

# Sistemi Operativi

AAF - Secondo anno - 3CFU

A.A. 2022/2023

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)

- Assumiamo che:
  - la memoria venga partizionata in parti piccole 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
- Ogni pagina, per essere usata, deve essere collocata in un frame
  - una pagina può essere messa in un *qualunque* frame
  - pagine contigue possono essere messe in frame distanti

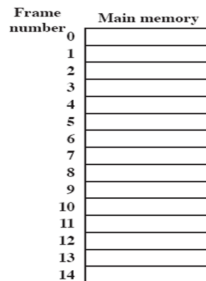
# Paginazione

- I SO che adottano la paginazione mantengono una **tabella delle pagine** per ogni processo
- Per ogni pagina del processo, questa tabella specifica in quale **frame effettivo** si trova
- 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 ricaricata

# 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

Frame number	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

# 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

Frame number	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	



# 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
- Non ci sono però 5 frame liberi contigui

Frame number	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	
5	
6	
7	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

# Paginazione

## Esempio

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

Frame number	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	D.0
5	D.1
6	D.2
7	C.0
8	C.1
9	C.2
10	C.3
11	D.3
12	D.4
13	
14	

## Paginazione: Esempio

Tabelle delle pagine per i processi attivi (non suspended)

0	0
1	1
2	2
3	3

Process A  
page table

0	—
1	—
2	—

Process B  
page table

0	7
1	8
2	9
3	10

Process C  
page table

0	4
1	5
2	6
3	11
4	12

Process D  
page table

13
14

Free frame  
list

# 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)

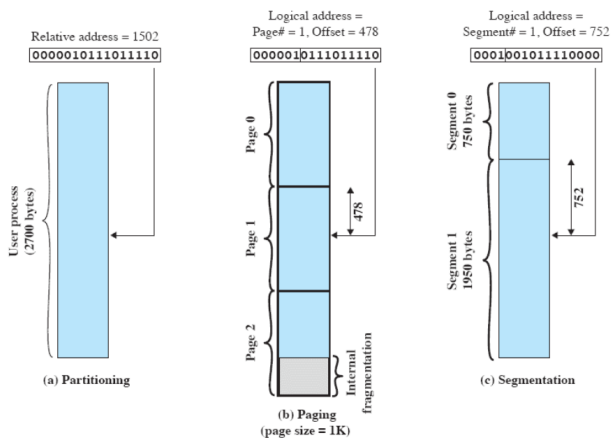
Gestione della memoria

# Segmentazione

# Segmentazione (Semplice)

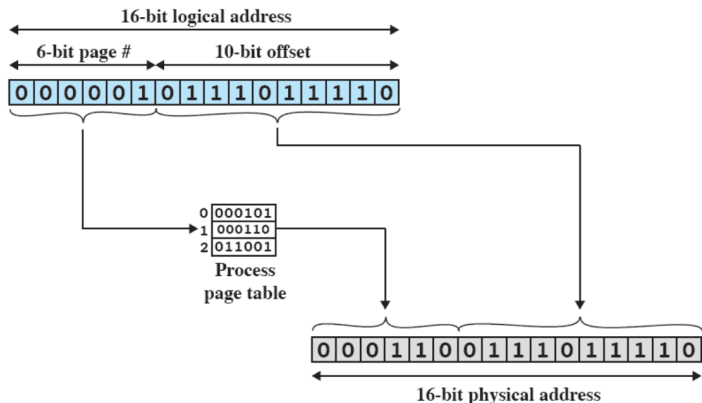
- I programmi vengono divisi in **segmenti**:
  - di dimensione (lunghezza) variabile
  - con un limite massimo alla dimensione
- Simile al partizionamento dinamico
  - ma con una differenza fondamentale: il programmatore o il compilatore devono gestire esplicitamente la segmentazione
  - cioè dire quanti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

## Indirizzi Logici



# Paginazione

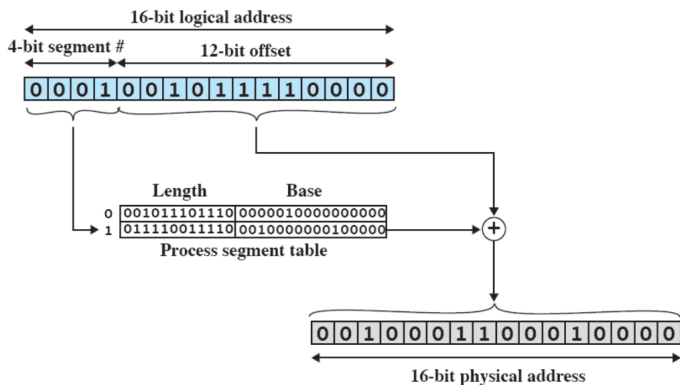
Per ogni processo, il numero di pagine è al più il numero di frames  
(non sarà così con la memoria virtuale)





# 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