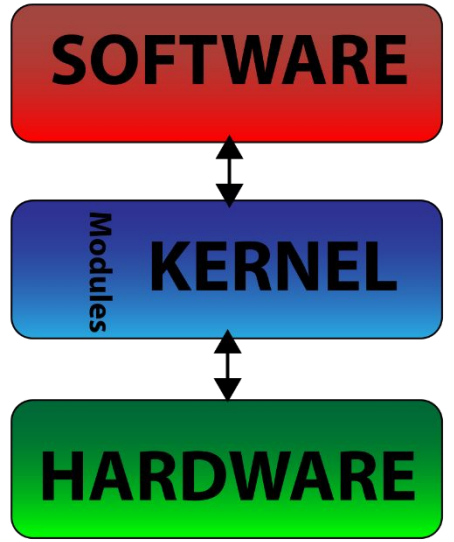
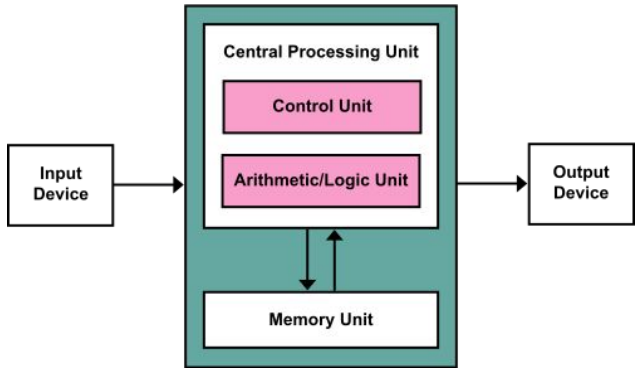
UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Operativi

Memoria centrale

Docente:
Domenico Daniele Bloisi

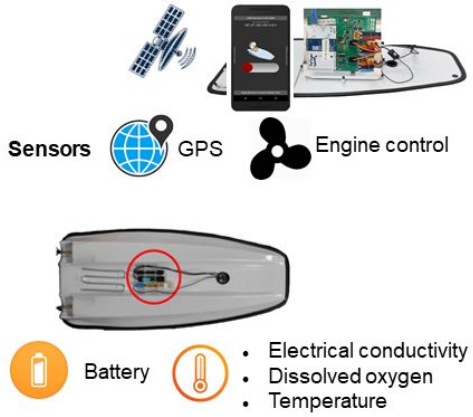


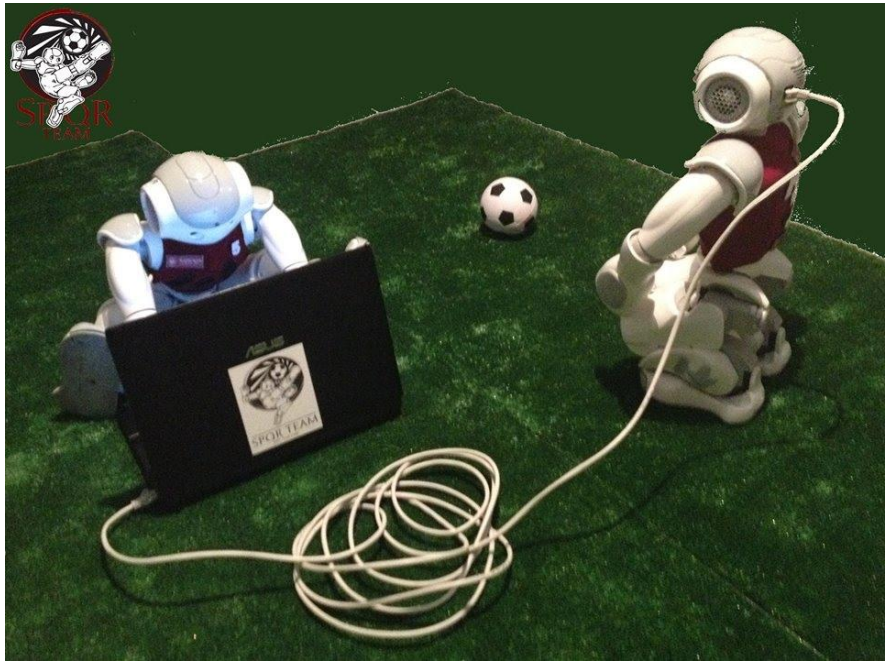


Domenico Daniele Bloisi

- Professore Associato
  Dipartimento di Matematica, Informatica ed Economia
  Università degli studi della Basilicata
  http://web.unibas.it/bloisi

- SPQR Robot Soccer Team
  Dipartimento di Informatica, Automatica e Gestionale Università degli studi di Roma “La Sapienza”
  http://spqr.diag.uniroma1.it



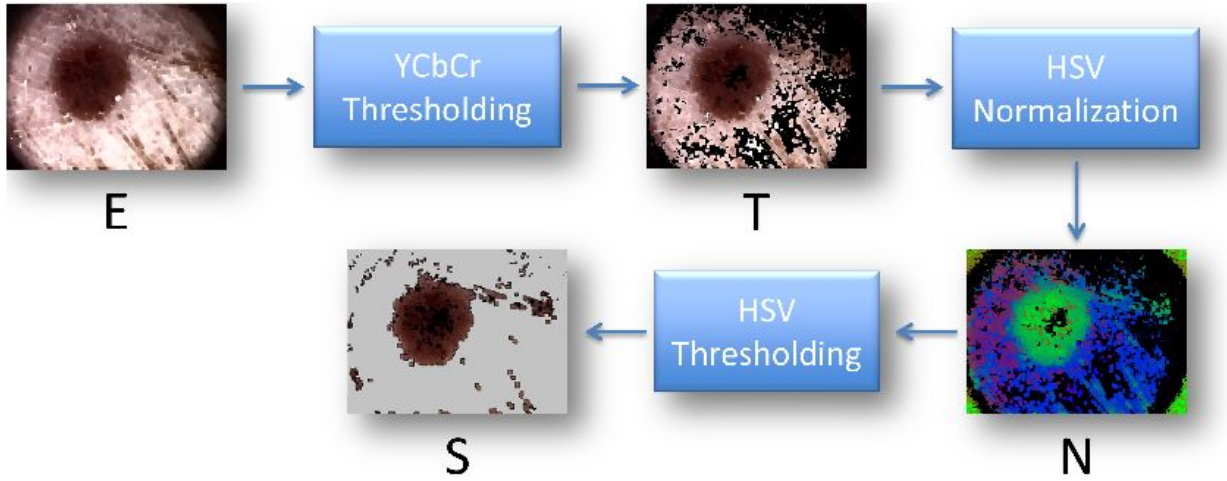


Interessi di ricerca

- Intelligent surveillance
- Robot vision
- Medical image analysis

https://youtu.be/2KHNZX7UIWQ

https://youtu.be/9a70Ucgbi_U



UNIBAS Wolves https://sites.google.com/unibas.it/wolves

- UNIBAS WOLVES is the robot soccer team of the University of Basilicata. Established in 2019, it is focussed on developing software for NAO soccer robots participating in RoboCup competitions.

- UNIBAS WOLVES team is twinned with SPQR Team at Sapienza University of Rome

https://youtu.be/ji0OmkaWh20

Informazioni sul corso

- Home page del corso:
  http://web.unibas.it/bloisi/corsi/sistemi-operativi.html
- Docente: Domenico Daniele Bloisi
- Periodo: I semestre ottobre 2022 – gennaio 2023

  - Lunedì dalle 15:00 alle 17:00 (Aula Leonardo)
  - Martedì dalle 08:30 alle 10:30 (Aula 1)

Ricevimento

- In presenza, durante il periodo delle lezioni:
  Lunedì dalle 17:00 alle 18:00
  presso Edificio 3D, II piano, stanza 15
  Si invitano gli studenti a controllare regolarmente la bacheca degli avvisi per eventuali variazioni
- Tramite google Meet e al di fuori del periodo delle lezioni:
  da concordare con il docente tramite email

Per prenotare un appuntamento inviare una email a
domenico.bloisi@unibas.it

Programma – Sistemi Operativi

- Introduzione ai sistemi operativi
- Gestione dei processi
- Sincronizzazione dei processi
- Gestione della memoria centrale
- Gestione della memoria di massa
- File system
- Sicurezza e protezione

Esecuzione di un programma

Durante l'esecuzione, i programmi e i dati cui essi accedono devono trovarsi, almeno parzialmente, in memoria centrale.

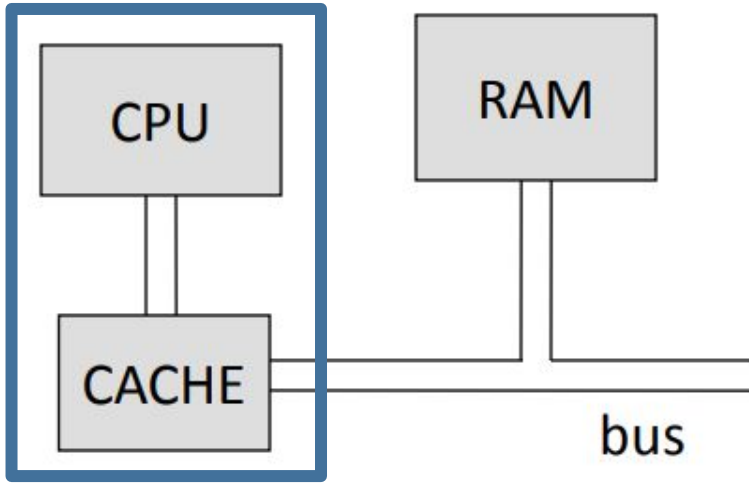
La scelta di un metodo di gestione della memoria, per un sistema specifico, dipende da molteplici fattori, in particolar modo dall'*architettura hardware*.

Memoria centrale

- ROM (Read-Only Memory - memoria a sola lettura) è una memoria non volatile in grado di mantenere memorizzati i dati anche in assenza di alimentazione elettrica

- RAM (Random-Access Memory - memoria ad accesso casuale) è una memoria volatile, cioè essa perde il proprio contenuto allo spegnimento del computer

- CACHE (dal termine francese caché che significa “nascosto”) un tipo di memoria volatile come la RAM, ma molto più veloce e più costosa di quest’ultima



https://www.unirc.it/documentazione/materiale_didattico/597_2012_322_14551.pdf

Spazio di memoria separato

- Ciascun processo deve avere uno spazio di memoria separato, in modo da proteggere i processi l'uno dall'altro.

- Questo è fondamentale per avere più processi caricati in memoria per l'esecuzione concorrente.

Spazio di memoria protetto

- Bisogna proteggere il sistema operativo dall'accesso dei processi utenti e, in sistemi multiutente, salvaguardare i processi utenti uno dall'altro.

- Tale protezione è implementata a livello hardware poiché il sistema operativo (per questioni di prestazioni) non interviene negli accessi della CPU alla memoria.

Registro base e registro limite

Si può implementare il meccanismo di protezione tramite due registri, detti **registro base** e **registro limite**

Registro base contiene il più piccolo indirizzo legale della memoria fisica (per es. 300040)

Registro limite determina la dimensione dell'intervallo ammesso (per es. 120900)

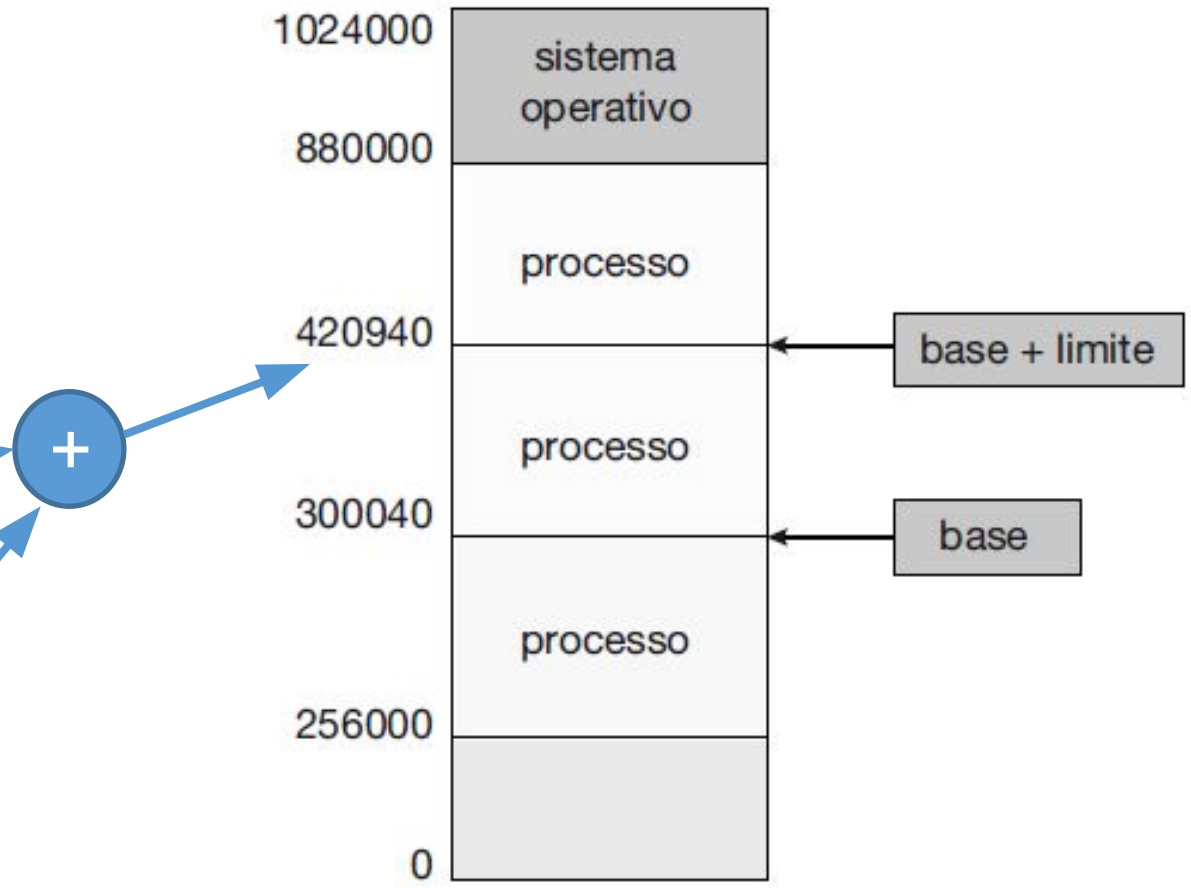


Figura 9.1 I registri base e limite definiscono lo spazio degli indirizzi logici.

Protezione delle aree di memoria

Qualsiasi tentativo da parte di un programma eseguito in modalità utente di accedere alle aree di memoria riservate al sistema operativo o a una qualsiasi area di memoria riservata ad altri utenti comporta l'invio di una **eccezione** (*trap*) che restituisce il controllo al sistema operativo che, a sua volta, interpreta l'evento come un errore fatale.

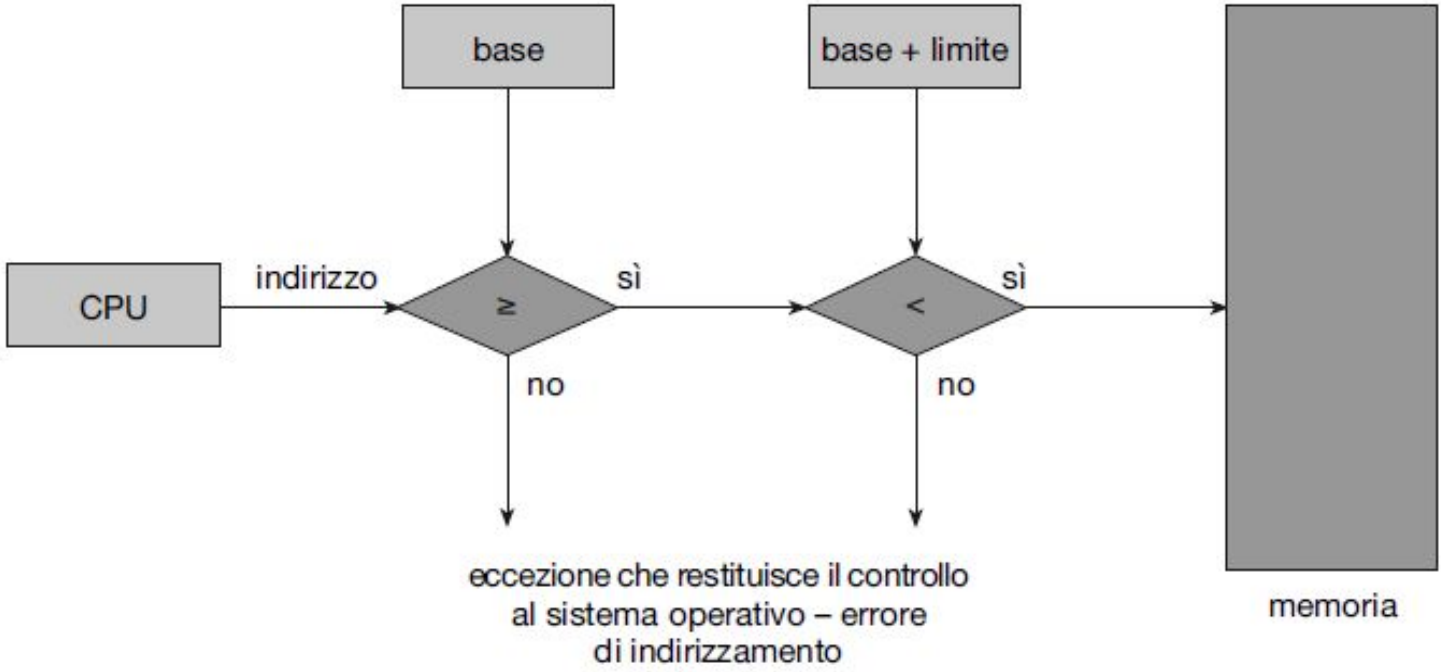


Figura 9.2 Protezione hardware degli indirizzi tramite registri base e limite.

Associazione degli indirizzi

Nella maggior parte dei casi un programma utente, prima di essere eseguito, deve passare attraverso *varie fasi*, alcune delle quali possono essere facoltative

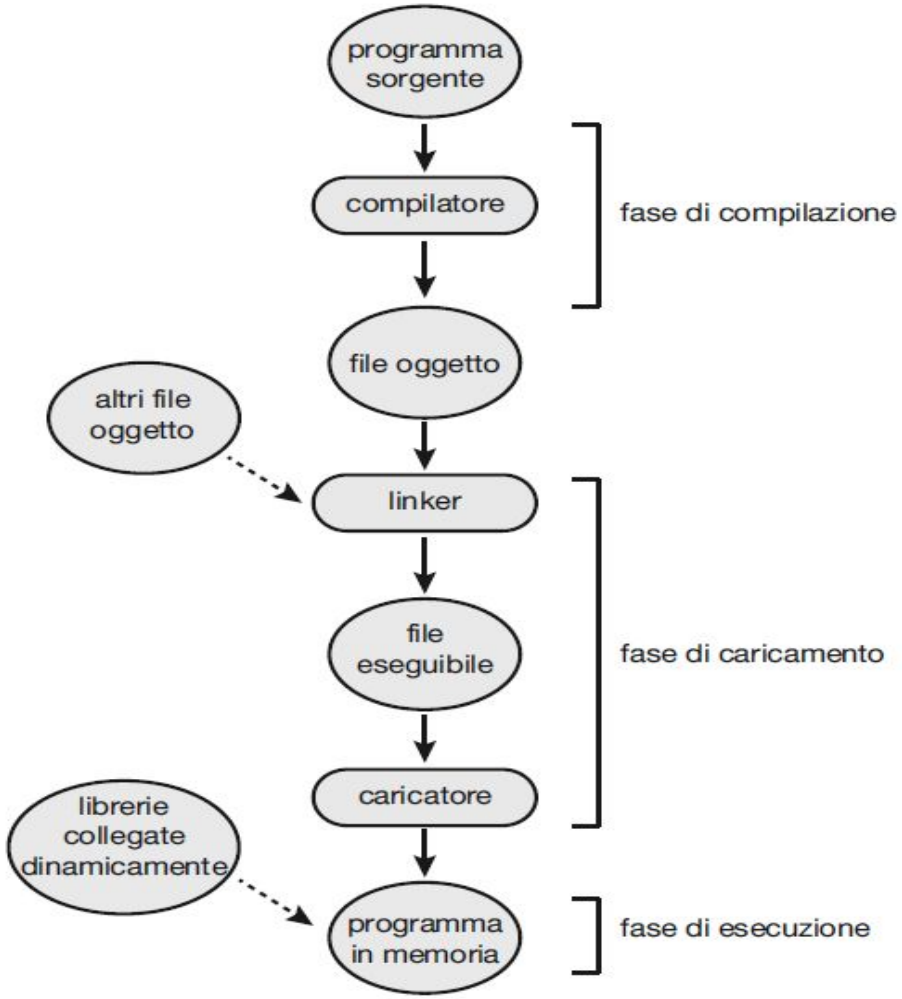


Figura 9.3 Fasi di elaborazione di un programma utente.

Associazione degli indirizzi

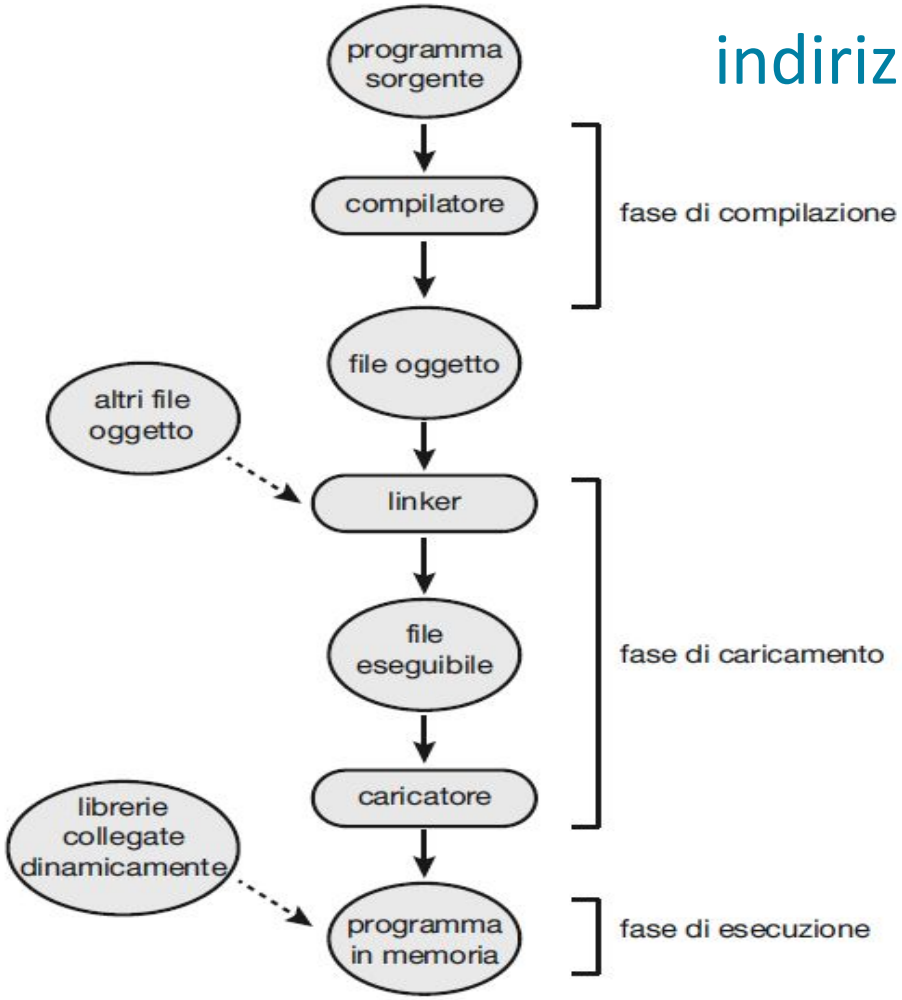


Figura 9.3 Fasi di elaborazione di un programma utente.

indirizzi simbolici (es. variabile `count`)

binding

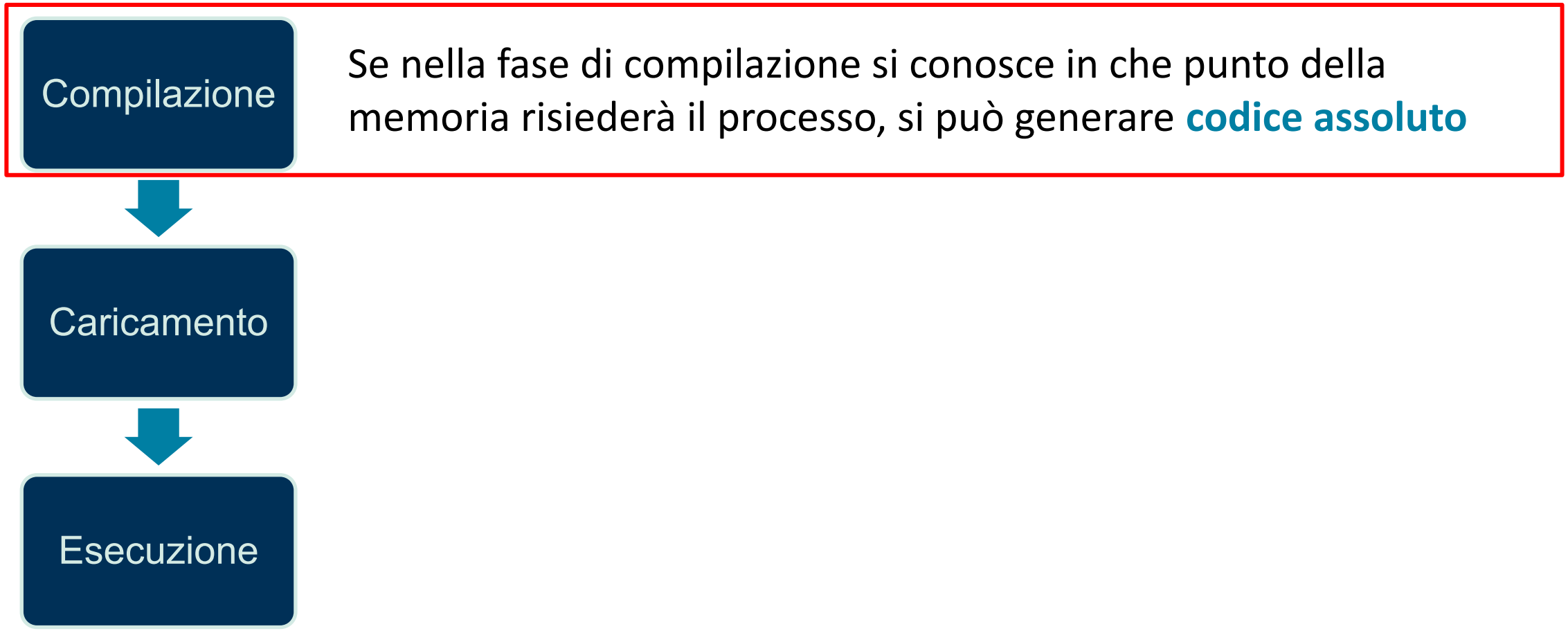
indirizzi rilocabili

indirizzi assoluti

Associazione degli indirizzi

Generalmente, l'associazione di istruzioni e dati a indirizzi di memoria si può compiere in qualsiasi *fase* del seguente percorso.

Codice assoluto



Se nella fase di compilazione si conosce in che punto della memoria risiederà il processo, si può generare **codice assoluto**

Codice rilocabile

Se nella fase di compilazione non è possibile conoscere in che punto della memoria risiederà il processo, il compilatore deve generare **codice rilocabile**

Associazione a runtime

Se durante l'esecuzione il processo può essere spostato da un segmento di memoria ad un altro, si deve **ritardare** l'associazione degli indirizzi fino alla fase di esecuzione

Spazio di indirizzi

indirizzo logico → indirizzo generato dalla CPU

indirizzo fisico → indirizzo caricato nel **registro dell'indirizzo di memoria**

indirizzi virtuali → perché con il metodo di associazione nella fase di esecuzione gli indirizzi logici non coincidono con gli indirizzi fisici

Unità di gestione della memoria

L'**unità di gestione della memoria** (*memory management unit, **MMU***) svolge l'associazione nella fase d'esecuzione dagli indirizzi virtuali agli indirizzi fisici.

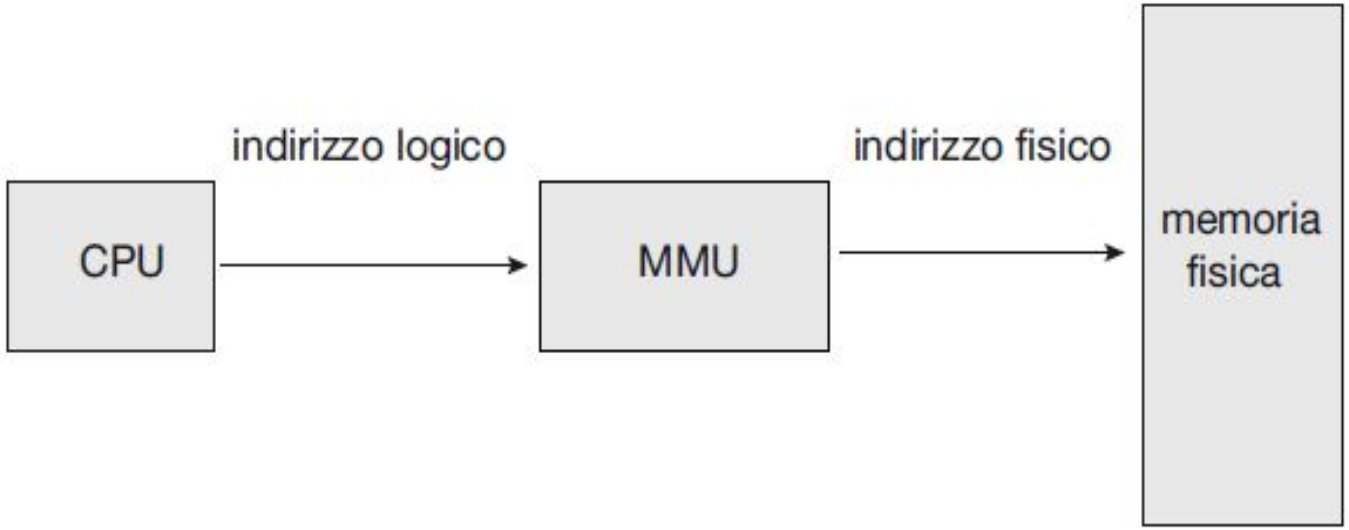


Figura 9.4 Unità di gestione della memoria (MMU).

Registro di rilocazione

Quando un processo utente genera un indirizzo, prima dell'invio all'unità di memoria, si somma a tale indirizzo il valore contenuto nel registro di rilocazione.

il registro di base è ora denominato **registro di rilocazione**

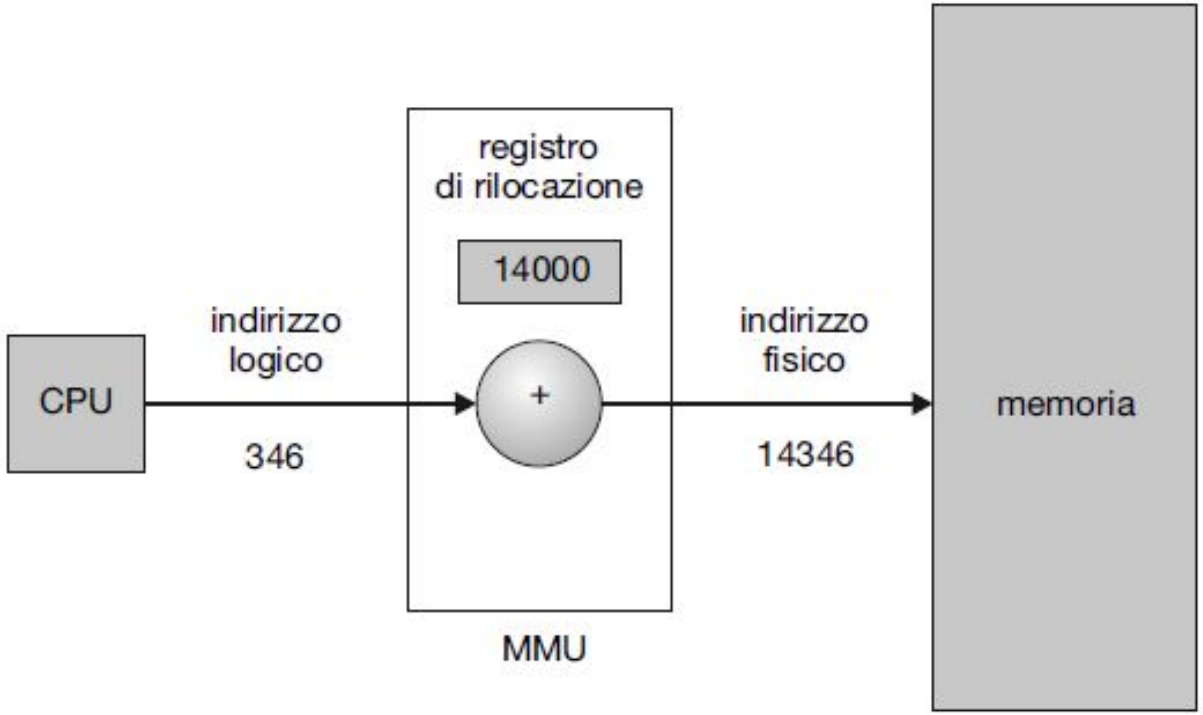


Figura 9.5 Rilocazione dinamica tramite un registro di rilocazione.

Allocazione contigua della memoria

Con l'**allocazione contigua della memoria** ciascun processo è contenuto in una singola sezione di memoria contigua a quella che contiene il processo successivo.

Protezione della memoria

Ogni indirizzo logico deve cadere nell'intervallo specificato dal registro limite

La **MMU** fa corrispondere *dinamicamente* l'indirizzo fisico all'indirizzo logico sommando a quest'ultimo il valore contenuto nel registro di rilocazione (Figura 9.6) e invia l'indirizzo risultante alla memoria

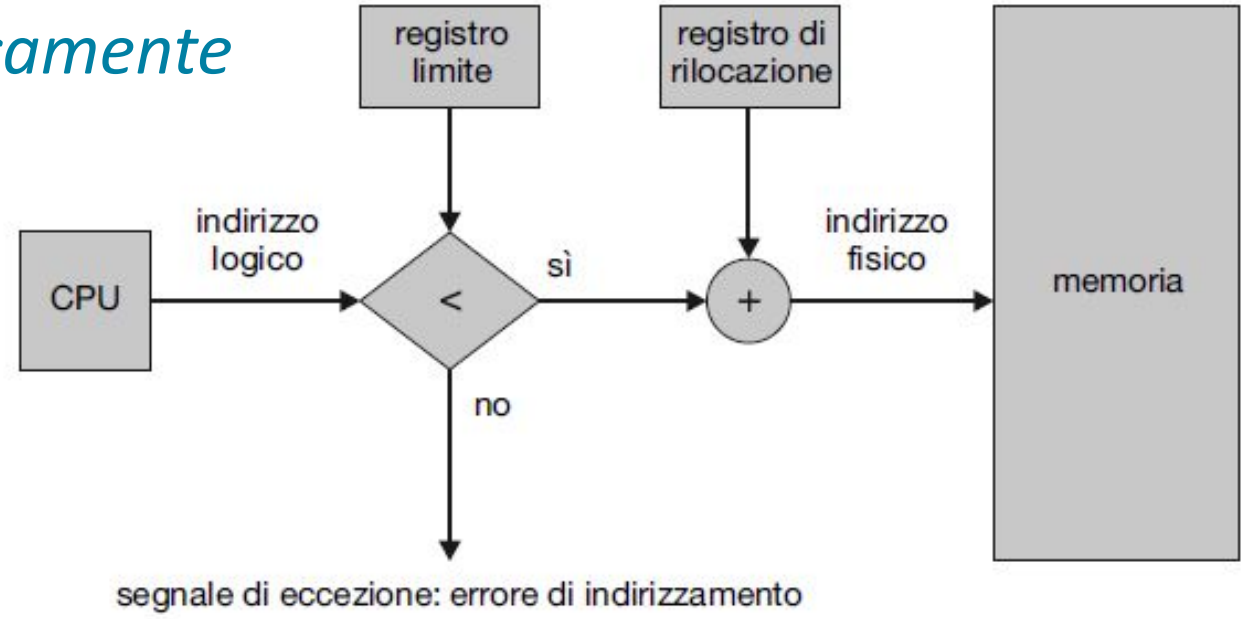


Figura 9.6 Registri di rilocazione e limite.

Vantaggi del registro di rilocazione

Lo schema con registro di rilocazione consente al sistema operativo di cambiare dinamicamente le proprie dimensioni.

Tale flessibilità è utile in molte situazioni. Per esempio, se un driver di periferica non è attualmente in uso, può essere rimosso dalla memoria.

Metodi per l’allocazione della memoria

Uno dei metodi più semplici per l’allocazione della memoria consiste nel suddividere la stessa in **partizioni di dimensione variabile**, dove ciascuna partizione può contenere esattamente un processo.

Schema a partizione variabile

Nello **schema a partizione variabile** il sistema operativo conserva una tabella in cui sono indicate le partizioni di memoria disponibili e quelle occupate.

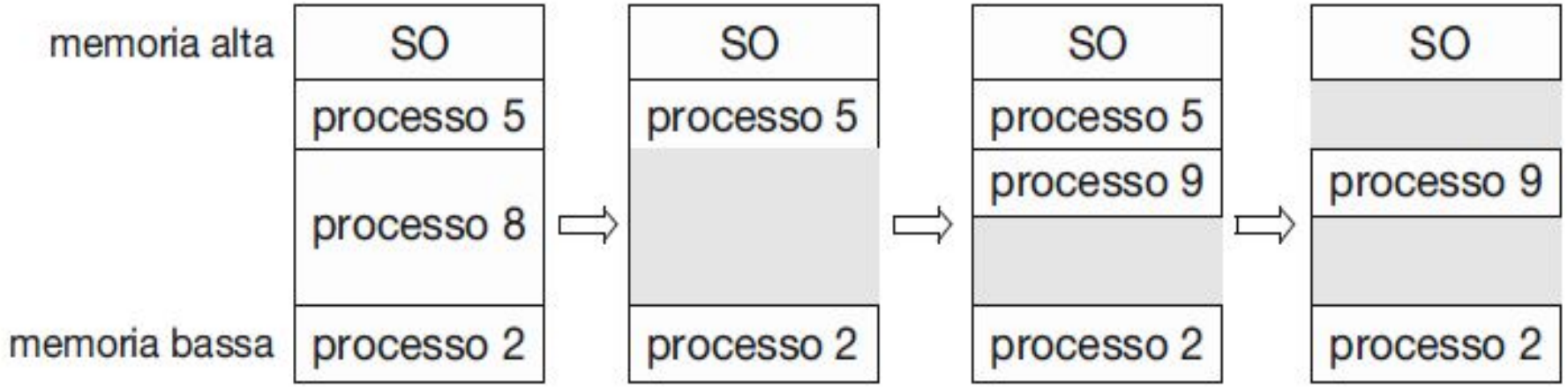


Figura 9.7 Schema a partizione variabile.

Allocazione dinamica

L'**allocazione dinamica della memoria** consente di soddisfare una richiesta di dimensione *n* data una lista di **buchi** liberi

I criteri più usati per scegliere un buco libero tra quelli disponibili nell'insieme sono i seguenti:

First-fit

Si assegna il primo buco abbastanza grande

Best-fit

Si assegna il più piccolo buco in grado di contenere il processo

Worst-fit

Si assegna il buco più grande

Problema della frammentazione

First-fit

Si assegna il primo buco abbastanza grande

Best-fit

Si assegna il più piccolo buco in grado di contenere il processo

Worst-fit

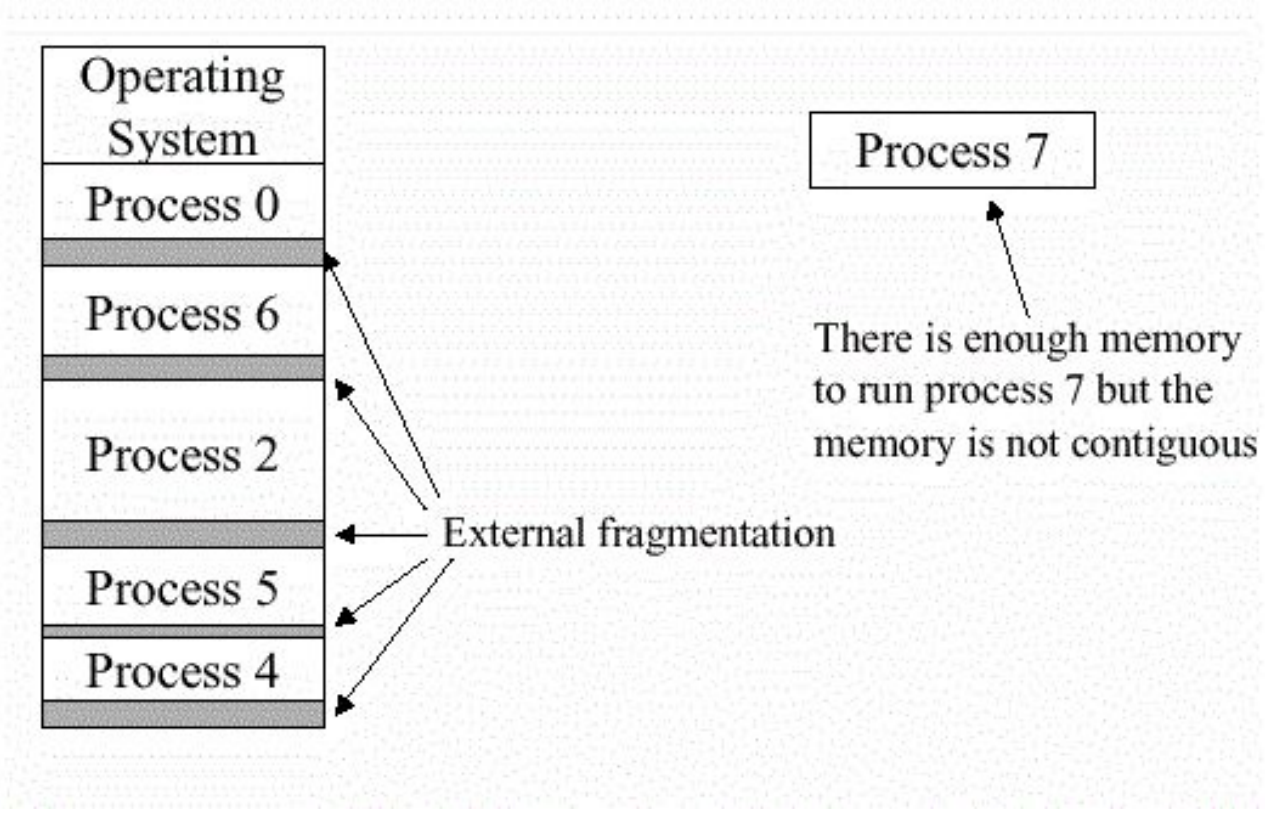
Si assegna il buco più grande

I criteri *first-fit* e *best-fit* di allocazione della memoria soffrono di **frammentazione esterna**

Frammentazione esterna

External fragmentation

Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous so it can not be used.



https://stackoverflow.com/questions/1200694/internal-and-external-fragmentation/43870017

Regola del 50%

L'analisi statistica dell'algoritmo first-fit rileva che per *n* blocchi assegnati si perdono altri *0,5n* blocchi a causa della frammentazione esterna.

Ciò significa che potrebbe essere inutilizzabile ben un terzo della memoria.

Questa caratteristica è nota come regola del 50%

Compattazione

Una soluzione al problema della frammentazione esterna è data dalla compattazione.

Si riordina il contenuto della memoria per riunire le porzioni di memoria libera in un unico grosso blocco.

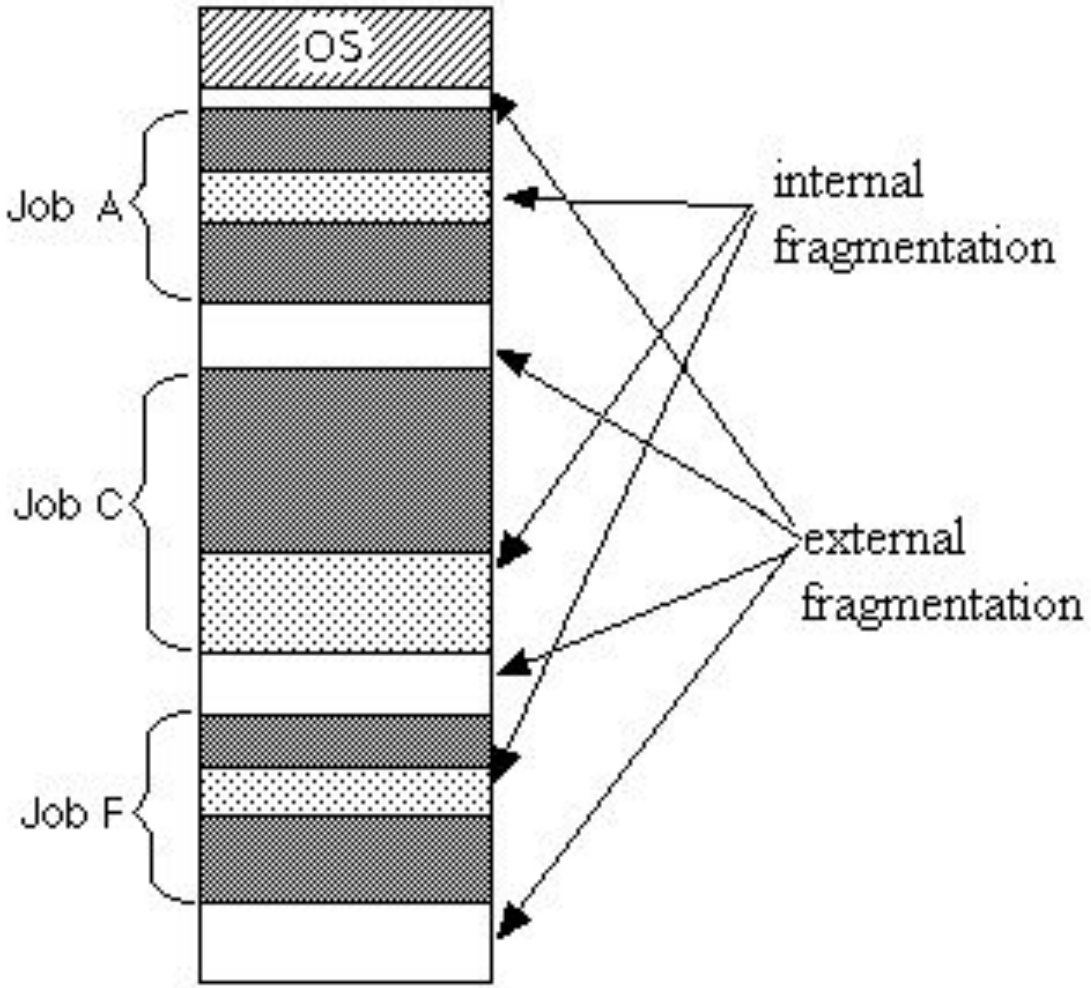
La compattazione è possibile solo se la rilocazione è dinamica e si effettua nella fase di esecuzione.

Frammentazione interna

Internal fragmentation

Memory block assigned to process is bigger.

Some portion of memory is left unused as it cannot be used by another process.



https://stackoverflow.com/questions/1200694/internal-and-external-fragmentation/43870017

Paginazione

Paginazione → schema di gestione della memoria che consente allo spazio di indirizzamento fisico di un processo di essere non contiguo

La paginazione è implementata sfruttando la cooperazione tra sistema operativo e hardware del computer

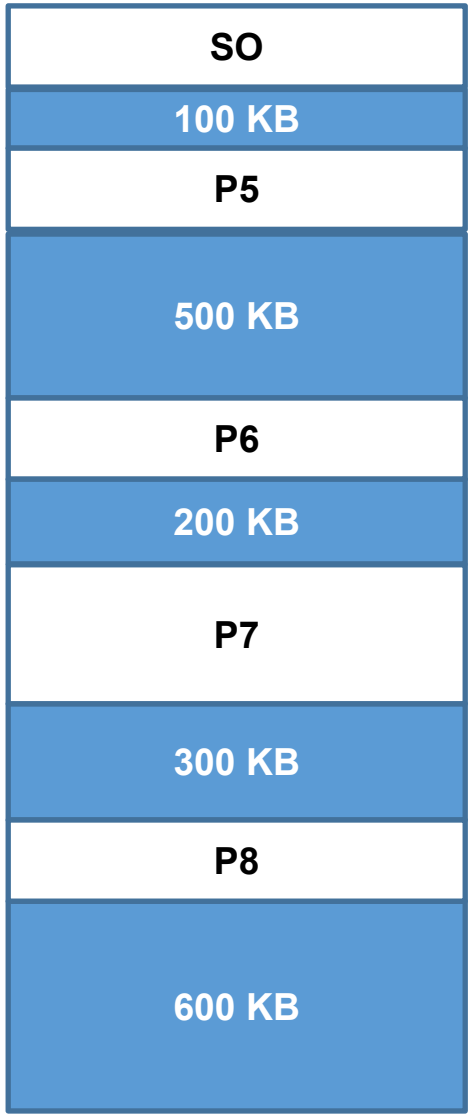
Esercizio 1

Si assuma di avere:

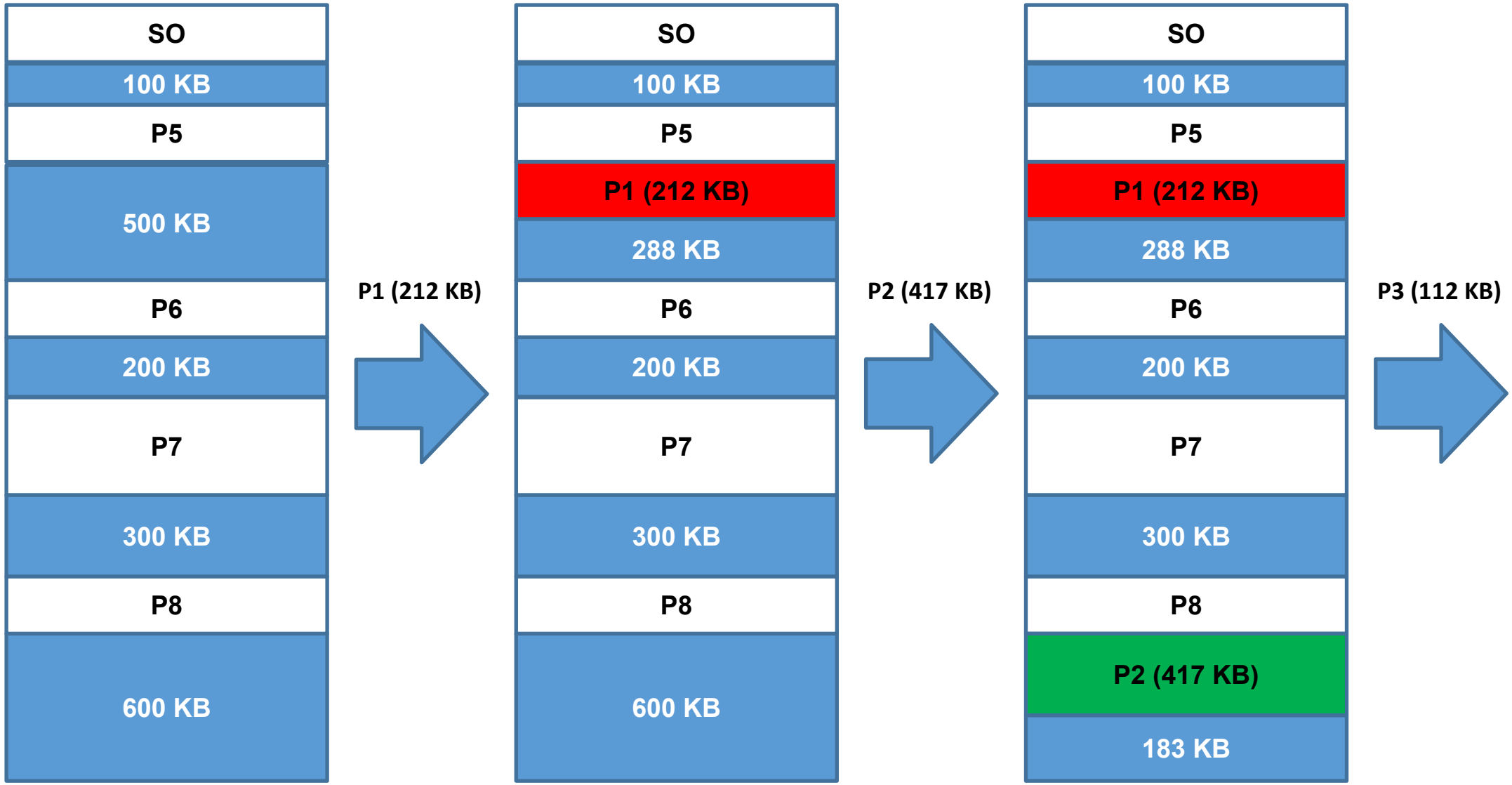
- un sistema di allocazione contigua dei processi in memoria
- la memoria nella situazione illustrata a lato

Utilizzando un approccio first-fit, come verranno allocati i processi seguenti?

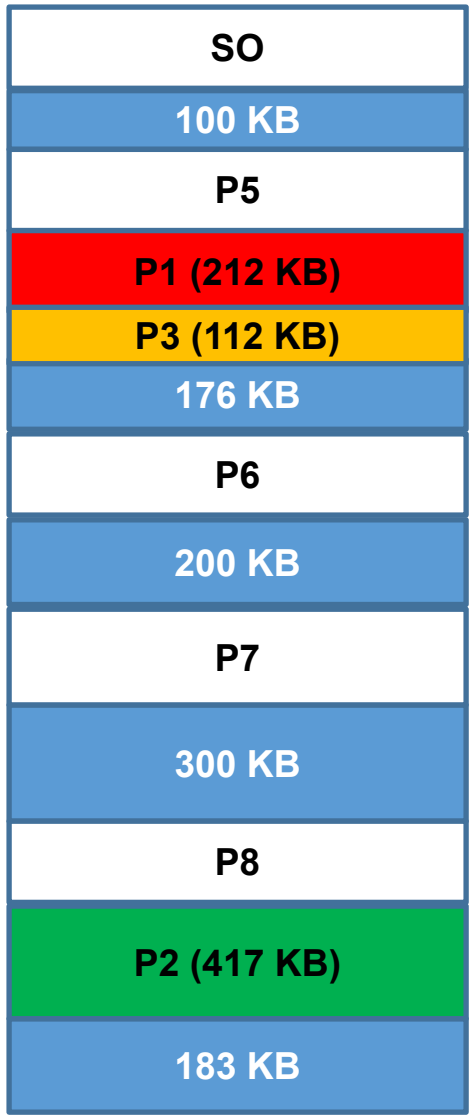
- P1 richiede 212 KB
- P2 richiede 417 KB
- P3 richiede 112 KB
- P4 richiede 426 KB



Soluzione Esercizio 1



Soluzione Esercizio 1



P4 (426 KB)

P4 (426KB) dovrà attendere perché non vi è memoria sufficiente per l'allocazione

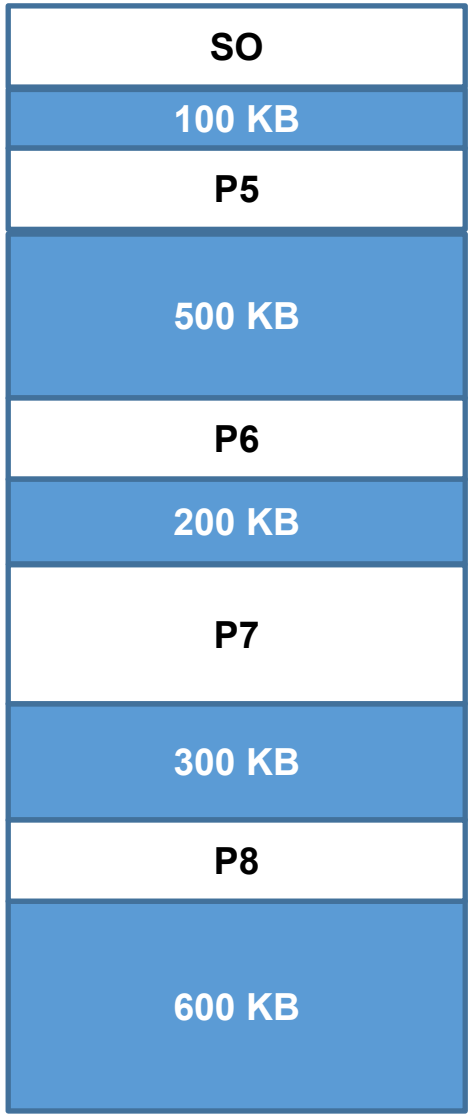
Esercizio 2

Si assuma di avere:

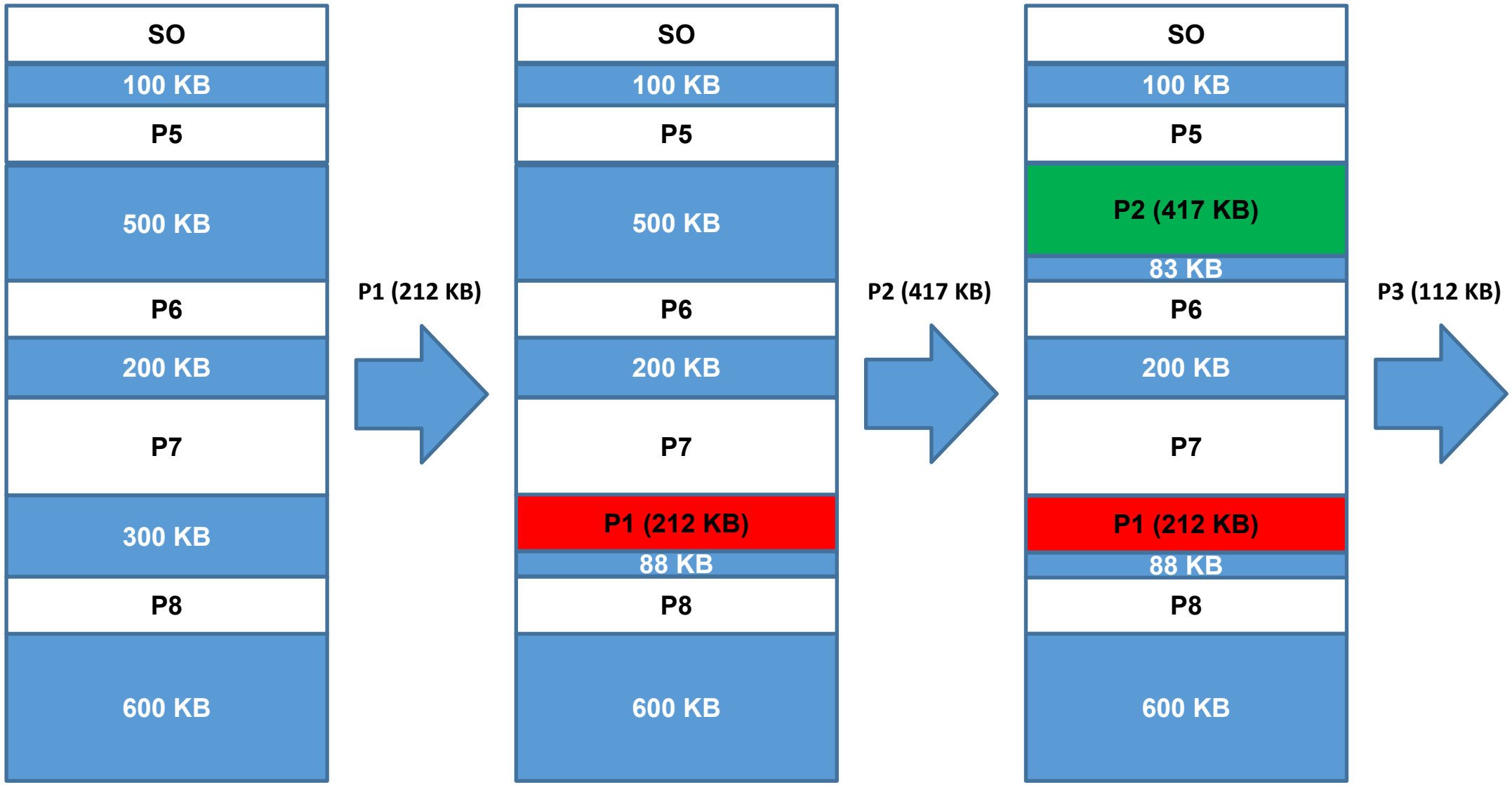
- un sistema di allocazione contigua dei processi in memoria
- la memoria nella situazione illustrata a lato

Utilizzando un approccio best-fit, come verranno allocati i processi seguenti?

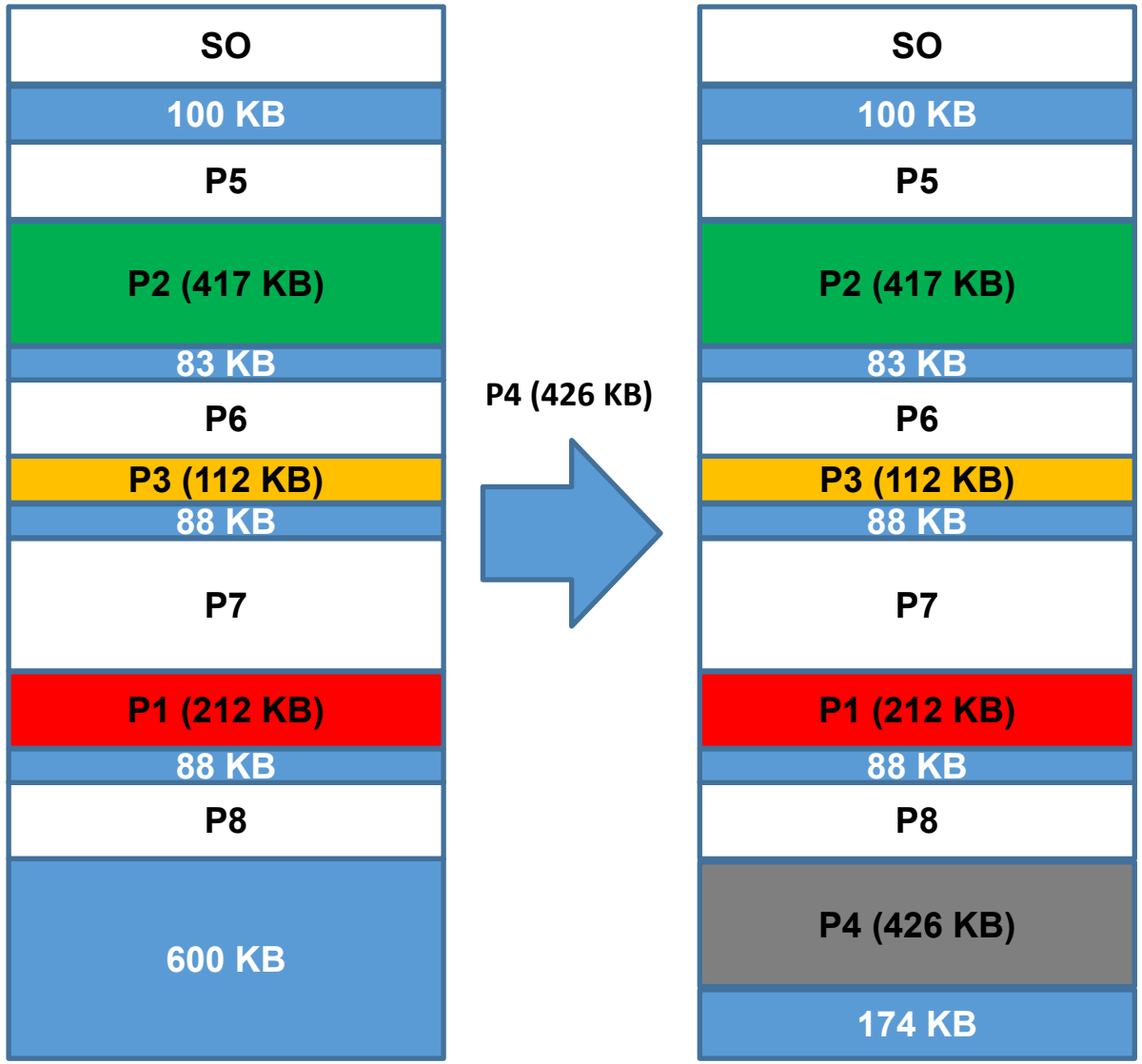
- P1 richiede 212 KB
- P2 richiede 417 KB
- P3 richiede 112 KB
- P4 richiede 426 KB



Soluzione Esercizio 2



Soluzione Esercizio 2



Tutti i processi in attesa sono stati allocati

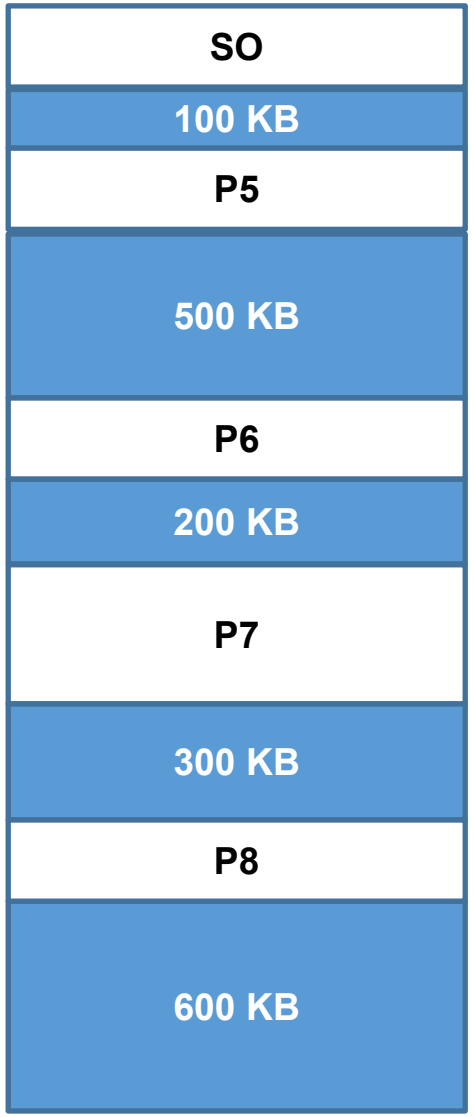
Esercizio 3

Si assuma di avere:

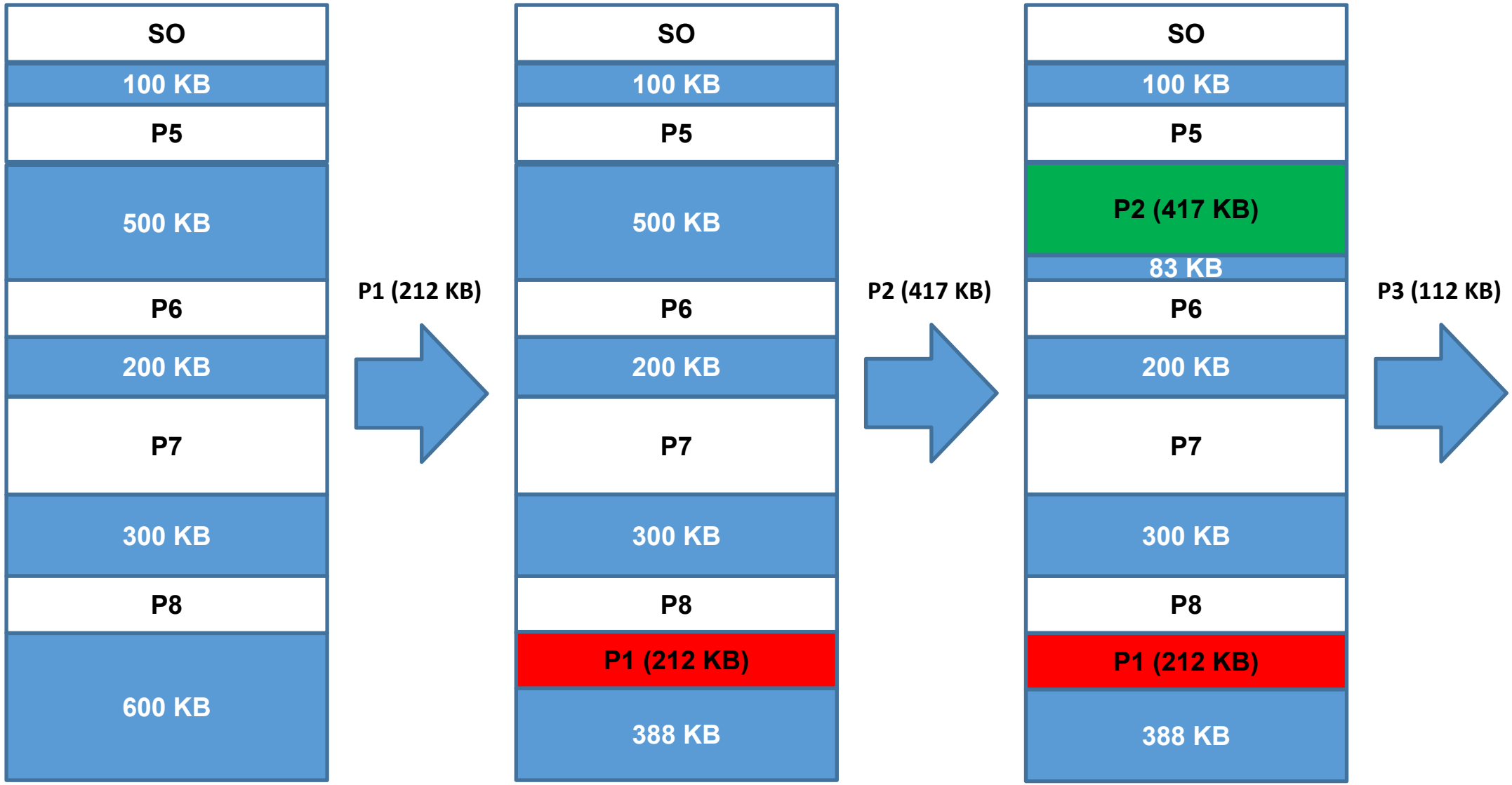
- un sistema di allocazione contigua dei processi in memoria
- la memoria nella situazione illustrata a lato

Utilizzando un approccio worst-fit, come verranno allocati i processi seguenti?

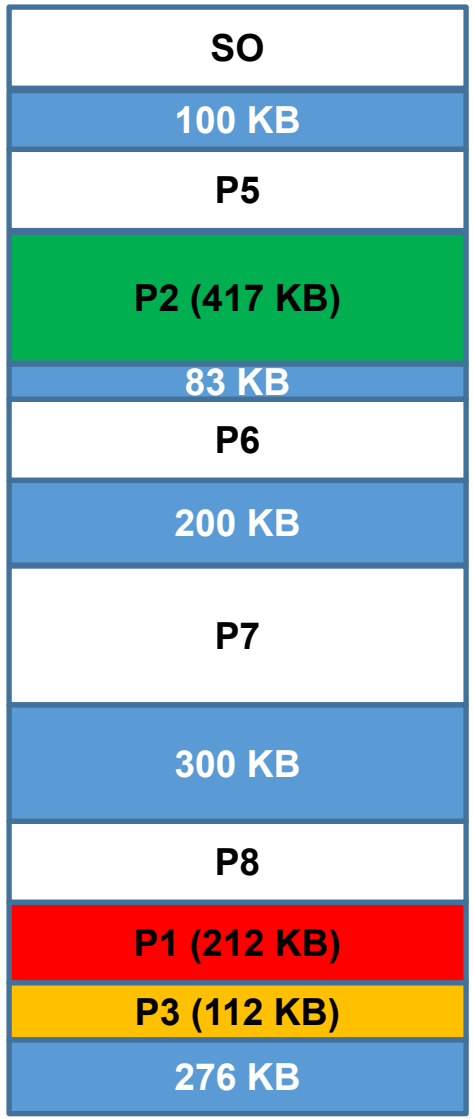
- P1 richiede 212 KB
- P2 richiede 417 KB
- P3 richiede 112 KB
- P4 richiede 426 KB



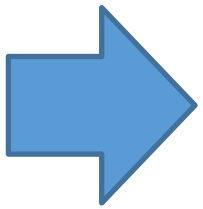
Soluzione Esercizio 3



Soluzione Esercizio 3



P4 (426 KB)

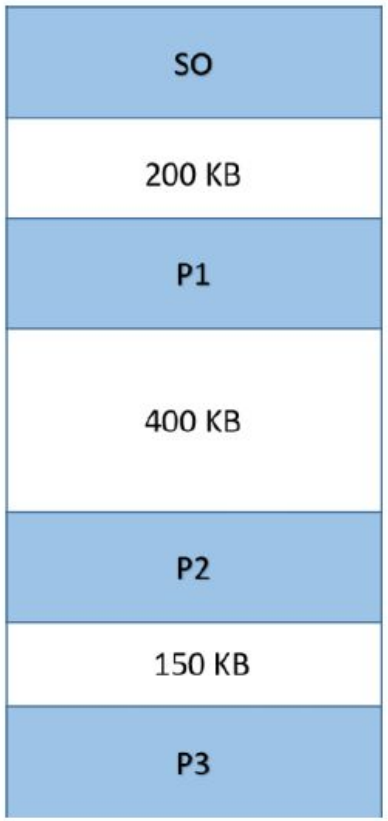


P4 (426KB) dovrà attendere perché non vi è memoria sufficiente per l'allocazione

Esame del 7 Nov 2022

Esercizio 2 (max 7,5 punti)

Si assuma di avere la memoria nella situazione illustrata a lato, con la seguente lista delle allocazioni disponibili:
200 KB, 400 KB, 150 KB.
Si vogliono allocare in memoria per intero, con una politica di scheduling FCFS, i seguenti processi (elencati in ordine di arrivo)
P4 (350 KB), P5 (120 KB), P6 (280 KB), P7 (130 KB).
Indicare come verranno allocati P4, P5, P6 e P7 adottando un approccio di allocazione best-fit.
Motivare la risposta, mostrando graficamente la variazione dell'occupazione della memoria con l'allocazione dei processi

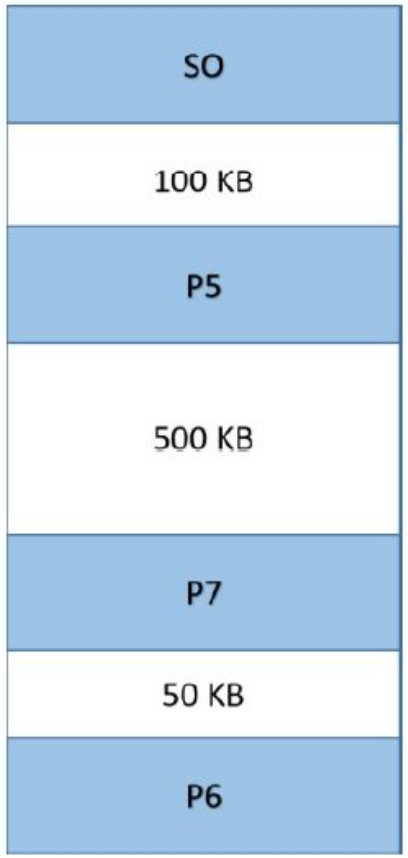


SO 21/22 **0711**

Esame del 15 Feb 2022

Esercizio 2 (max 7,5 punti)

Si assuma di avere la memoria nella situazione illustrata a lato, con la seguente lista delle allocazioni disponibili:
100 KB, 500 KB, 50 KB.
Si vogliono allocare in memoria per intero i seguenti processi, elencati in ordine di arrivo
P1 (350 KB), P2 (20 KB), P3 (280 KB), P4 (130 KB).
Indicare come verranno allocati P1, P2, P3 e P4 adottando un approccio di allocazione worst-fit con una politica di scheduling FCFS.
Motivare la risposta, mostrando graficamente la variazione dell'occupazione della memoria con l'allocazione dei processi sopra elencati.



SO 21/22 **1502**

Esame del 14 Lug 2021

Esercizio 2 (max 7,5 punti)

Si assuma di avere la memoria nella situazione illustrata a lato, con la seguente lista delle allocazioni disponibili:
100 KB, 700 KB, 300 KB, 200 KB.
Si vogliono allocare in memoria per intero i seguenti processi P1 (350 KB), P2 (20 KB), P3 (280 KB), P4 (150 KB) adottando un approccio di allocazione best-fit con una politica di scheduling FCFS. Come verranno allocati P1, P2, P3 e P4? Motivare la risposta, mostrando graficamente come vari l'occupazione della memoria con l'allocazione dei processi sopra elencati.

SO 20/21 **1407**

UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Operativi

Memoria centrale

Docente:
Domenico Daniele Bloisi