

Cosa è un Sistema Operativo (S.O.)

Modulo software costituito da un insieme di programmi per:

permettere all'utente l'uso dell'elaboratore senza la conoscenza approfondita dell'hardware

» S.O. supporto all'utente

realizzare l'integrazione, l'ottimizzazione e la sincronizzazione della comunicazioni tra le componenti hardware

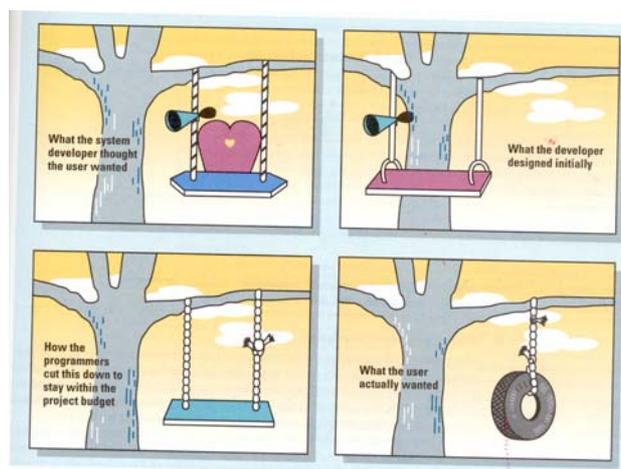
» S.O. come gestore delle risorse

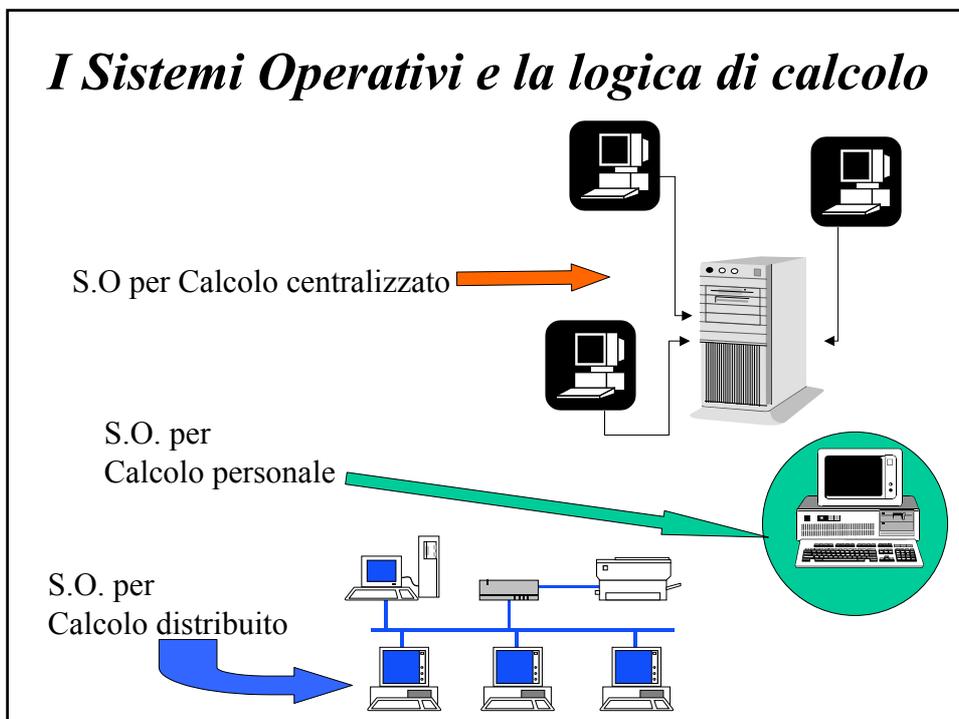
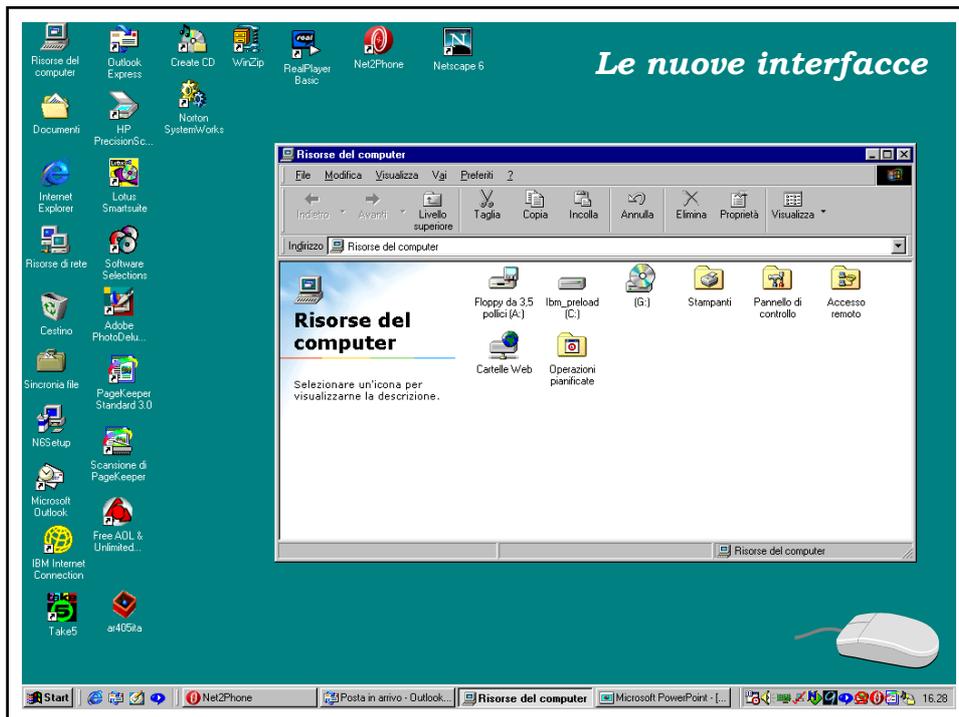
Sia il programmatore che l'utente preferiscono trattare con una ASTRAZIONE della macchina reale .

Il S.O. nasconde al programmatore la "verità" sui livelli hardware sottostanti e presenta un'interfaccia gradevole e amichevole all'utente

Evoluzione delle interfacce utente

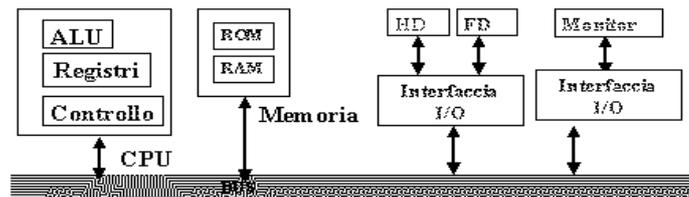
Dal What You See Is What You Get
al What You Get Is What You Want





S.O. come gestore delle risorse

- ▣ *Punto di vista Bottom Up (dal basso verso l'alto)*
- ▣ Gestisce tutte le componenti di un sistema complesso



►► Turn Around Time (T_{ta})

►► $T_{ta} = T_O - T_I$

►► è il tempo che passa tra l'ingresso dei dati e la produzione dell'output

►► Throughput

►► E' proporzionale al numero di lavori eseguiti dal sistema nell'unità di tempo

►► E' detto anche CARICO del SISTEMA

►► Overhead di Gestione

►► Sovraccarico Gestionale . E' proporzionale al tempo consumato dal S.O.

►► per eseguire il suo compito

Un sistema efficiente ...

Throughput Elevato

TurnAround Time Basso

Overhead Gestionale Basso

•Ma Quando il Turnaround è basso?

La storia dei S.O.

Per comprendere le funzionalità di un S.O. è comodo seguirne l'evoluzione storica

anni 40 (assenza di S.O.)

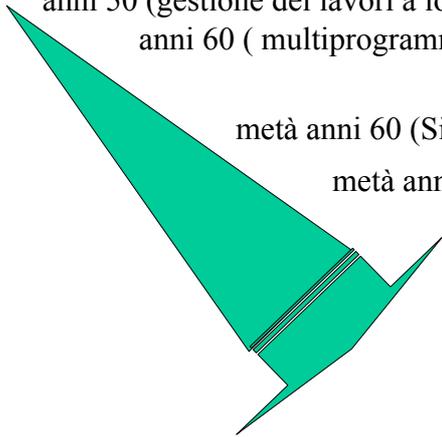
anni 50 (gestione dei lavori a lotti)

anni 60 (multiprogrammazione)

metà anni 60 (Sistemi General Purpose)

metà anni 70 (DB, Reti, S.Distribuiti)

oggi



Assenza di S.O.

- ✓ **Scrittura di Programmi in Linguaggio Macchina**
- ✓ **Caricamento ed Avvio del Programma a carico dell'Utente**
- ✓ **Tta minimo ... ma Throughput basso**

- inattività della CPU
- intervento UMANO tra l'esecuzione di due programmi
- gestione dell'input/output

Un passo avanti ...

• **Eliminazione dell'intervento umano**
(Proposta della general motors GM per un sistema IBM 701) con programmi che operano così:

MENTRE ci sono programmi da eseguire
FAI

- *carica programma in memoria*
- *esegui il programma*
- *controlla e riavverte lo stato del sistema per una successiva esecuzione*

FINE

Gestione a Batch

Per superare i tempi morti necessari per caricare ed eseguire i programmi uno per volta, si introducono i primi monitor che prevedevano un tipo di lavoro cosiddetto "batch" (a lotti) in cui i programmi (le schede) venivano caricati tutti insieme ed eseguiti in sequenza

Job Control Language (JCL)

Necessità di direttive per segnalare la fine di un programma ...

JCL è un linguaggio compreso dal S.O. e che consentiva di distinguere i vari programmi da eseguire

Con questo tipo di gestione, si automatizza la fase di inizializzazione e caricamento dei vari programmi, ma i programmi sono ancora eseguiti uno per volta in modo sequenziale

Per ridurre i tempi

Le operazioni di I/O vengono fatte "fuori linea" usando nastri magnetici più veloci degli altri organi di Input Output

- presenza di elaboratori dedicati più piccoli

La CPU ha una velocità molto elevata rispetto a quella dei dispositivi di I/O ...

il vero problema era quello di sfruttare i tempi morti della CPU dovuti alle operazioni di I/O.

Infatti, mentre i tempi di elaborazione di un carattere da parte della CPU sono dell'ordine dei nanosecondi (10⁻⁶ sec.), tempi tipici per stampare un carattere sono dell'ordine di circa 10 microsecondi (10⁻² sec.)

Canale dati

Per risolvere questo problema e ottimizzare l'uso della CPU, tra gli anni '50 e '60, viene introdotto una importante innovazione: il canale di dati.

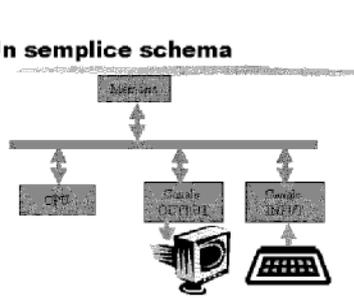
Tale dispositivo hardware e' l'elemento che sovrintende e controlla la comunicazione e la trasmissione dei dati tra l'elaboratore centrale e i dispositivi di I/O, sollevando la CPU da tale compito

Il CANALE è un Processore dedicato al controllo dell'unità periferica

- La memoria centrale e' divisa tra CPU e il canale e contiene i dati e programmi di entrambi
- Un Canale, attivato dalla CPU, evolve in modo asincrono e parallelo alla CPU

Un esempio di canale dati

Un semplice schema



Con l'introduzione dei canali di dati si risolve il problema della sovrapposizione delle operazioni di I/O ma e' necessario poter sovrapporre le operazioni di calcolo e quelle di I/O (**interrupt**)

Multiprogrammazione

**Possibilita' di caricare piu' programmi in memoria
in grado di condividere l'uso della CPU e delle risorse
dell'elaboratore**

- E' possibile che il S.O., per realizzare la multiprogrammazione, non mantenga tutti i programmi in memoria ma carichi di volta in volta quello piu' idoneo da un supporto di massa
- Tale tecnica, detta di "swapping", e' poco efficiente, ma risulta necessaria quando si hanno problemi di dimensione della memoria

Processo = Programma in esecuzione

Nucleo (o Kernel) = gestore dei processi

Problema: nella multiprogrammazione più programmi sono in memoria, ma una sola è la risorsa processore.

Come assegnare un processo al processore?

Al processo è associato uno stato ed una priorità

Stato dei Processi

Ready pronto ad essere eseguito, ma non in esecuzione
running in esecuzione
waiting in attesa di un evento o di un dato

- Un *processo* è un **programma in esecuzione**. Un processo necessita di alcune risorse per assolvere il proprio compito: tempo di CPU, memoria, file e dispositivi di I/O.
- *Il SO è responsabile delle seguenti attività in connessione con la gestione dei processi:*
 - **Creazione e cancellazione di processi.**
 - **Sospensione e riattivazione di processi.**
 - **Fornire meccanismi per:**
 - » **sincronizzazione di processi**
 - » **comunicazione fra processi**

Il Supervisore E' il cuore del S.O e ..

prende le decisioni sulle modalità di attribuzione delle risorse, quando un processo si ferma per terminazione o attesa di I/O

Il Supervisore interviene

- quando un processo fa richiesta di I/O
- quando il canale termina le sue operazioni
- quando un processo in running termina la sua esecuzione

Priorità

**Come scegliere nel caso di più processi pronti per l'esecuzione o in attesa di uno stesso canale di I/O?
Vengono associati ai processi delle priorità (numeri interi)**

- STATICHE fissate dal gestore UNA TANTUM
- DINAMICHE il valore viene alterato durante l'esecuzione

La priorità da sola non basta ...

- ...PIU' PROCESSI CON LA STESSA PRIORITA'
A ciascuna risorsa viene allora associata una coda di richieste contenente
- l'identificatore di processi (PID)
- la priorità
Il Supervisore sceglie il processo in testa con la massima priorità

Sistemi Time Sharing

Un processo rimane in running solo per un intervallo fissato di tempo (Time Slice**)**

Per evitare che resti padrone della CPU a tempo indeterminato

Il Supervisore riacquista il controllo...

- quando è terminata la time slice ,
- il programma termina la sua esecuzione ,
- il programma avanza una richiesta di I/O.

È ovvio che un programma per terminare la sua esecuzione avrà bisogno di più time slice . Così...

- aumenta il turnaround time dei singoli processi
- migliora il throughput complessivo

Il File System

Dal punto di vista dell'utente il File System è caratterizzato dalle operazioni che ha a disposizione, cioè ...

- come si definisce un file
- come il file vengono denominati e protetti
- che operazioni sono permesse sui file e così via

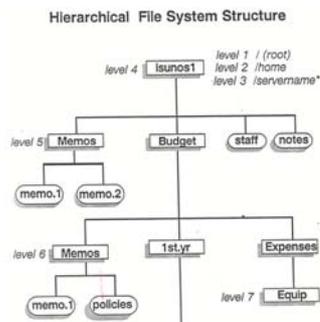
Per tenere traccia dei file, il File system mette a disposizione dell'utente dei **directory contenenti un certo numero di elementi, uno per file**

E' molto comune che un utente voglia raggruppare i suoi file in modo logico, attraverso una struttura gerarchica (ad albero)

Memorizzazione dei File

Un file è di solito composto da una sequenza di blocchi . Il File System deve tenere traccia dei blocchi di ciascun file Negli ambienti Microsoft ad ogni disco è associata una tabella d'allocatione dei file o FAT, contenente un elemento per ogni blocco del disco.

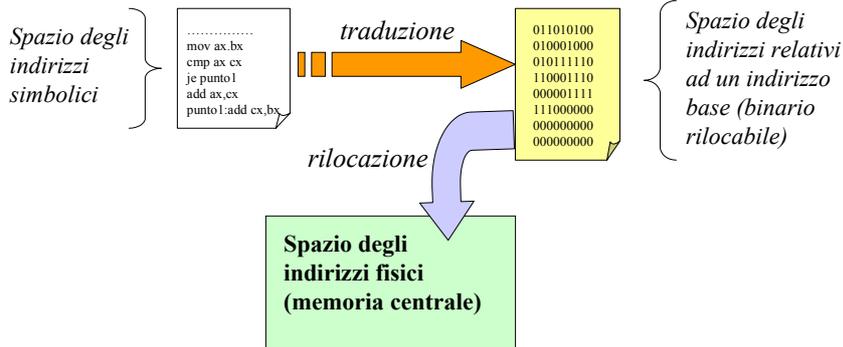
- Il directory contiene il numero di blocco del primo blocco dei file



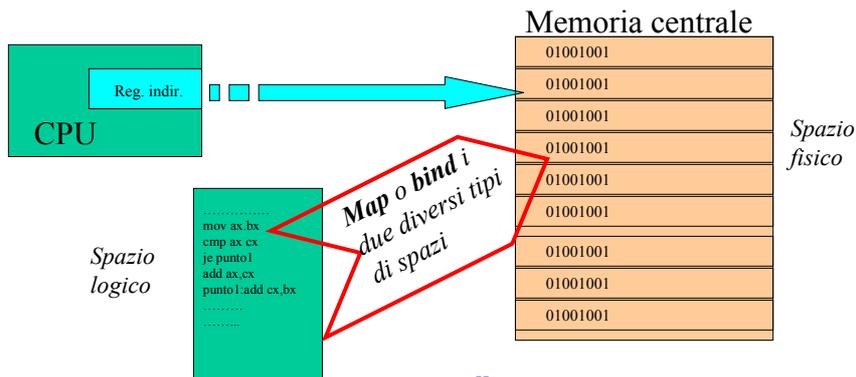
Il Gestore della Memoria

Il compito del SO è anche di gestire la Memoria Centrale:

- ricordare quali parti della memoria sono attualmente in uso e quali non lo sono
- assegnare la memoria ai processi ,
- deallocare la memoria assegnata ,
- amministrare gli scambi tra memoria centrale e disco quando la prima non è più in grado di contenere i processi (swapping).

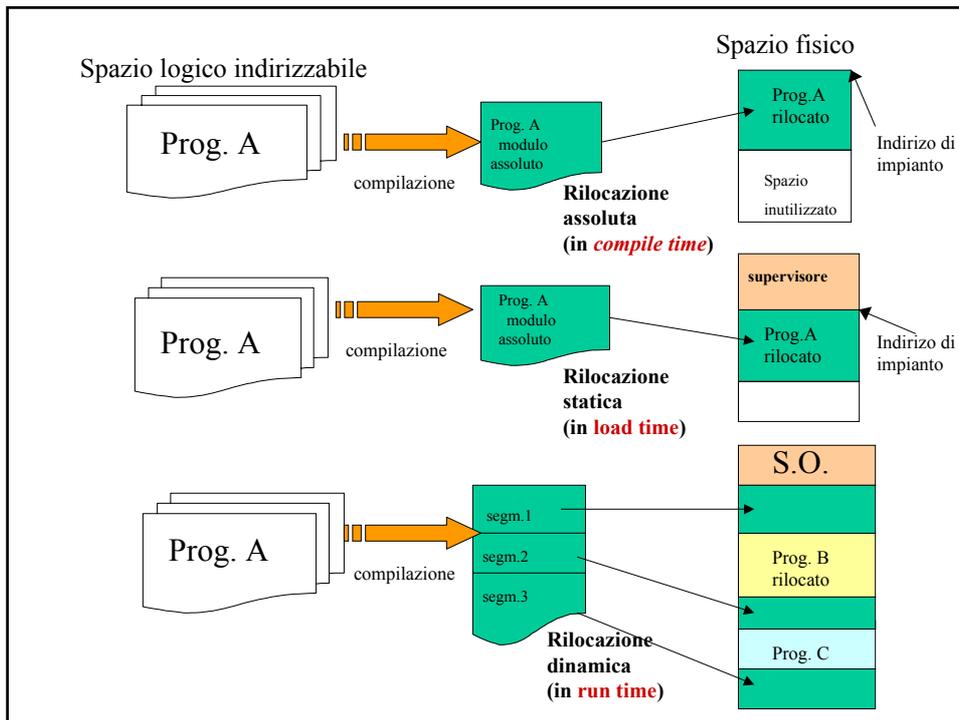


Il Gestore della Memoria



- ✓ **allocare** ai programmi la quantità di memoria necessaria
- ✓ **rilocare** i programmi , (tradurre i nomi dello spazio indirizzabile in locazioni di memoria centrale)

- **rilocazione assoluta** = prima che il programma sia caricato in memoria centrale
- **rilocazione statica** = quando il programma viene caricato in memoria
- **rilocazione dinamica** = durante l'esecuzione del programma



Tecniche di mapping tra spazio logico e spazio fisico

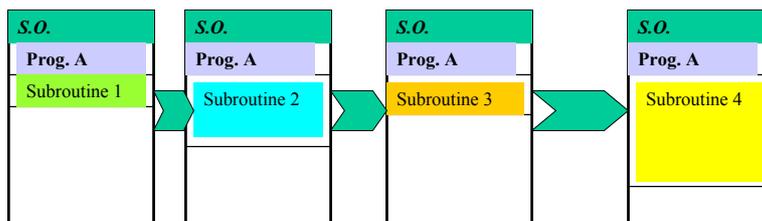
In sistemi uniprogrammati

➤ - sequenziale :

consiste nel suddividere l'area di lavoro in due parti : una riservata al Sistema Operativo e l'altra assegnata al programma utente

➤ - overlay:

il programma viene suddiviso in blocchi, ciascuno dei quali contiene una entità logica a sé stante, che può venire caricata al momento del suo utilizzo e rilasciata subito dopo.



Tecniche di mapping tra spazio logico e spazio fisico

In sistemi multiprogrammati

➤ Partizionamento:

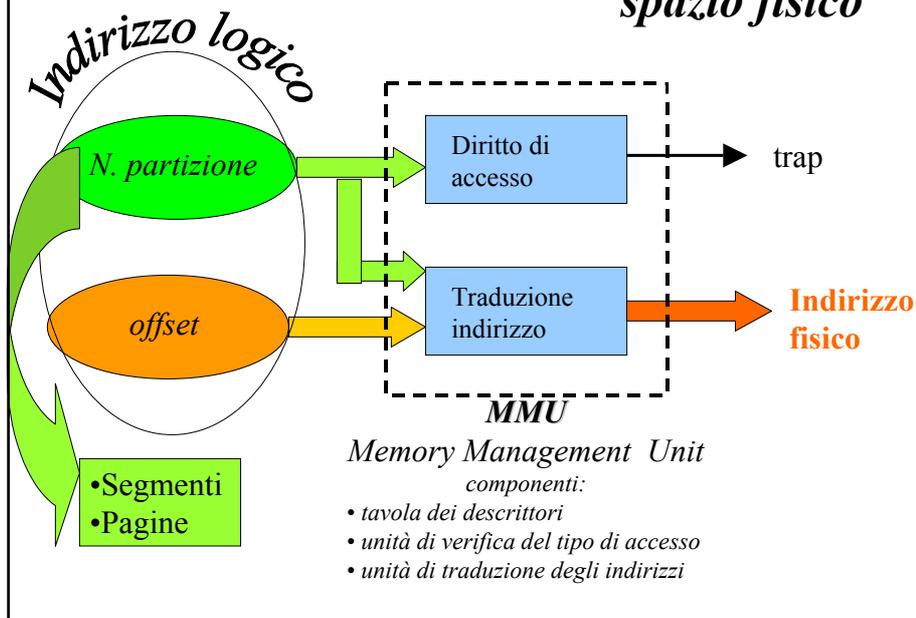
La memoria fisica viene suddivisa in un certo numero di aree (PARTIZIONI) di dimensioni prefissate, ciascuna delle quali può essere assegnata separatamente ad un lavoro.

Le partizioni possono essere create PRIMA che i vari programmi vengano caricati in memoria (P. STATICO) o all'atto del caricamento dei programmi (P. DINAMICO). Questa tecnica è efficiente se si dispone di un meccanismo di rilocazione dinamica

In ogni caso :

occorre effettuare un mapping da indirizzo a singola componente ad indirizzo a doppia componente → area di indirizzi (partizione) e offset (indirizzo nell'area)

Tecniche di mapping tra spazio logico e spazio fisico



PAGINAZIONE

Partizioni Fisse

La memoria è divisa in N partizioni (pagine) di lunghezza uguale

Lo spazio degli indirizzi fisici e quello degli indirizzi logici viene diviso in blocchi della stessa dimensione (in genere da 1 a 4 KB)

Tavola dei descrittori delle pagine:

N. pag.	Page frame	Start address su disco	Modalità di accesso
0	1	3000	100
1	5	20000	110
2	3	200	001
.....

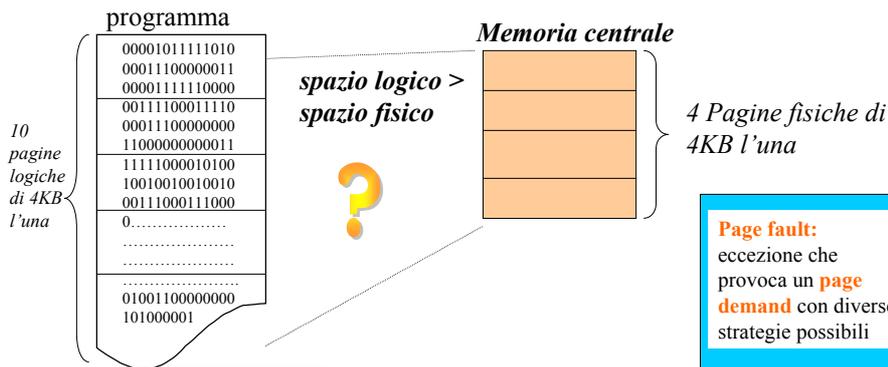
Nel programma

Opcode	num.pag.	ind.relat.
--------	----------	------------

Page frame	offset
------------	--------

r w e

Memoria Virtuale *Per la gestione dinamica della paginazione (demand paging memory management)*



N. pagina logica = indice tavola descrittori

Tavola dei descrittori delle pagine:

flag	Page frame	Start address su disco	Modalità di accesso ...
1	1	3C4T8S	100
0	5	1C2T8S	110
1	3	2C2T3S	001
.....

in mem. Centrale →
non in mem. centrale →

SEGMENTAZIONE

Segmento = raggruppamento logico di informazioni
partizioni di lunghezza diversa che contengono
ciascuno un solo oggetto di elaborazione !

A differenza della paginazione, necessità di :

- ✓ - controllare che l'offset non superi la lunghezza del segmento,
- ✓ - controllare le modalità di accesso

Tavola dei descrittori dei segmenti:

