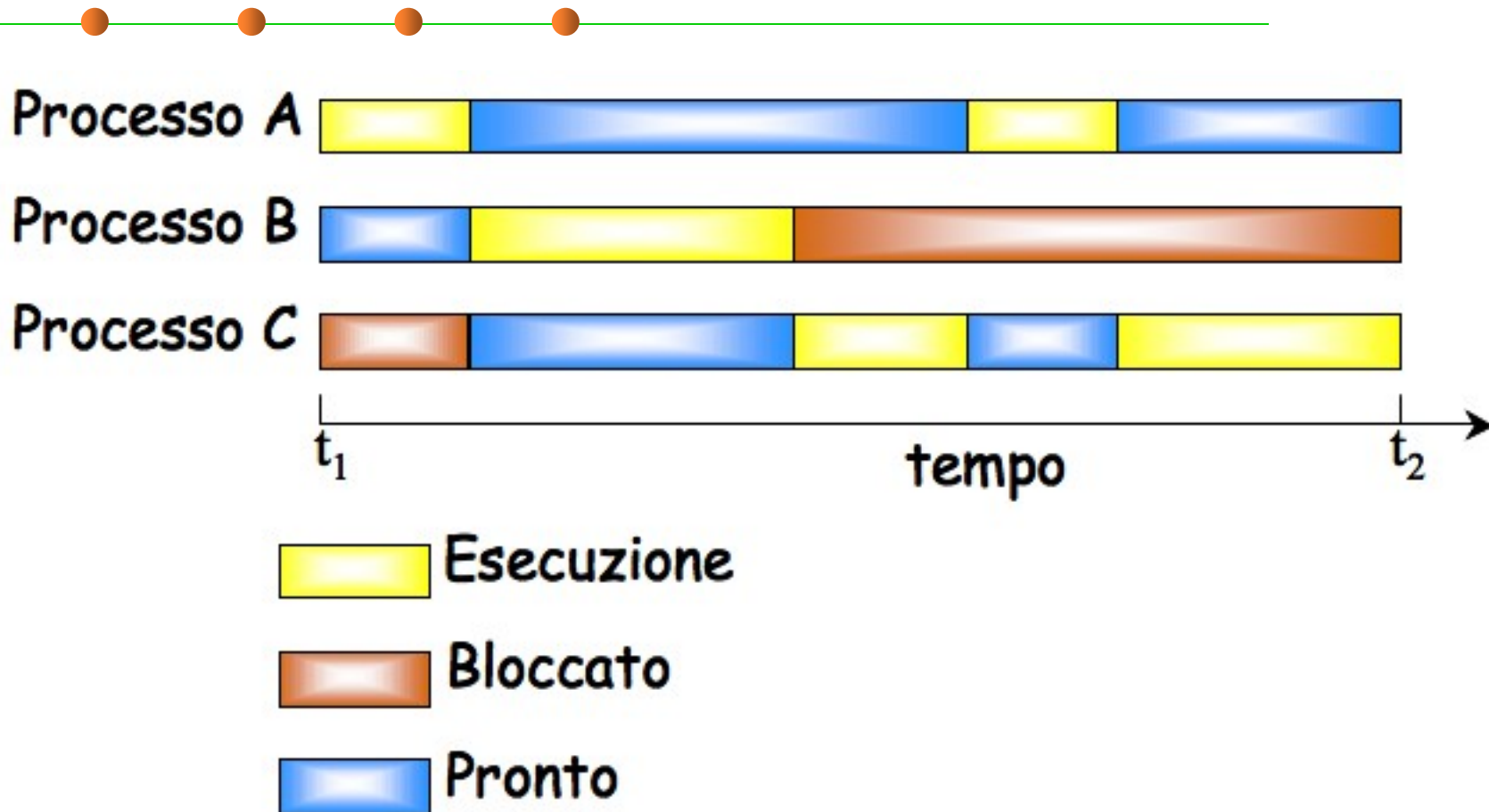
Allo scadere del quanto (oppure in attesa di un evento), l'esecuzione del processo viene sospesa e la CPU viene assegnata ad un altro processo. Il passaggio da un processo all'altro viene chiamato **commutazione di processo** oppure **commutazione di contesto**

Meccanismo che il SO adotta per eseguire "in parallelo" più processi (processi concorrenti).

Stati di un processo



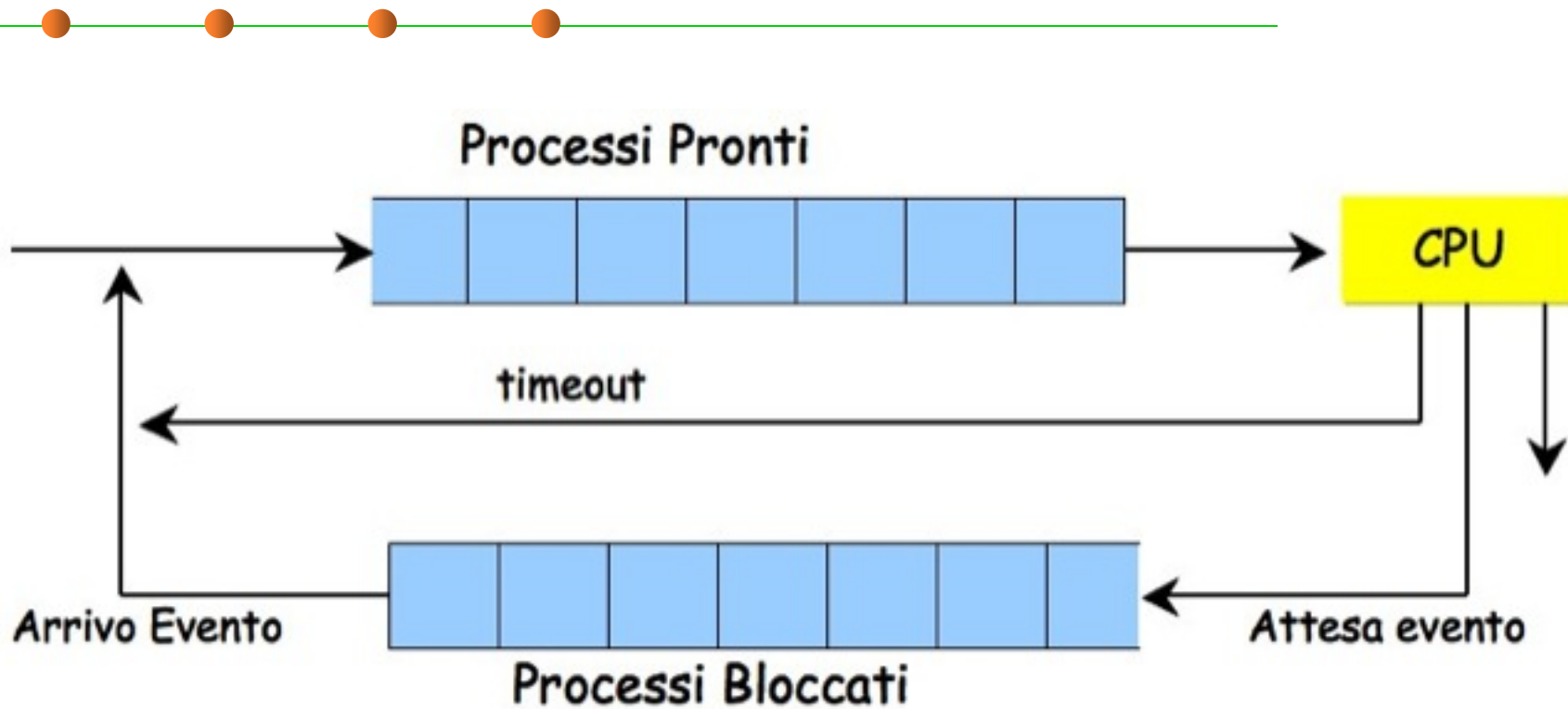
Assegnamento del processore



Assegnamento del processore

Il classico algoritmo di schedulazione denominato *round-robin* abbina la tecnica del *time-sharing* con la politica FIFO (*First-In First-Out*) di selezione dei processi in coda nello stato di *Ready* e consente all'utente del computer di avere la sensazione di esecuzione contemporanea di più programmi interattivi.

Assegnamento del processore



Gestione memoria principale

Memoria principale: risorsa essenziale all'interno di un elaboratore

- Un programma per essere eseguito deve risiedere all'interno della memoria principale e lo stesso discorso vale per i dati su cui esso opera
- Se molti processi devono essere eseguiti contemporaneamente, si dovrà trovare un meccanismo per permettere a questi processi di condividere l'uso della memoria principale

Memoria Centrale

La memoria centrale è costituita da una sequenza di celle, tutte della stessa lunghezza

- Ogni cella è caratterizzata da un indirizzo che viene utilizzato per leggere/scrivere le informazioni
- Di solito si sceglie di destinare al sistema operativo le prime celle della memoria al Sistema Operativo e le ultime celle al Processo (o ai Processi) utente (o viceversa)

Attivazione di un programma

Per essere eseguito, un programma deve essere portato (almeno in parte) in memoria centrale ed “essere attivato come *processo*” a partire da un indirizzo

- Quando un programma non è in esecuzione, non è strettamente necessario che stia in memoria centrale
- Coda di entrata: **processi su disco** che sono in attesa di essere caricati in memoria centrale per l'esecuzione

Sviluppo del software

Sistemi monoprogrammati: Un solo programma in memoria (obsoleto)

Sistemi multiprogrammati: più processi sono contemporaneamente pronti in memoria per l'esecuzione

- processi nel sistema devono coesistere nello stesso spazio di indirizzamento fisico
- i processi devono coesistere in memoria anche con il SO
- **Condivisione della memoria:** La memoria è logicamente partizionata in un'area di sistema e una per i processi utente
- **Separazione degli spazi di indirizzamento:** Le differenti aree di memoria devono essere separate in modo da non permettere ad un processo utente di corrompere il SO o addirittura bloccare il sistema

Vantaggi della multiprogrammazione

- rende più semplice programmare un'applicazione dividendola in due o più processi
- fornisce un servizio interattivo a più utenti contemporaneamente
- evita spreco di tempo di CPU dato che la maggior parte dei processi passa gran parte del tempo aspettando che vengano completate azioni di I/O del disco, in un sistema monoprogrammato durante questi intervalli di tempo la CPU non lavora

Gestore della memoria

- Ha il compito di gestire la memoria centrale (e una parte della memoria di massa) al fine di supportare l'esecuzione parallela dei processi
- Funzioni principali:
 - Allocazione
 - Protezione
 - Condivisione controllata
 - Sfruttamento delle *gerarchie di memoria*

Fase di caricamento del programma

- **Caricamento statico:** l'intero programma e tutti i suoi dati sono in memoria fisica
- La dimensione di un programma non deve superare la dimensione della memoria (fisica) disponibile
- **Caricamento dinamico:** si carica una porzione di programma solo quando viene richiamata
- Si evita di occupare memoria caricando tutto il programma

Swapping

- Un processo può essere temporaneamente scambiato (**swapped**) spostandolo dalla memoria **centrale** ad una memoria **secondaria (area di swap)** e poi in seguito riportato **interamente** in memoria centrale per continuarne l'esecuzione
- È detto **avvicendamento semplice**, o **swapping**
- Memoria secondaria: disco veloce abbastanza grande da accogliere le copie di tutte le immagini della memoria centrale per tutti gli utenti, e che fornisce accesso diretto a queste immagini

Swapping

- Lo **swapping** è molto comune nei sistemi con schedulatore Round Robin
- Il processo che finisce il quanto di tempo subisce lo swap-out
- **Roll out, roll in** è una variante dello swapping usata per algoritmi di schedulazione **basati sulla priorità**
- un processo a bassa priorità è scambiato con un processo ad alta priorità in modo che quest'ultimo possa essere caricato ed eseguito

Memoria secondaria

I dati sono memorizzati all'interno delle memorie di massa (permanenti)

Tempo di accesso (access time) delle memorie di massa, (tempo medio per ritrovare i dati e metterli a disposizione), è estremamente grande rispetto alla velocità di lavoro della CPU.

Access time delle memorie di massa: millisecondi ($10^{-3}s$) che è un'eternità rispetto alla velocità di elaborazione delle moderne CPU che hanno frequenze del clock dell'ordine dei GHz. Se non si vuole introdurre nel sistema un collo di bottiglia, richiedono dei tempi di accesso ai dati che siano dell'ordine dei nanosecondi ($10^{-9}s$)

Memoria secondaria

Soluzione: i dati sui quali un'applicazione deve compiere delle elaborazioni, prima che possano essere utilizzati dalla CPU, transitano attraverso la memoria centrale RAM, che è in grado di assicurare un access time dell'ordine dei nanosecondi, trasferendo in essa le varie porzioni (blocchi) che di volta in volta sono coinvolti nel processo di elaborazione.

Meccanismo gestito dal Sistema Operativo e, più nello specifico, dal File System che svolge tutte le funzioni di gestione dei dati, consentendo alle applicazioni di utilizzare i dati senza preoccuparsi delle operazioni più a basso livello necessarie per gli accessi di Input/Output (I/O)

Memoria secondaria

Organizzazione di dati e programmi in memoria secondaria:

- indipendente dal dispositivo fisico (nastri, dischi magnetici e ottici, ecc.)
- unità di memorizzazione: **file**
 - insieme di informazioni, correlate e registrate nella memoria secondaria, cui è stato assegnato un identificativo (nome).
 - una sequenza di byte accessibile in lettura e scrittura
- qualsiasi dato o programma si trova all'interno di un file
- Es.
 - Programmi
 - Dati (in rappresentazione binary)
 - Dati (in rappresentazione testuale)
 -

File System

- I file sono organizzati in strutture gerarchiche ad albero: **directory**
- Ogni file è **individuato da (almeno) un nome simbolico** mediante il quale può essere riferito (ad esempio, nell'invocazione di comandi o system call)
- Ogni file è **caratterizzato da un insieme di attributi** (proprietario, permessi di accesso, tipo, dimensioni, data di creazione e modifica, ecc.)

Allocazione dei file

I file devono essere allocati in blocchi (**record blocking**).
Blocco: unità di trasferimento dati tra disco e memoria principale.

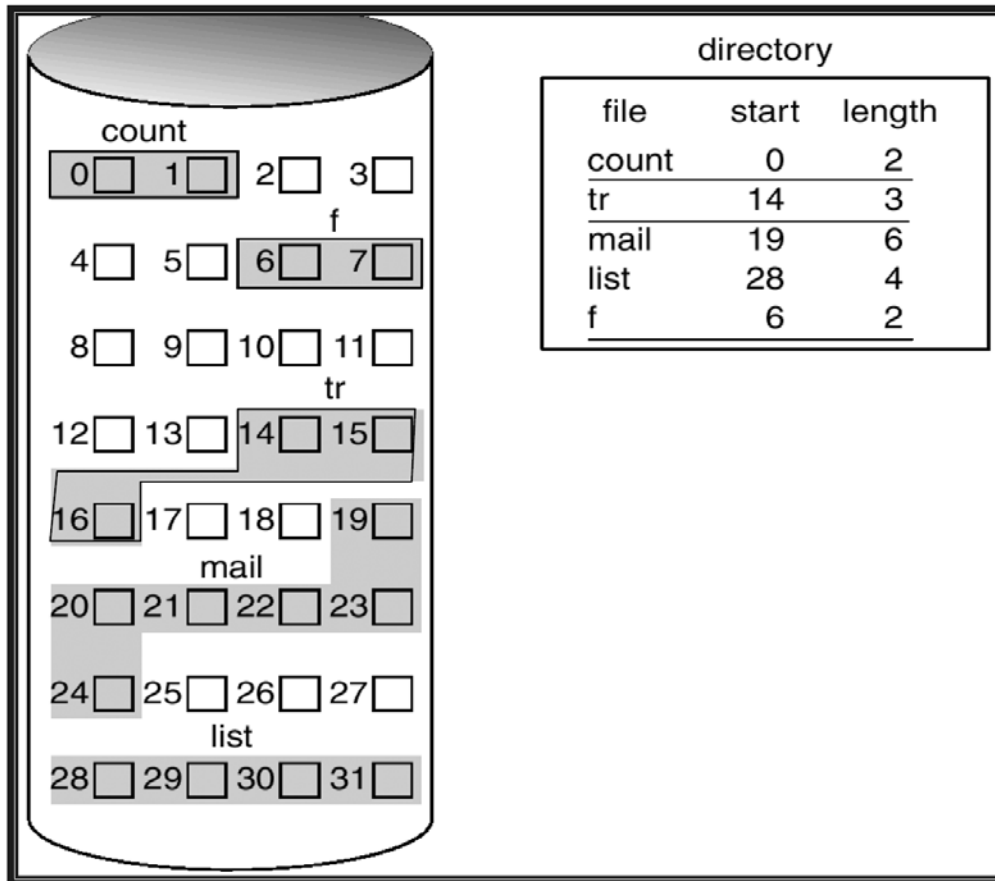
Generalmente i file che devono essere memorizzati hanno una dimensione molto maggiore di quella di un singolo blocco: è quindi necessario suddividere il file in più blocchi, ed è possibile adottare diverse tecniche per allocare i blocchi del medesimo file sul disco:

- ▶ allocazione contigua;
- ▶ allocazione concatenata;
- ▶ allocazione indicizzata.

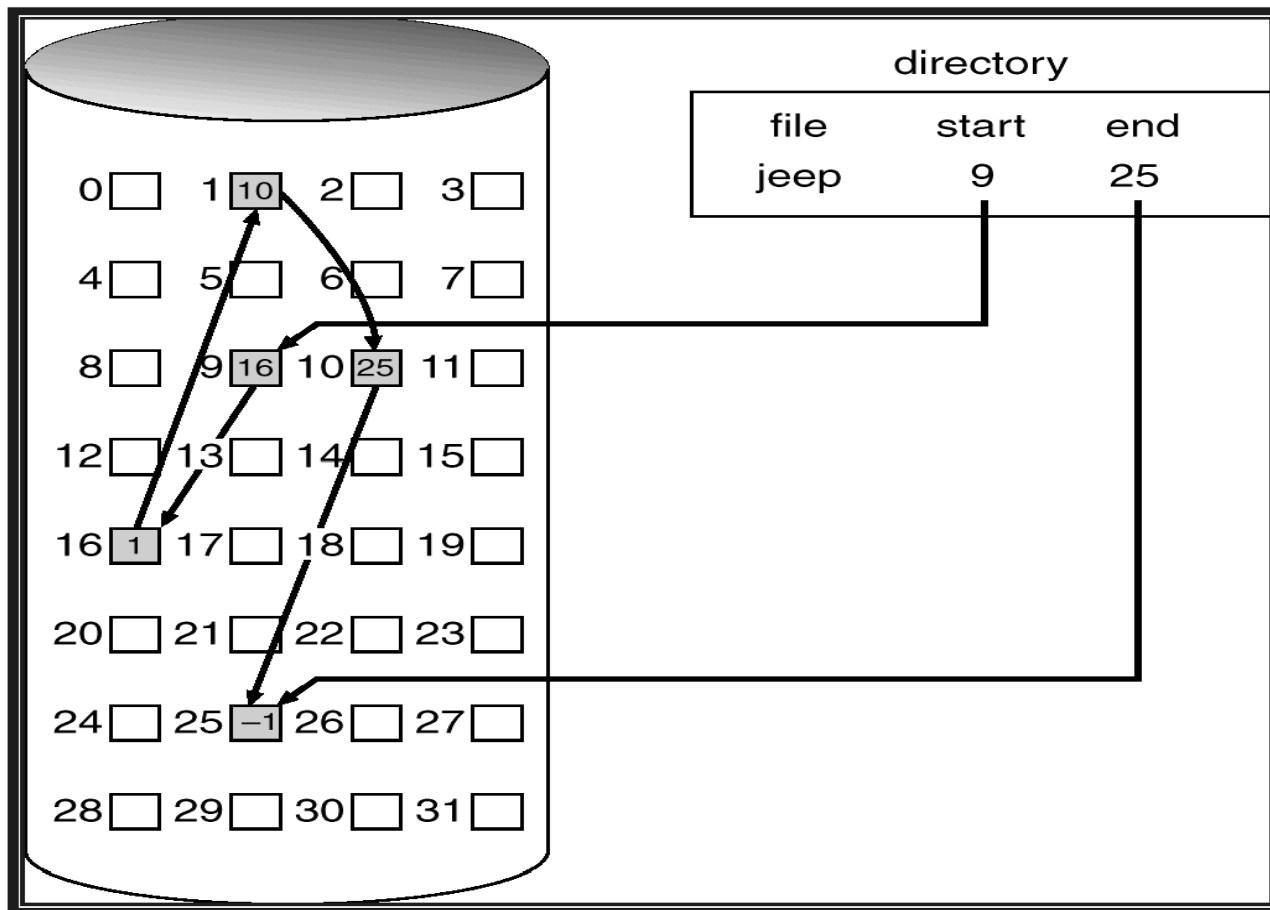
Sviluppo del software

1. **Allocazione contigua.** I blocchi del file sono allocati in blocchi consecutivi sul disco. Vantaggio: lettura rapida dell'intero file. Svantaggio: espansione difficoltosa.
2. **Allocazione con collegamenti.** Ogni blocco del file contiene un puntatore al blocco successivo. Vantaggio: espansione estremamente semplice. Svantaggio: scansione/lettura dell'intero file dispendiosa/lenta.
3. **Allocazione indicizzata.** Uno o più blocchi di indici contengono i puntatori agli effettivi blocchi del file.

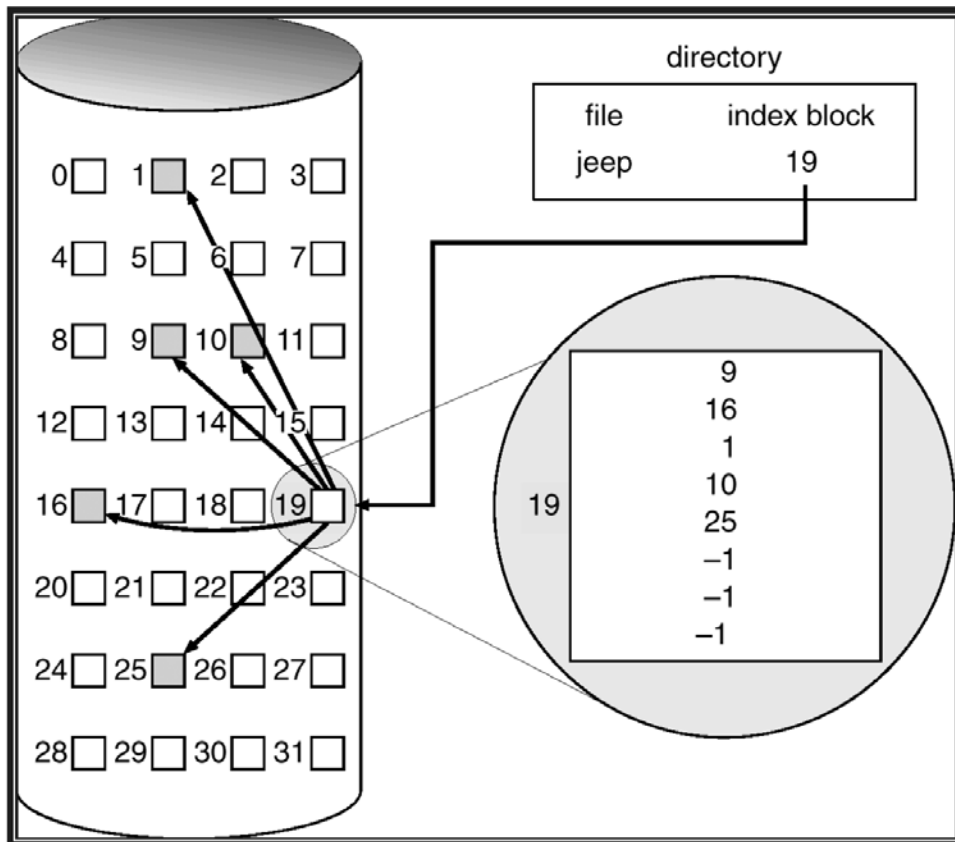
Allocazione contigua



Allocazione con collegamenti



Allocazione indicizzata



Gestione delle periferiche

Obiettivo principale del SO per la gestione delle periferiche: fornire una **visione astratta** per cui l'utente:

- abbia l'impressione di essere l'unico utilizzatore della periferica
- effettui le operazioni di I/O in maniera indipendente dalle caratteristiche fisiche del dispositivo usato

Il **SO** deve quindi farsi carico di:

- coordinare e gestire le richieste per le periferiche condivise (es. stampanti)
- rendere le operazioni di I/O indipendenti dal dispositivo fisico utilizzato

Gestione delle periferiche

Per ogni **dispositivo** di I/O esiste un processo di sistema detto **driver del dispositivo** che funge da interfaccia tra il dispositivo fisico e l'utente.

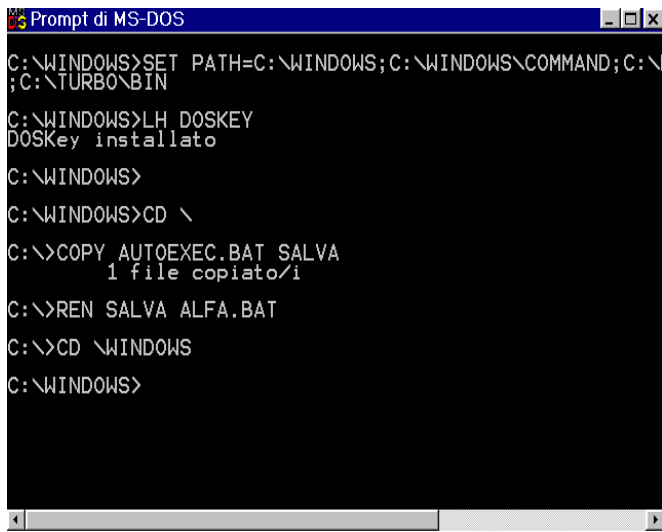
Un **driver di periferica**, in informatica, è un [programma](#) che gestisce o controlla un particolare tipo di [dispositivo](#), connesso ad un [computer](#).

Un driver di periferica consente a un dispositivo, per esempio una stampante, di comunicare con il [sistema operativo](#). I driver sono dipendenti dall'hardware e dal [sistema operativo](#) specifico.

SISTEMA OPERATIVO

Esempi

- MS-DOS (monotasking)
- MS-Windows: 95, NT, 98, 2000, XP, 10
- Unix: Linux, BSD, SCO, Solaris, ecc.
- MacOS, BeOS



```
Prompt di MS-DOS
C:\WINDOWS>SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\WINDOWS\COMMAND;C:\TURBO\BIN
C:\WINDOWS>LH DOSKEY
DOSKey installato
C:\WINDOWS>
C:\WINDOWS>CD \
C:\>COPY AUTOEXEC.BAT SALVA
1 file copiato/i
C:\>REN SALVA ALFA.BAT
C:\>CD \WINDOWS
C:\WINDOWS>
```

