

# Capitolo 4

## Gestione della Memoria

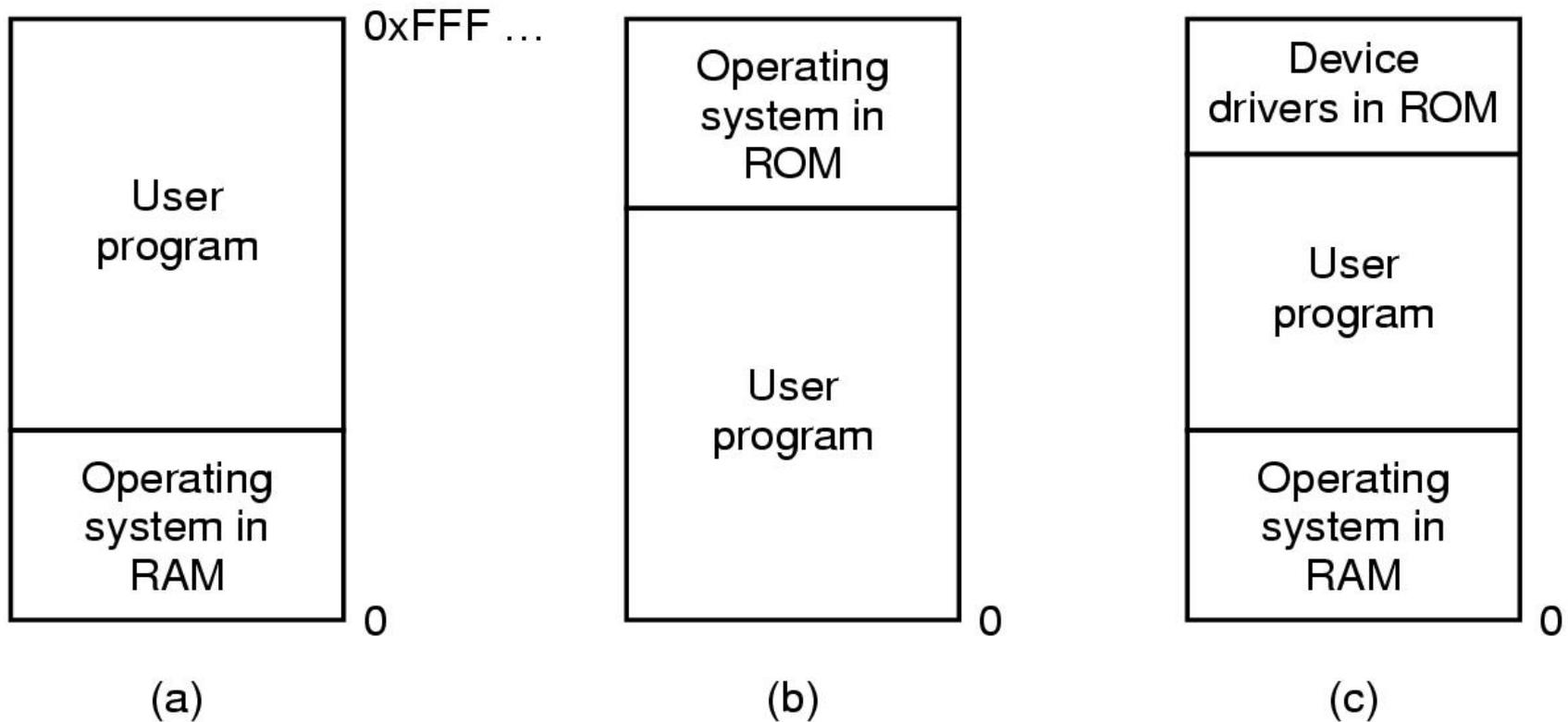
- 4.1 Introduzione alla gestione della memoria
- 4.2 Swapping
- 4.3 Memoria virtuale
- 4.4 Implementazione
- 4.5 Algoritmi di sostituzione
- 4.6 Criteri di progetto per la paginazione
- 4.7 Case study: Unix
- 4.8 Case study: Windows 2000

# Gestione della Memoria

- Idealmente la memoria dovrebbe essere
  - grande
  - veloce
  - non volatile
- Gerarchia di memorie
  - Cache: piccola, veloce e costosa
  - Memoria principale: mediamente veloce e costosa
  - Disco: lento, capiente ed economico
- Gestire la memoria significa gestire la gerarchia di memorie

# Introduzione alla Gestione della Memoria

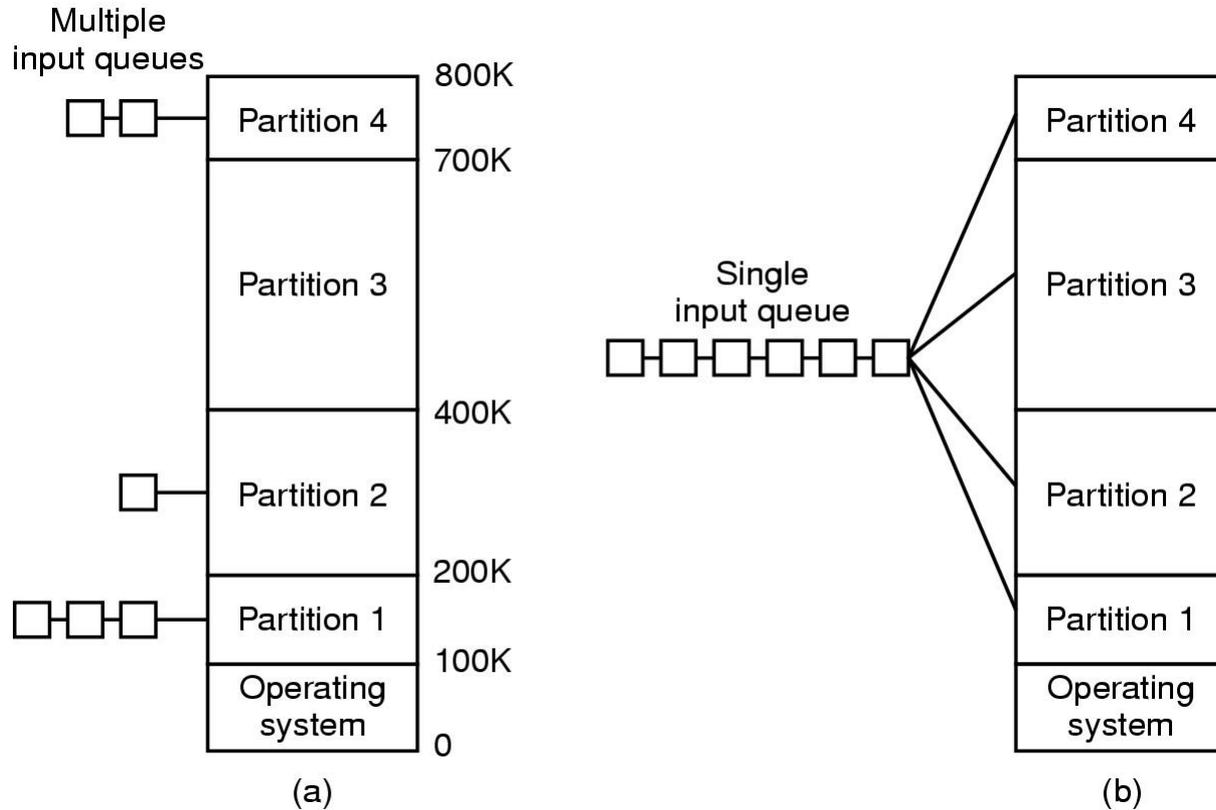
Ambiente monoprogrammato senza swapping nè paginazione



Tre modi semplici di organizzare la memoria

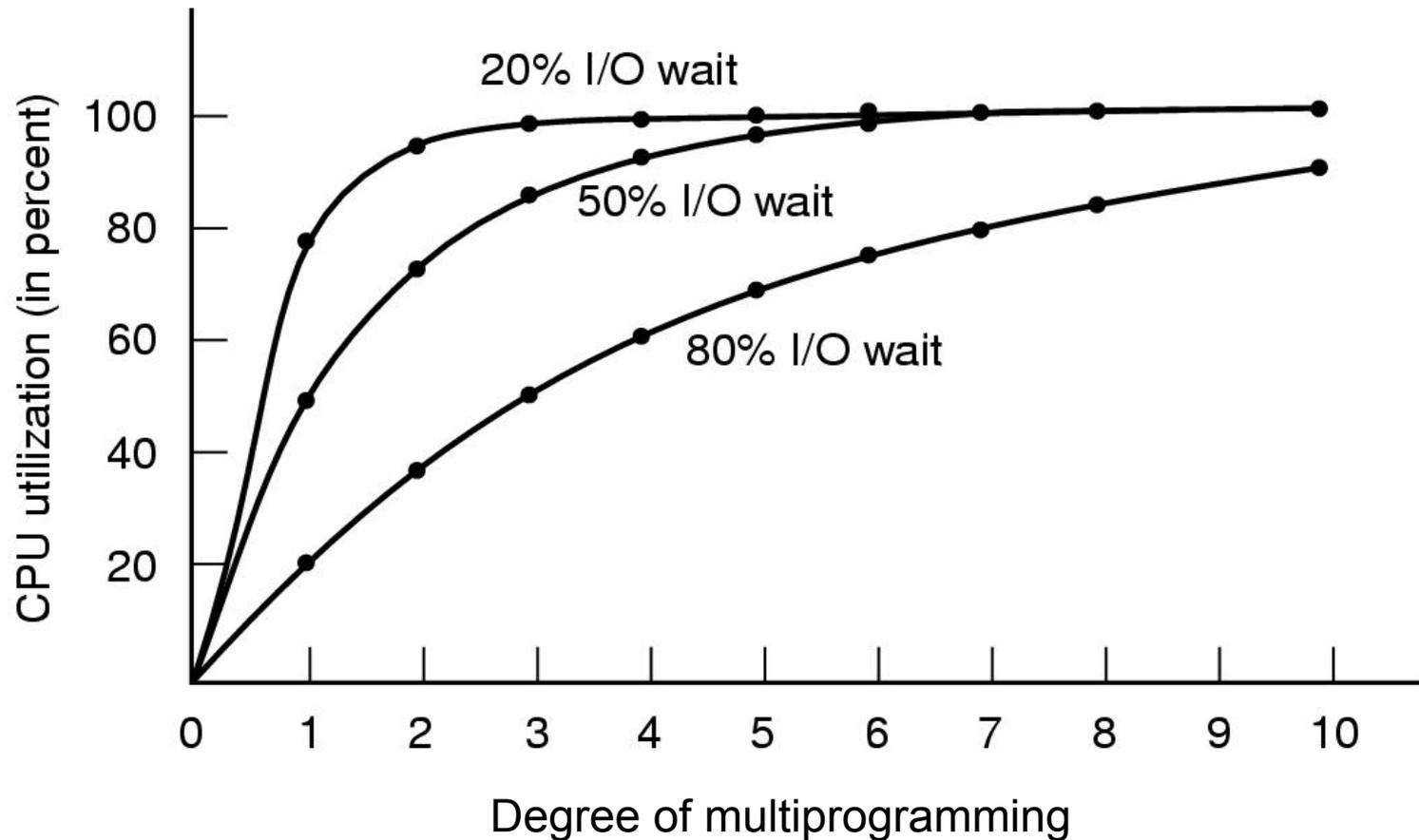
- un sistema operativo con un solo processo utente

# Ambiente multiprogrammato con partizioni fisse



- Partizioni fisse
  - Con code dei job distinte per ogni partizione
  - Con unica coda dei lavori

# Modelli per la Multiprogrammazione

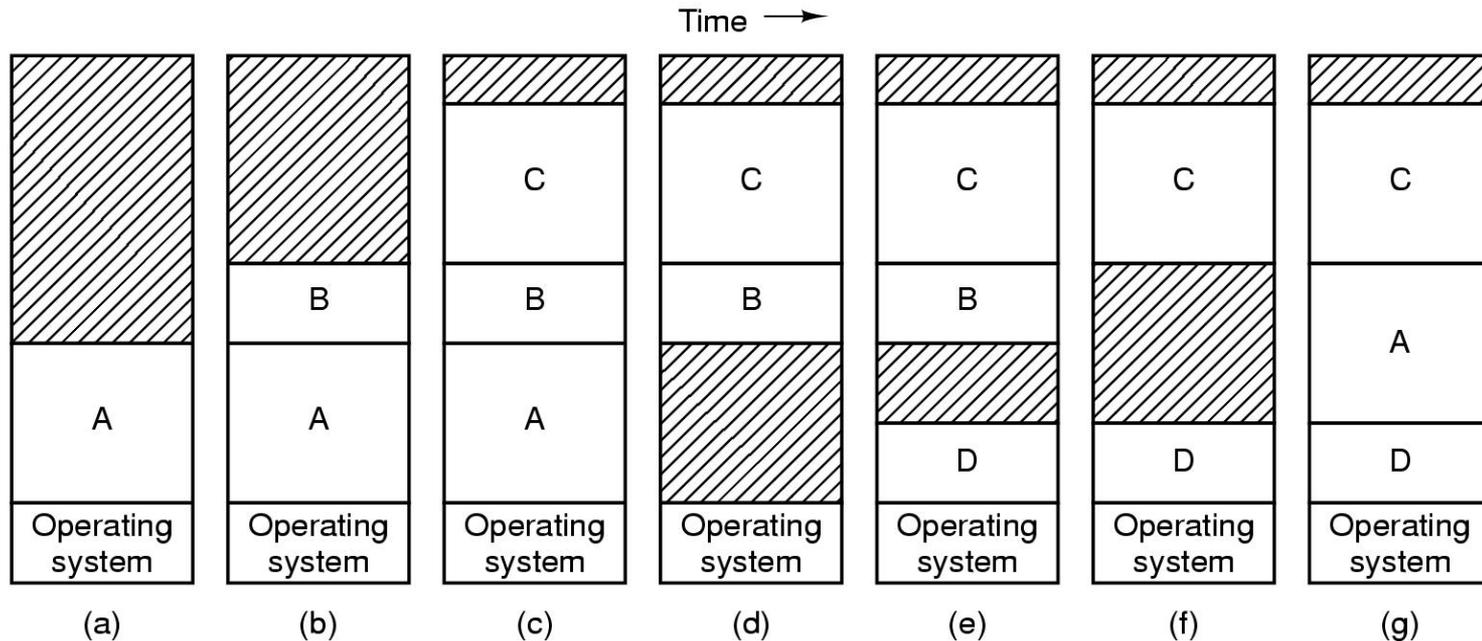


Utilizzazione della CPU in funzione del numero di processi in memoria

# Rilocazione e Protezione

- Non si conosce l'istante nel quale il programma verrà caricato in memoria
  - Gli indirizzi dei dati e del codice non possono essere assoluti
  - E' necessario mantenere separati gli spazi dei processi
- Registri Base e Limite
  - *Indirizzo fisico = indirizzo logico + registro base*
  - *Errore se indirizzo logico > registro limite*

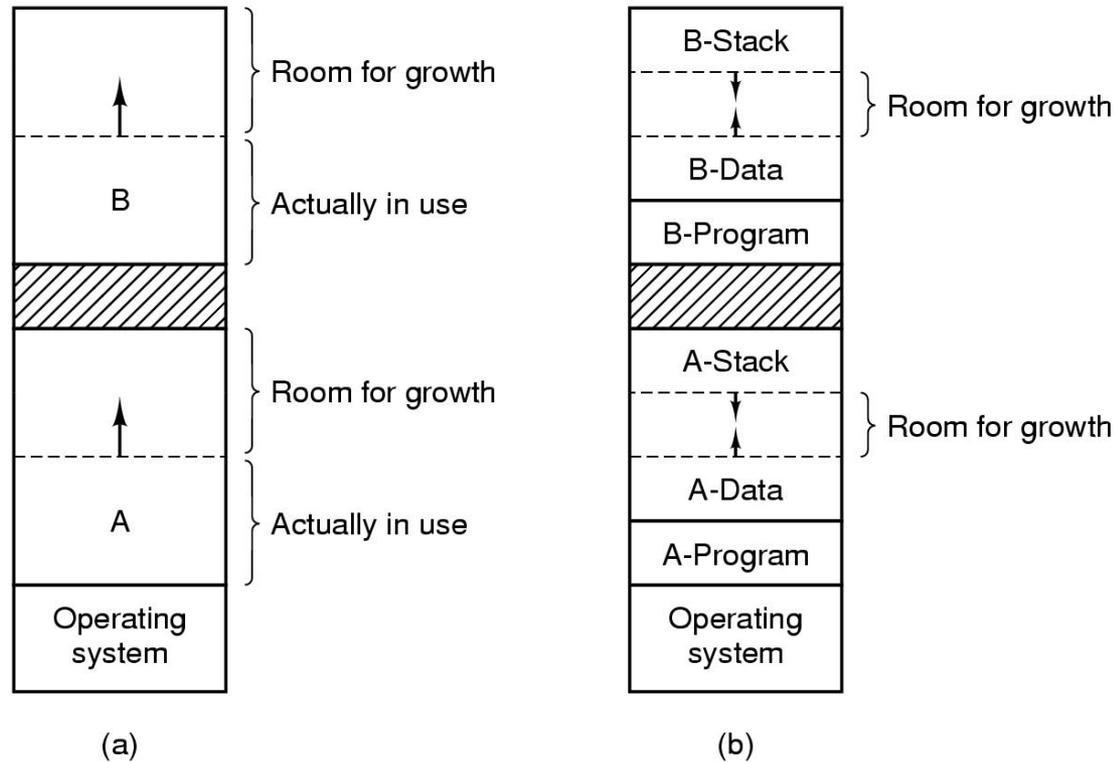
# Swapping (1)



L'allocazione della memoria cambia quando:

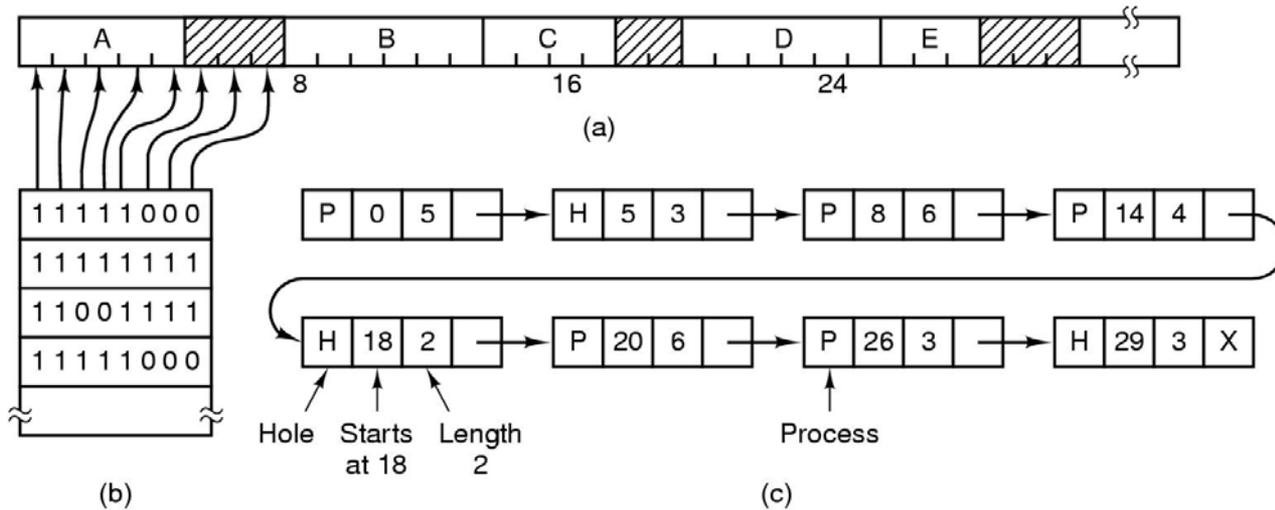
- I processi vengono caricati in memoria
- I processi rilasciano la memoria

# Swapping (2)



- a) Allocare dinamicamente memoria nel segmento dati
- b) Allocare dinamicamente memoria nello stack e nel segmento dati

# Gestione della Memoria con Bit Map



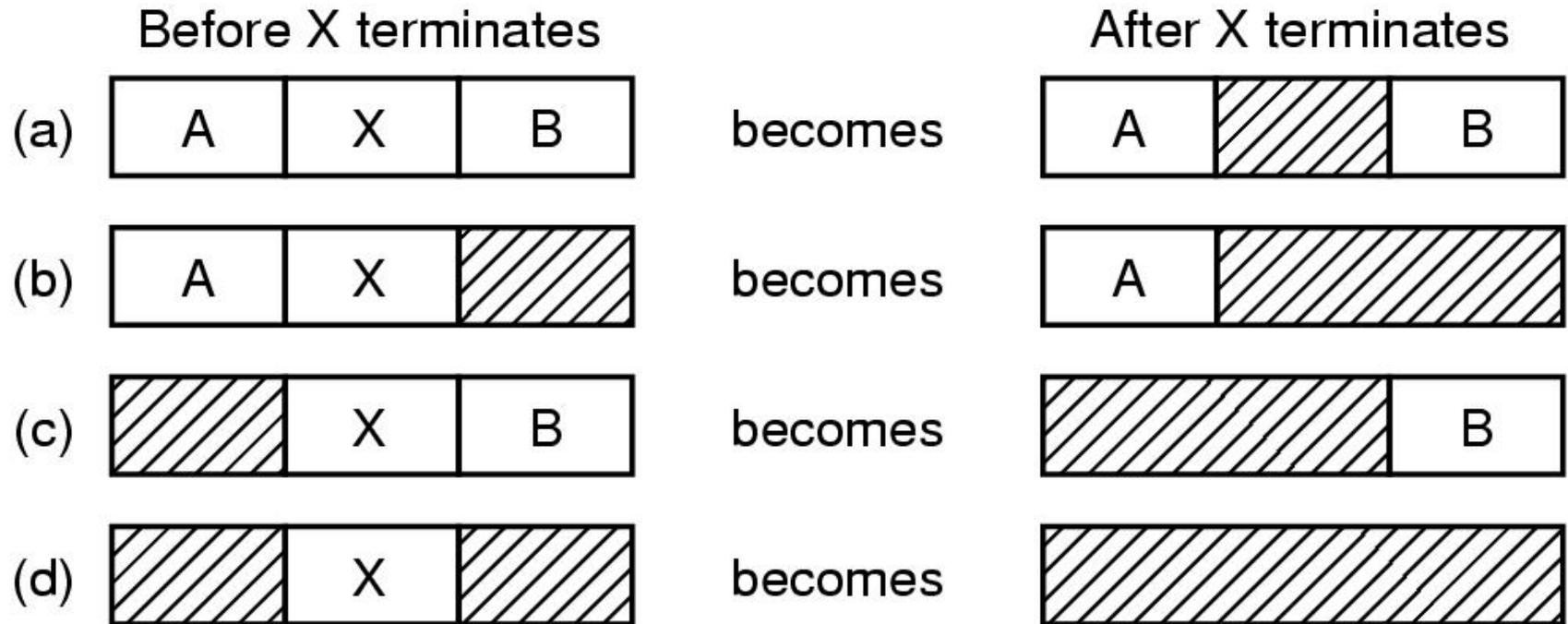
a) Porzione di memoria con 5 processi e 3 aree libere

- Le suddivisioni indicano l'unità di allocazione

b) Bit Map corrispondente

c) Lista di allocazione corrispondente

# Gestione della Memoria con Liste



Quattro possibili combinazioni di allocazione in seguito alla terminazione del processo X

# Allocazione della Memoria

Allocazione di un blocco di memoria di dimensione  $x$ :

- **First Fit**

- Individua la prima porzione di memoria libera di dimensione  $\geq x$

- **Best Fit**

- Individua la più piccola porzione di memoria libera di dimensione  $\geq x$

