Sistemi Operativi

AAF - Secondo anno - 3CFU
A.A. 2023/2024
Corso di Laurea in Matematica

La Gestione della Memoria - Parte 2

Annalisa Massini

Dipartimento di Informatica
Sapienza Università di Roma

Argomenti trattati

1. Gestione della memoria
   - Paginazione
   - Segmentazione
2. Memoria virtuale
   - Memoria virtuale: concetti generali

Gestione della memoria

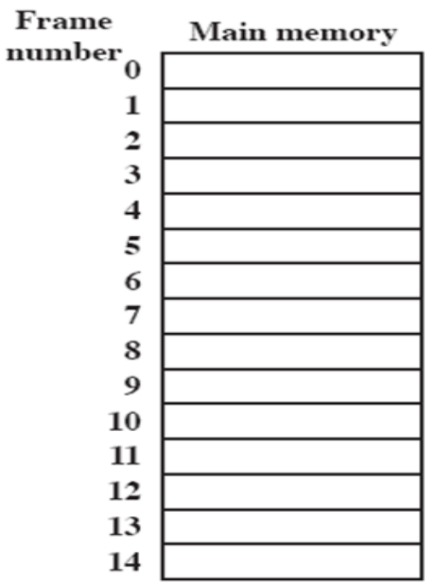
Paginazione

Paginazione (Semplice)

- Sia il partizionamento fisso che quello dinamico sono inefficienti a causa della frammentazione
- Con la **paginazione** assumiamo che:
  - la memoria sia divisa in piccole parti di grandezza uguale: **frame**
  - i processi vengano anch'essi partizionati in parti: **pagine**
  - una pagina ed un frame hanno la **stessa dimensione**
  - ma ci sono **più pagine che frame**
- Per essere usata la pagina deve essere collocata in un frame
  - una pagina può essere messa in un *qualunque* frame
  - pagine contigue possono essere messe in frame distanti

Paginazione

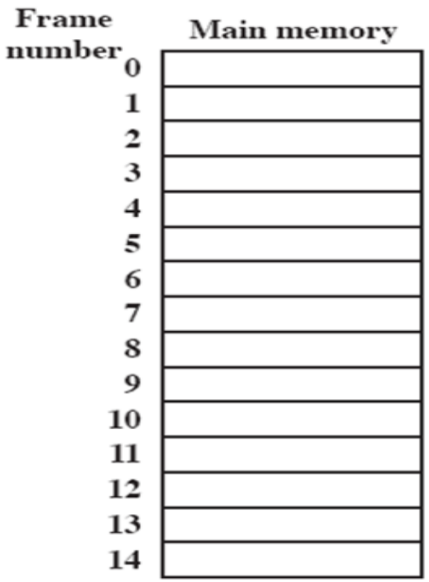
- I SO che adottano la paginazione mantengono una **tabella delle pagine** per ogni processo
- La tabella specifica in quale **frame effettivo** si trova per ogni pagina del processo
- Un indirizzo di memoria può essere visto come un *numero di pagina* e uno *spiazzamento* al suo interno
- Quando c'è un process switch, la tabella delle pagine del nuovo processo deve essere caricata



Paginazione

Esempio

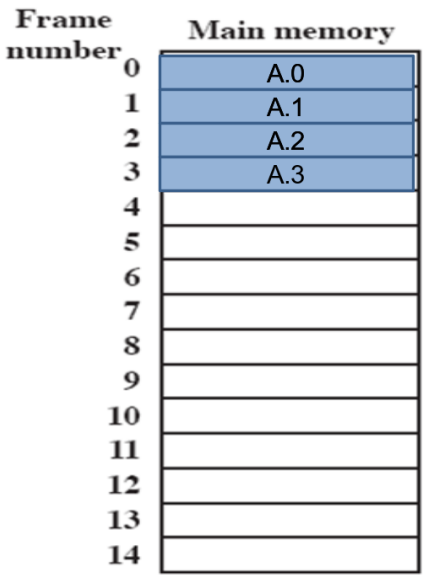
- Il SO deve sempre tenere aggiornata la lista dei frame liberi
- Quando è il momento di caricare un processo, il SO cerca il numero di frame liberi per caricare quel processo
- All'inizio tutti i frame sono liberi



Paginazione

Esempio

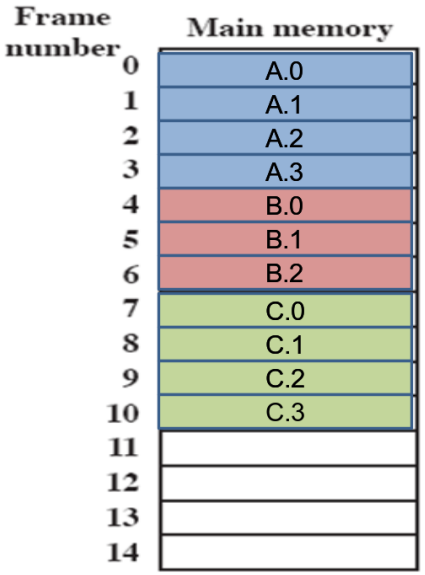
- Il **processo A**, memorizzato sul disco rigido, consiste di 4 pagine
- Quando è il momento di caricare il processo, il SO cerca 4 frame liberi
- Le pagine vengono caricate in memoria nei primi 4 frame



Paginazione

Esempio

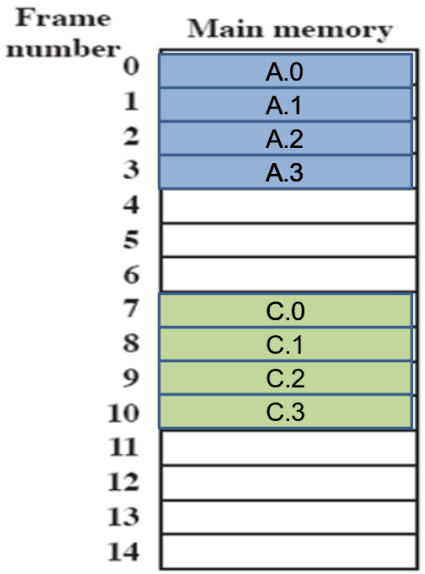
- Successivamente vengono caricati il **processo B**, che consiste di 3 pagine, e poi il **processo C**, che consiste di 4 pagine
- Ad un certo punto tutti i processi sono bloccati e il SO vuole caricare un nuovo processo



Paginazione

Esempio

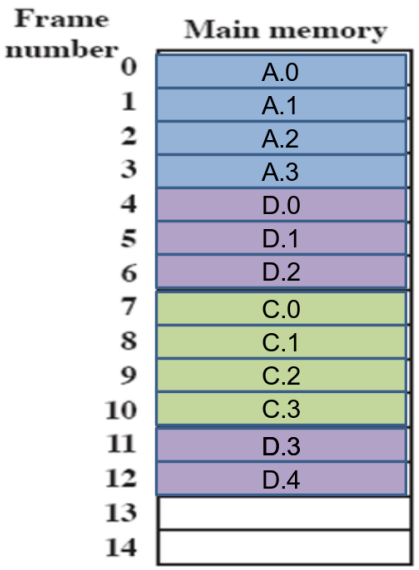
- Il processo B (bloccato) viene scelto per essere swappato in memoria secondaria e viene portato nello stato Suspended
- Il SO vuole poi portare in memoria principale il **processo D**, che consiste di 5 pagine
- Però non ci sono 5 frame liberi contigui



Paginazione

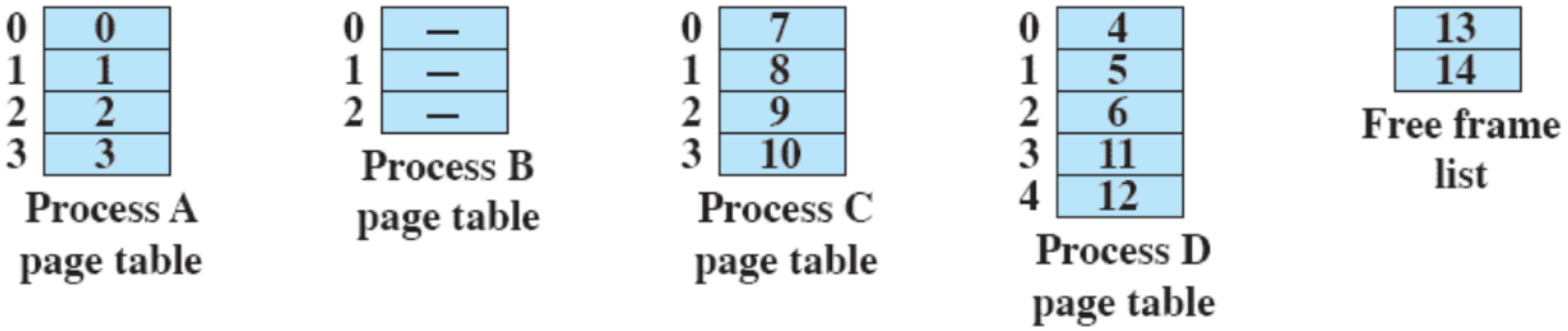
Esempio

- Le cinque pagine del **processo D** vengono caricate nei frame 4-5-6 e 11-12
- Per la realizzazione della paginazione serve una **tabella delle pagine** per ogni processo per memorizzare in quali dei frame sono allocate le diverse pagine
- Con il partizionamento dinamico, non sarebbe stato possibile caricare il processo D in memoria



Paginazione: Esempio

Tabelle delle pagine per i processi attivi (non suspended)



Paginazione

- Per ottenere l'indirizzo fisico non basta avere solo base register, ma si usa la tabella delle pagine
- La traduzione da indirizzo logico a indirizzo fisico è fatta con il supporto dell'hardware.
- Il processore usa l'informazione riguardante il frame in cui collocata la pagina presente nella tabella delle pagine
- L'indirizzo logico (page number-offset) viene trasformato in indirizzo fisico (frame number-offset)

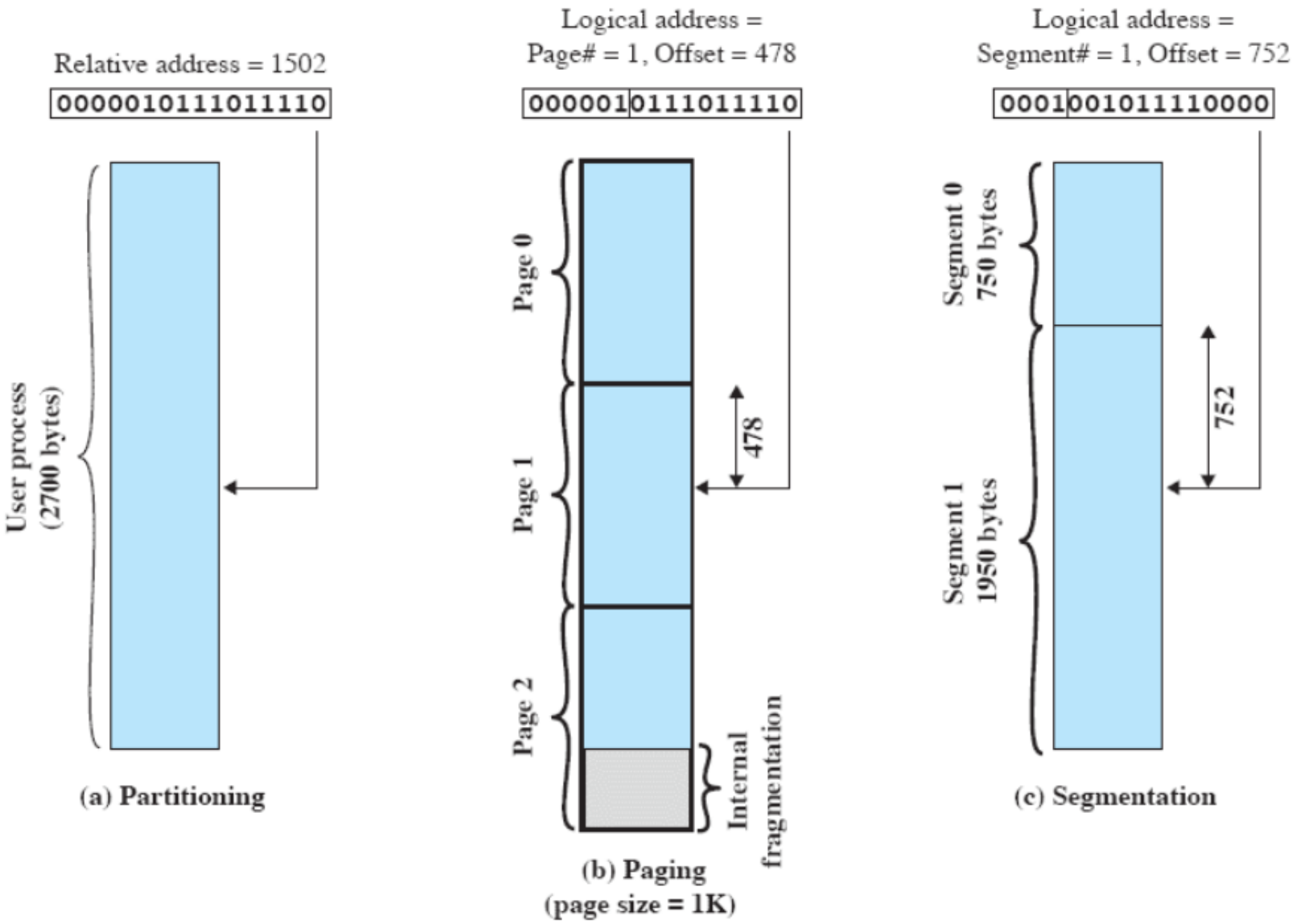
Gestione della memoria

Segmentazione

Segmentazione (Semplice)

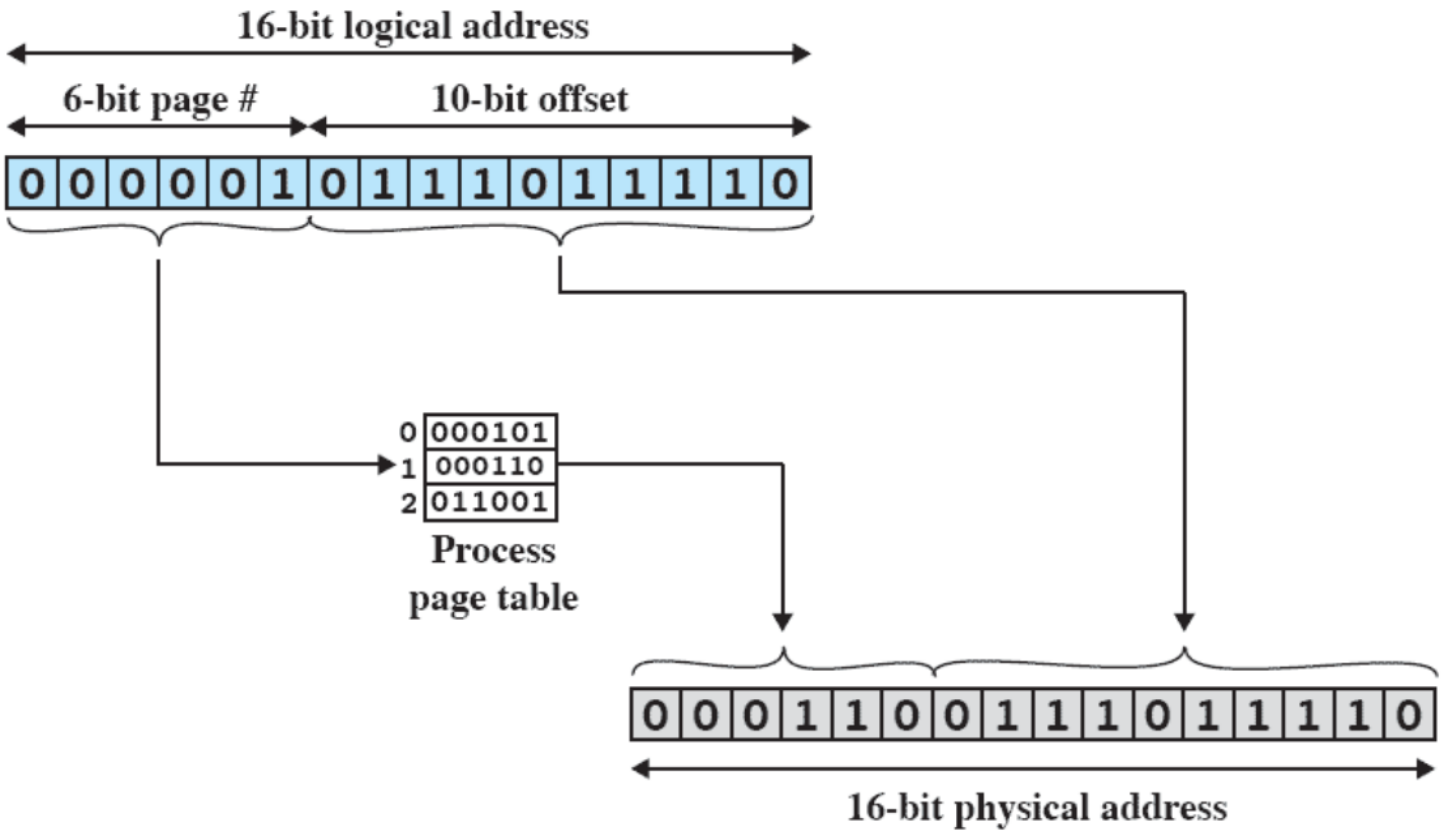
- I programmi vengono divisi in **segmenti**:
  - di dimensione (lunghezza) variabile
  - con un limite massimo alla dimensione
- Idea simile al partizionamento dinamico ma con una differenza fondamentale:
  - il programmatore o il compilatore devono gestire esplicitamente la segmentazione
  - cioè dire quanti segmenti ci sono e qual è la loro dimensione
  - e metterli effettivamente in RAM
- Il SO si occupa invece di *risolvere gli indirizzi* con supporto hardware
- Un indirizzo di memoria è ccomposto dal **numero di segmento** e da uno **spiazzamento** al suo interno

Indirizzi Logici



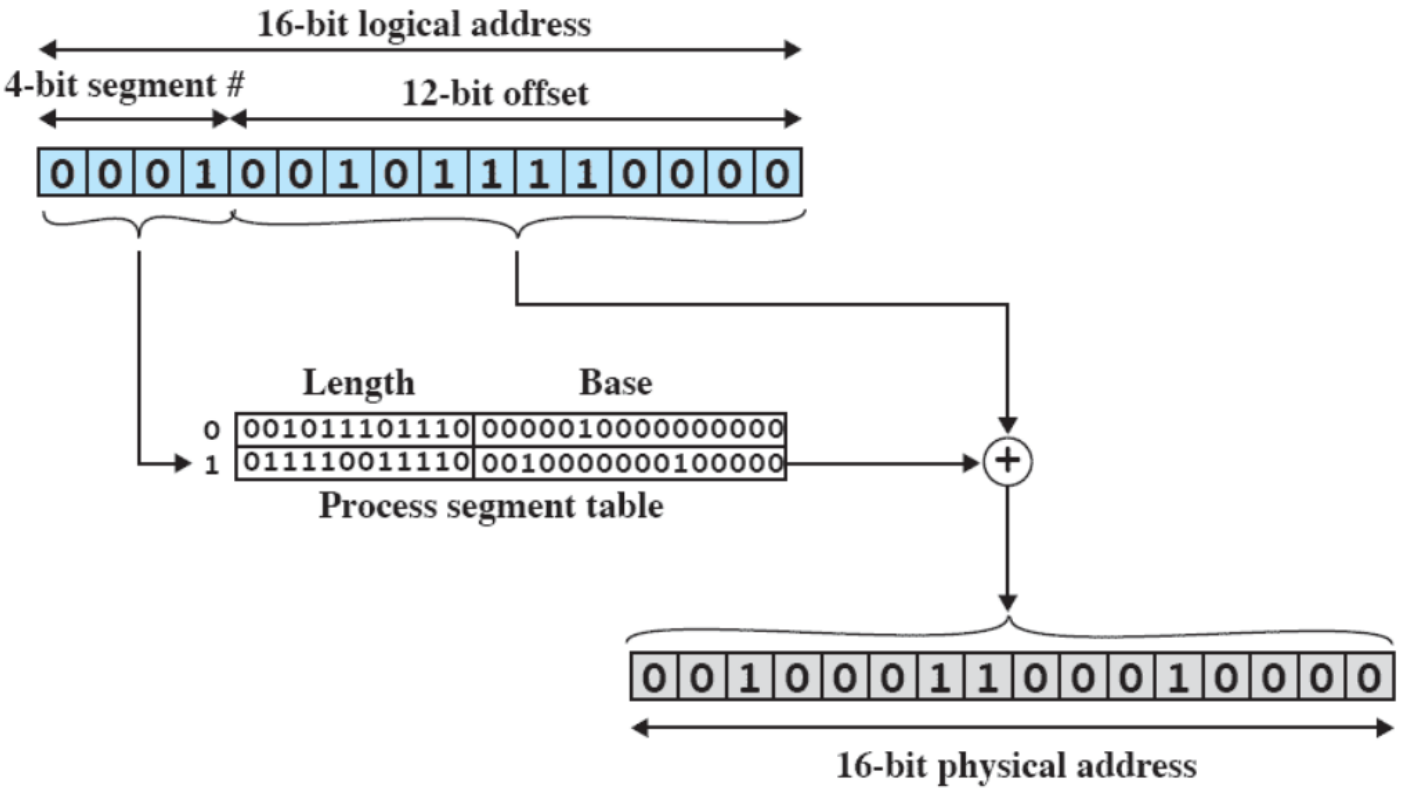
Paginazione

Per ogni processo, il numero di pagine è al più il numero di frames (non sarà così con la memoria virtuale)



Segmentazione

Con la segmentazione le cose sono leggermente diverse
Si usa la tabella dei segmenti (analoga alla tabella della pagine)



(b) Segmentation

Memoria virtuale

Memoria virtuale: concetti fondamentali

Gestione della Memoria: concetti fondamentali

- Il confronto tra partizionamento fisso e dinamico con **paginazione** e **segmentazione**, danno l'intuizione della svolta nella gestione della memoria che ha portato alla **memoria virtuale**
- Due caratteristiche di **paginazione** e **segmentazione** sono la chiave della svolta

Gestione della Memoria: concetti fondamentali

1. Tutti i riferimenti di memoria in un processo sono *indirizzi logici* tradotti in indirizzi fisici a tempo di esecuzione
   - così un processo può essere spostato più volte dalla memoria principale alla secondaria e viceversa durante l'esecuzione, occupando ogni volta zone di memoria diverse
2. Un processo può essere spezzato in *più parti* (pagine o segmenti), che non necessariamente occuperanno una zona contigua di memoria principale
   - si sfrutta la traduzione dinamica dell'indirizzo e la tabella della pagine o dei segmenti

La svolta: idea chiave

L'**idea chiave** della svolta è basata sulle seguenti osservazioni:

- Non occorre che tutte le pagine o tutti i segmenti di un processo siano in memoria principale durante l'esecuzione (e il processo venga concesso il processore)
- Se la successiva istruzione da eseguire e i dati su cui eseguirla sono in memoria principale, allora l'esecuzione può andare avanti (almeno per un po')

Memoria Virtuale: Terminologia

Memoria virtuale: schema di allocazione di memoria, in cui la memoria secondaria può essere usata come se fosse principale

- gli indirizzi usati nei programmi e quelli usati dal sistema sono diversi
- c'è una fase di traduzione automatica dai primi nei secondi
- la dimensione della memoria virtuale è limitata dallo schema di indirizzamento, oltre che ovviamente dalla dimensione della memoria secondaria
- la dimensione della memoria principale, invece, non influisce sulla dimensione della memoria virtuale

Memoria Virtuale: Terminologia

Indirizzo virtuale: l'indirizzo associato ad una locazione della memoria virtuale, alla quale si accede come se fosse parte della memoria principale

Spazio degli indirizzi virtuali: la quantità di memoria virtuale assegnata ad un processo

Spazio degli indirizzi: la quantità di memoria assegnata ad un processo

Indirizzo reale: indirizzo di una locazione di memoria principale

Memoria virtuale ed esecuzione di un processo

- Il SO porta in memoria principale alcune **porzioni** (pagine per *paging* o segmenti per *segmentation*) del programma
- All'inizio vengono portati solo (uno o) poche porzioni, cioè porzione iniziale di programma e porzione iniziale di dati
- La porzione di processo in memoria principale viene chiamato **resident set** (*insieme residente*)
- Se il processore trova un indirizzo logico che non è residente in memoria principale, genera un interrupt per *memory access fault*

Memoria virtuale ed esecuzione di un processo

- Il SO mette il processo in modalità blocked - si tratta di una richiesta di I/O a tutti gli effetti
- Affinchè il processo possa riprendere l'esecuzione, il SO deve portare in memoria principale la porzione di programma contenente l'indirizzo logico che ha causato l'interruzione
- Vengono eseguite le seguenti operazioni:
  - SO esegue richiesta di lettura su disco (I/O)
  - Un altro processo viene portato in esecuzione
  - Quando la porzione mancante viene portata in memoria principale, il controllo viene ridato al SO tramite un'interruzione
  - Il SO porta il processo blocked a ready

Vantaggi per il sistema

1. Più processi possono essere in memoria principale
   - Solo alcune parti di ciascun processo vengono portate in memoria principale
   - Questo vuol dire che è molto probabile che ci sia sempre almeno un processo ready
   - Uso più efficiente del processore
2. Un processo può richiedere più dell'intera memoria principale
   - Viene eliminata una delle principali complicazioni per il programmatore (conoscere la dimensione della memoria e suddividere il programma)
   - Con la memoria virtuale (paginazione o segmentazione), se ne occupa il sistema operativo con il supporto dell'hardware
   - Il programmatore vede la memoria grande come il disco rigido

Memoria Reale e Virtuale

- **Memoria reale**: è la memoria principale (la RAM)
- **Memoria virtuale**: è quella percepita dal programmatore e corrisponde alla memoria secondaria (cioè al disco rigido)
  - permette di avere una multiprogrammazione elevata
  - libera il programmatore dai vincoli della memoria principale

Problemi legati alla memoria virtuale

- La memoria virtuale basata su paginazione oppure su paginazione + segmentazione è una componente fondamentale dei moderni SO
- Però è stata oggetto di molte discussioni in passato
- **Esempio**:
  - abbiamo un programma molto grande che ha bisogno di un grande numero di array di dati di grandi dimensioni
  - se c'è un salto a un'istruzione o servono dati non presenti in memoria principale viene generata un'interruzione per *page o segmentation fault*
  - salti e riferimenti a porzioni diverse di dati sono molto frequenti
- Se la memoria principale è piena e ci sono molti processi attivi, ogni volta che c'è un memory fault il SO deve gestire lo swap di processi

Problemi legati alla memoria virtuale

- Il rischio è incorrere nel fenomeno del **trashing**: il SO impiega la maggior parte del suo tempo a swappare pezzi di processi, anzichè eseguire istruzioni
- Per evitarlo, o almeno minimizzarlo, il SO cerca di indovinare quali pezzi di processo saranno usati con minore o maggiore probabilità nell'immediato futuro
  - ovvero, quale sarà la prossima istruzione da eseguire o i prossimi dati richiesti
- Questo tentativo di *previsione* avviene sulla base della storia recente

Principio di Località

- A tale scopo si usa il **principio di località**
- I riferimenti che un processo fa tendono ad essere vicini
  - sia che si tratti di dati che di istruzioni
- Quindi solo poche porzioni di processo saranno necessarie di volta in volta
- Quindi si può prevedere abbastanza bene quali pezzi di processo saranno necessari nel prossimo futuro
- In conclusione la memoria virtuale può funzionare bene, e in effetti lo fa

Pagine e Località: Esempio

Di volta in volta, i riferimenti sono confinati ad un sottoinsieme delle pagine