UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

Sistemi Operativi

Gestione della Memoria

LEZIONE 21

prof. Antonino Staiano

Corso di Laurea in Informatica – Università di Napoli Parthenope

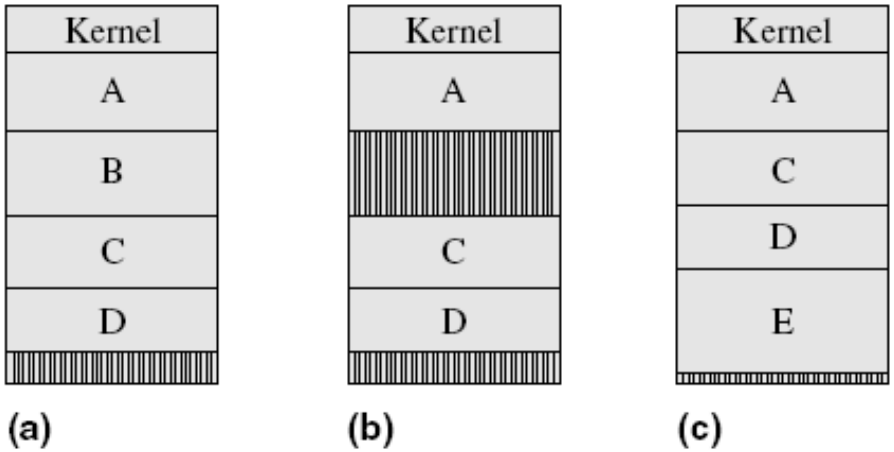
antonino.staiano@uniparthenope.it

Introduzione

- Allocazione contigua della memoria
- Allocazione non contigua della memoria
- Paginazione
- Segmentazione

Allocazione contigua della memoria

- Ad ogni processo è allocata una singola area contigua in memoria
- Affronta il problema della frammentazione della memoria
  - Applica tecniche di compattazione e riuso della memoria
    - La compattazione richiede un registro di rilocazione
    - Se manca tale registro, anche lo swapping diviene un problema



Allocazione non contigua della memoria

- Parti dello spazio di indirizzamento di un processo sono distribuite tra aree diverse di memoria
  - Riduce la frammentazione esterna

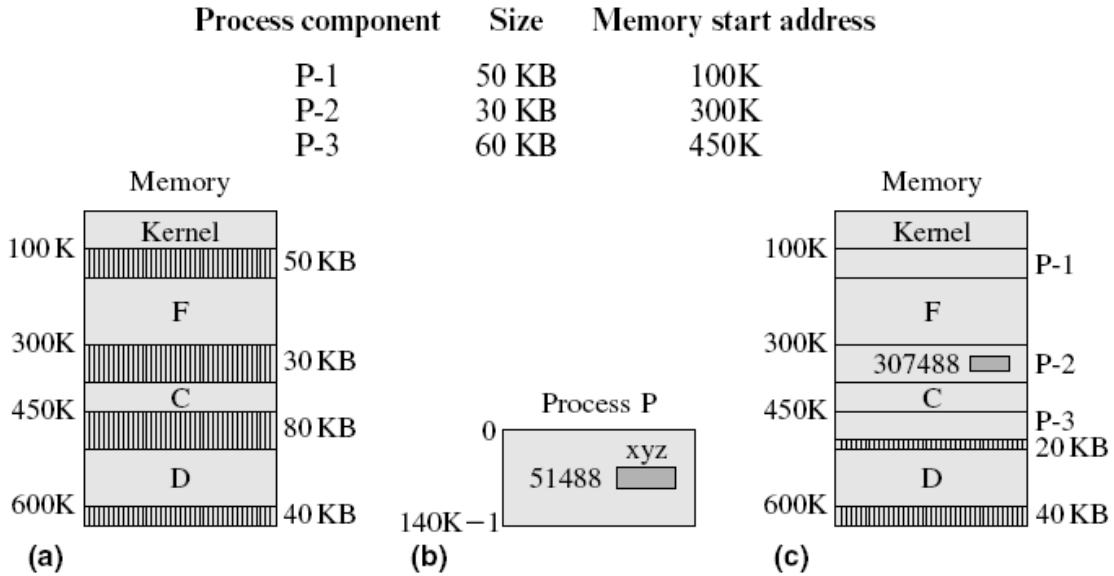
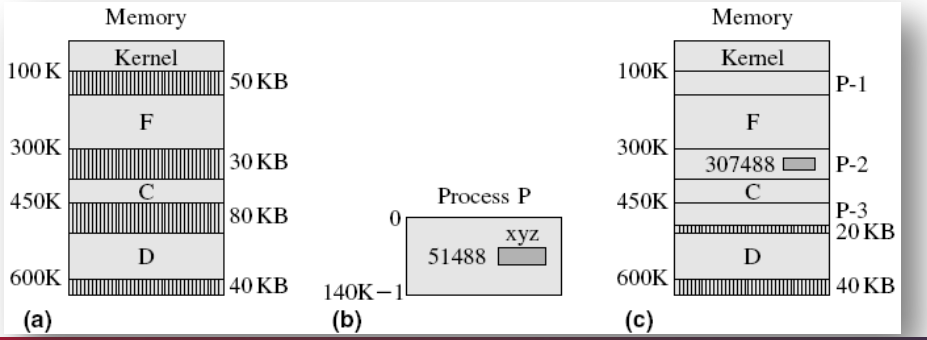


Figure 11.17 Noncontiguous memory allocation to process P.

Indirizzi Logici e Fisici, Traduzione indirizzi

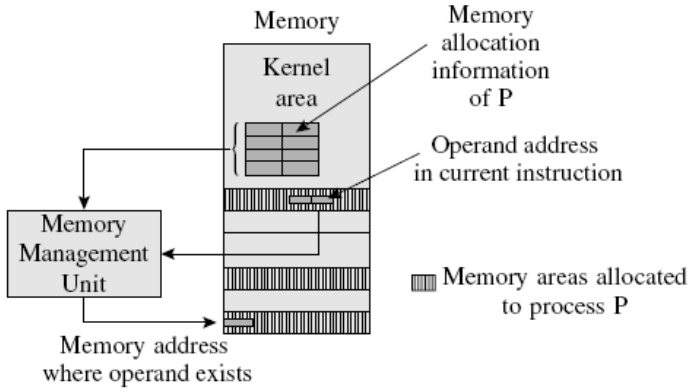
- Un *indirizzo logico* è l'indirizzo di un'istruzione o dato usato in un processo P
  - L'insieme degli degli indirizzi logici in P costituiscono il suo *spazio di indirizzamento logico*
- Un indirizzo fisico è l'indirizzo di memoria dove è memorizzata un'istruzione o un dato
  - L'insieme di indirizzi fisici nel sistema costituiscono il suo *spazio di indirizzamento fisico*



- P-1 ha dimensione 50KB ->51200 byte
  - xyz si trova in P-2
  - #Byte di P-2: 51488-51200=288
- P-2 è caricato a partire da 300KB (307200)
  - indirizzo fisico di xyz è 307488

Indirizzi Logici e Fisici, Traduzione indirizzi(cont.)

- Il kernel memorizza le informazioni sulle aree di memoria allocate al processo P
  - In una tabella disponibile alla MMU
- La CPU invia l'indirizzo logico di ogni dato o istruzione usato in P alla MMU
- La MMU usa le info sull'allocazione della tabella per calcolare l'indirizzo fisico
  - *indirizzo di memoria effettivo* del dato o dell'istruzione
- La procedura per determinare l'indirizzo di memoria effettivo è chiamata *traduzione dell'indirizzo*

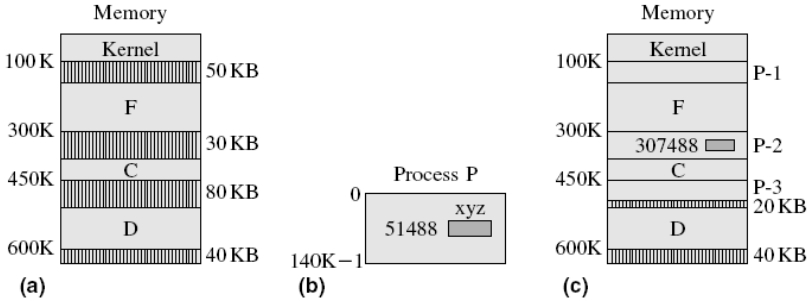


A schematic of address translation in noncontiguous memory allocation.

Indirizzi Logici e Fisici, Traduzione indirizzi(cont.)

- Un indirizzo logico consiste di due parti: **id componente di P** che contiene l'indirizzo e **id del byte** in tale componente di P
  - Visto come una coppia ($comp_i$, $byte_i$)
- L'indirizzo fisico è determinato dalla seguente equazione:

Effective memory address of $(comp_i, byte_i)$
$=$ start address of memory area allocated to $comp_i$
$+$ byte number of $byte_i$ within $comp_i$



- Esempio: P fa riferimento all'area dati di xyz con l'indirizzo logico **(P-2, 288)**
  - La MMU calcola l'indirizzo fisico effettivo come 307.200 + 288 = 307.488

Approcci allocazione non contigua della memoria

- Due approcci
  - Paginazione
    - I processi consistono di componenti di dimensioni fisse chiamate *pagine*
    - Elimina la frammentazione esterna
    - La dimensione della pagina è definita dallo hardware
  - Segmentazione
    - Il programmatore identifica le entità logiche in un programma (un insieme di funzioni, strutture dati, oggetti); ognuna è chiamata *segmento*
    - Semplifica la condivisione di codice, dati e moduli di programma tra i processi
    - I segmenti hanno dimensioni diverse quindi il kernel usa tecniche di riuso
      - possibile la frammentazione esterna
  - Approccio ibrido: **segmentazione con paginazione**
    - Facilita la condivisione di codice, dati e moduli tra processi
    - Evita la frammentazione esterna
    - Possibile la frammentazione interna come nella paginazione

Allocazione contigua vs Allocazione non contigua

Table 11.4 Comparison of Contiguous and Noncontiguous Memory Allocation

Function	Contiguous allocation	Noncontiguous allocation
Memory allocation	The kernel allocates a single memory area to a process.	The kernel allocates several memory areas to a process—each memory area holds one component of the process.
Address translation	Address translation is not required.	Address translation is performed by the MMU during program execution.
Memory fragmentation	External fragmentation arises if first-fit, best-fit, or next-fit allocation is used. Internal fragmentation arises if memory allocation is performed in blocks of a few standard sizes.	In paging, external fragmentation does not occur but internal fragmentation can occur. In segmentation, external fragmentation occurs, but internal fragmentation does not occur.
Swapping	Unless the computer system provides a relocation register, a swapped-in process must be placed in its originally allocated area.	Components of a swapped-in process can be placed anywhere in memory.

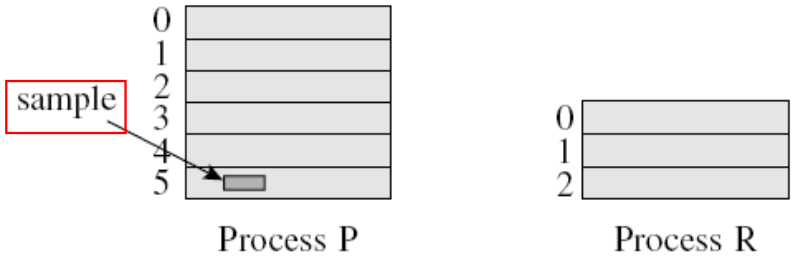
Protezione della memoria

- Le aree di memoria allocate ad un programma devono essere protette dalle interferenza con altri programmi
  - La MMU esegue questo compito con il controllo dei limiti
    - Mentre esegue la traduzione per un indirizzo logico ($comp_i$, $byte_i$), controlla se $comp_i$ si trova nel programma e se $byte_i$ si trova in $comp_i$
      - E' generato un interrupt di violazione della protezione se il controllo fallisce
- Il controllo dei limiti può essere semplificato con la paginazione
  - $byte_i$ non può superare la dimensione di una pagina

Paginazione

- Nella vista logica, lo spazio degli indirizzi di un processo consiste di un'organizzazione lineare delle pagine
- Ogni pagina contiene **s** byte, con s potenza di 2
  - Il valore di s è specificato nell'architettura del sistema
- I processi usano indirizzi logici numerici
  - La MMU scompone l'indirizzo logico nella coppia (p_i, b_i), con $p_i >= 0$ e $0 <= b_i < s$

Esempio:
Dimensione pagine 1KB
Processo **P** dimensione 5500 byte
Processo **R** 2500 byte
`sample` ha indirizzo 5248 = 5 x 1024 + 128
MMU vede l'indirizzo come la coppia (5, 128)

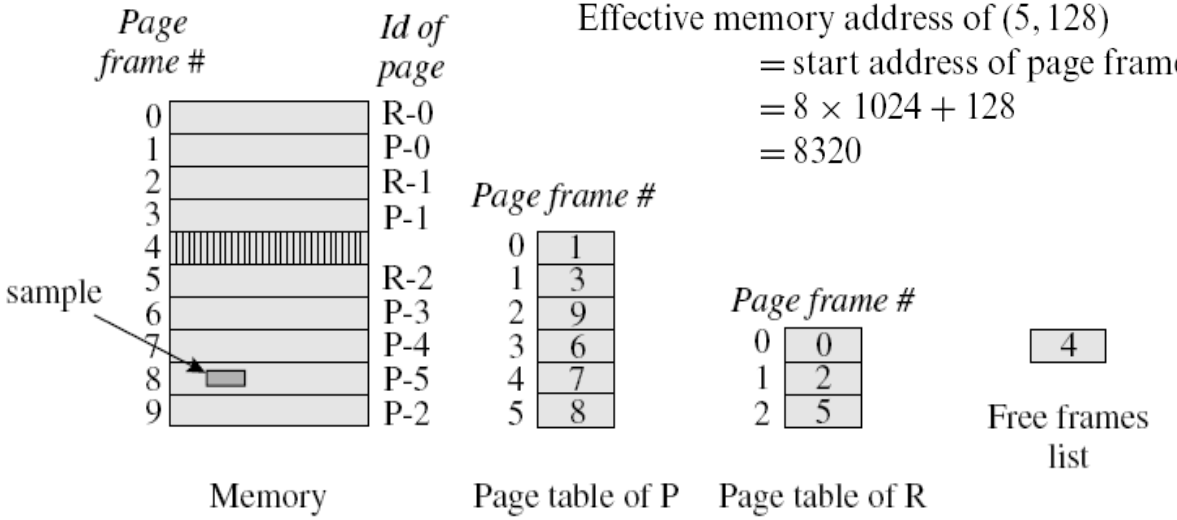


Logical view of processes in paging.

Paginazione (cont.)

- La memoria è divisa in aree chiamate *frame*
  - Numerati a partire da 0
- Un **frame** ha la stessa dimensione di una **pagina**
- Il kernel mantiene una lista, chiamata *lista dei frame liberi*, per tenere traccia dei numeri di frame liberi
- Quando carica un processo, il kernel consulta la lista dei frame liberi e alloca un frame libero a ogni pagina del processo
- Per facilitare la traduzione degli indirizzi, il kernel costruisce una tabella delle pagine (PT) per ciascun processo
  - La PT ha un'entrata per ogni pagina del processo che indica il frame allocato alla pagina
  - Mentre esegue la traduzione per un indirizzo logico (p_i,b_i), la MMU usa il numero di pagina p_i per
    - indicizzare la PT del processo
    - ottenere il numero di frame allocato a p_i
    - calcolare l'indirizzo di memoria effettivo

Paginazione (cont.)



Ogni frame dimensione di **1KB**
Memoria **10KB -> frame 0-9**
sample ha indirizzo logico **(5, 128)**

Paginazione (cont.)

- Notazione per la traduzione degli indirizzi

  s dimensione di una pagina

  l_l lunghezza di un indirizzo logico (cioè, numero di bit)

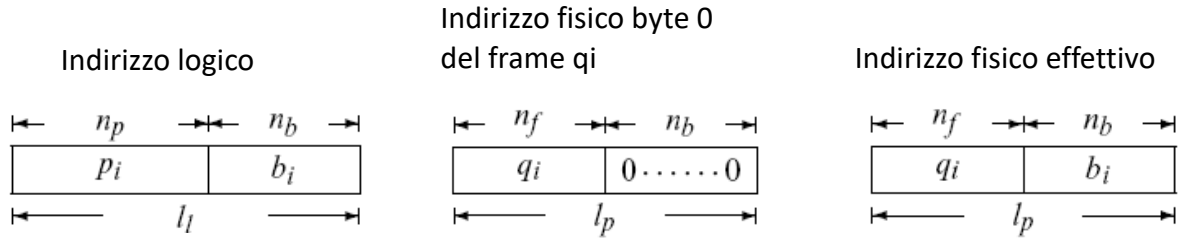
  l_p lunghezza di un indirizzo fisico

  n_b numero di bit usato per rappresentare il numero del byte in un indirizzo logico

  n_p numero di bit usato per rappresentare il numero di pagina in un indirizzo logico

  n_f numero di bit usato per rappresentare il numero di frame in un indirizzo fisico

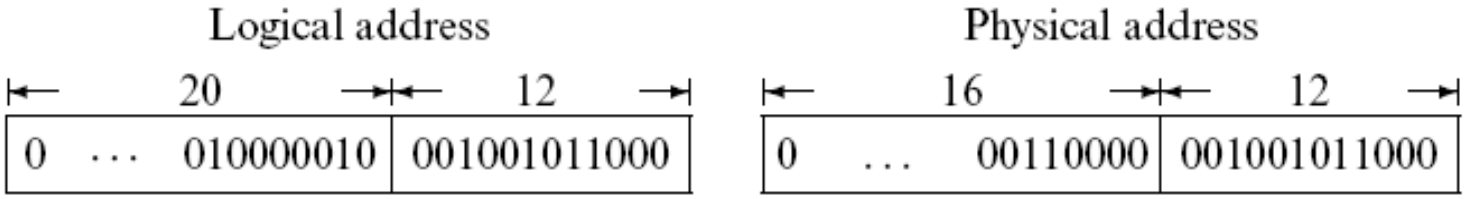
  - La dimensione di una pagina, s, è una potenza di 2
    - n_b è scelto in modo che $s = 2^{nb}$



- La MMU ottiene i valori di pi e bi semplicemente concatenando i bit di pi e bi
- L'indirizzo di memoria fisico si ottiene concatenando i bit di qi e bi

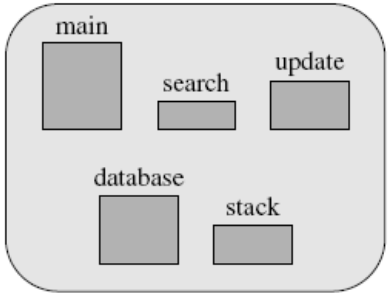
Esempio: traduzione di indirizzo nella paginazione

- Indirizzi logici a 32 bit
- Dimensione pagina 4KB
  - 12 bit sono adeguati per indirizzare i byte in una pagina
    - 2^{12} = 4KB
- Per una memoria di dimensioni di 256 MB, $l_p = 28$
- Se alla pagina 130 è allocato il frame 48,
  - $p_i = 130$ e $q_i = 48$
  - Se $b_i = 600$, gli indirizzi logici e fisici sono:



Segmentazione

- Un segmento è un'entità logica in un programma
  - Esempio, una funzione, una struttura dati o un oggetto



Process Q

Name	*Size*	*Address*
main	476	23500
database	20240	32012
search	378	76248
update	642	91376
stack	500	54500

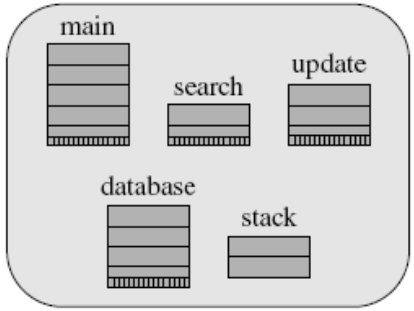
Segment table of Q

Figure 11.21 A process Q in segmentation.

- Ogni indirizzo logico usato in Q ha la forma (s_i, b_i)
  - s_i e b_i sono id di un segmento e un byte nel segmento
- Esempio:
  - Se in **update** c'è l'istruzione **get_sample** al byte 232
  - (*update*,*get_sample*) ha indirizzo di memoria effettivo: 91376 + 232 = 91608
- L'allocazione di memoria ai segmenti è contigua

Segmentazione con paginazione

- Ogni segmento in un programma è paginato separatamente
- Ad ogni segmento è allocato un numero intero di pagine
- Semplifica l'allocazione della memoria e la velocizza
- Evita la frammentazione esterna



Process Q

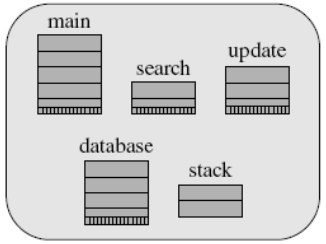
Name	*Size*	*Page table address*
main	476	
database	20240	
search	378	
update	642	
stack	500	

Segment table of Q

Figure 11.22 A process Q in segmentation with paging.

Segmentazione con paginazione (cont.)

- Per ogni segmento è costruita una tabella delle pagine
- L'indirizzo della tabella delle pagine è mantenuto nell'entrata del segmento della tabella dei segmenti
- La traduzione di un indirizzo logico (s_i,b_i) è fatta in due passi:
  1. Dall'entrata di s_i nella *tabella dei segmenti* è determinato l'indirizzo della tabella delle pagine
  2. Il numero del byte b_i è diviso nella coppia (ps_i, bp_i) dove ps_i è il numero di pagina nel segmento s_i e bp_i è il numero del byte nella pagina p_i. Il calcolo dell'indirizzo effettivo è fatto come nella paginazione



Process Q

Name	*Size*	*Page table address*
main	476	
database	20240	
search	378	
update	642	
stack	500	

Segment table of Q

Figure 11.22 A process Q in segmentation with paging.

Riepilogando

- La CPU ha un *registro di rilocazione* per semplificare la rilocazione
- L'allocazione / deallocazione della memoria può portare alla frammentazione: interna ed esterna
  - Le strategie first-fit, next-fit e best-fit cercano di ridurre la frammentazione interna
  - Gli allocatori buddy system e potenze di 2 eliminano la frammentazione esterna
  - L'allocazione non contigua riduce la frammentazione esterna
    - Richiede l'uso di un'unità di gestione della memoria (MMU) da parte della CPU