

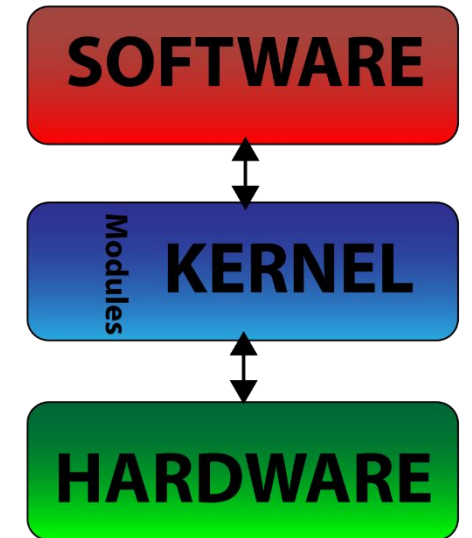
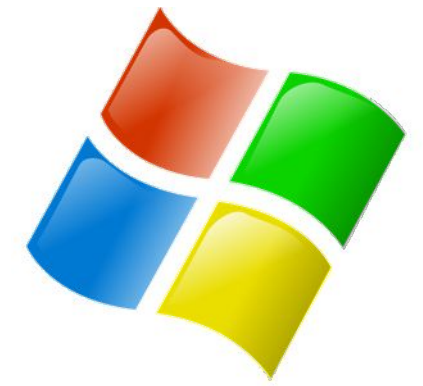


**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

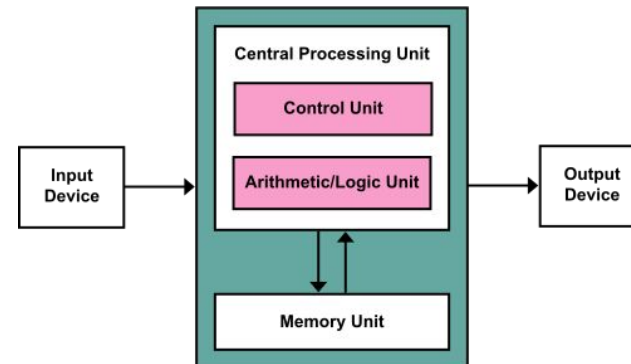
*Corso di Sistemi Operativi*

**Esercitazione**

**Memoria  
centrale**

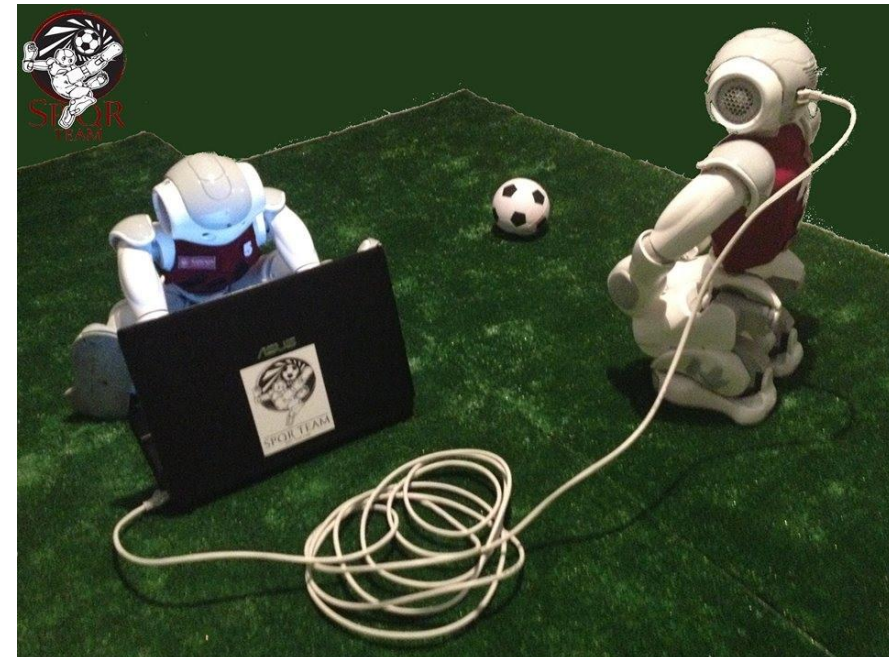
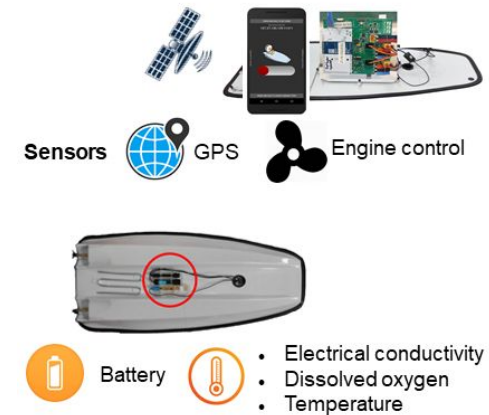


Docente:  
**Domenico Daniele  
Bloisi**



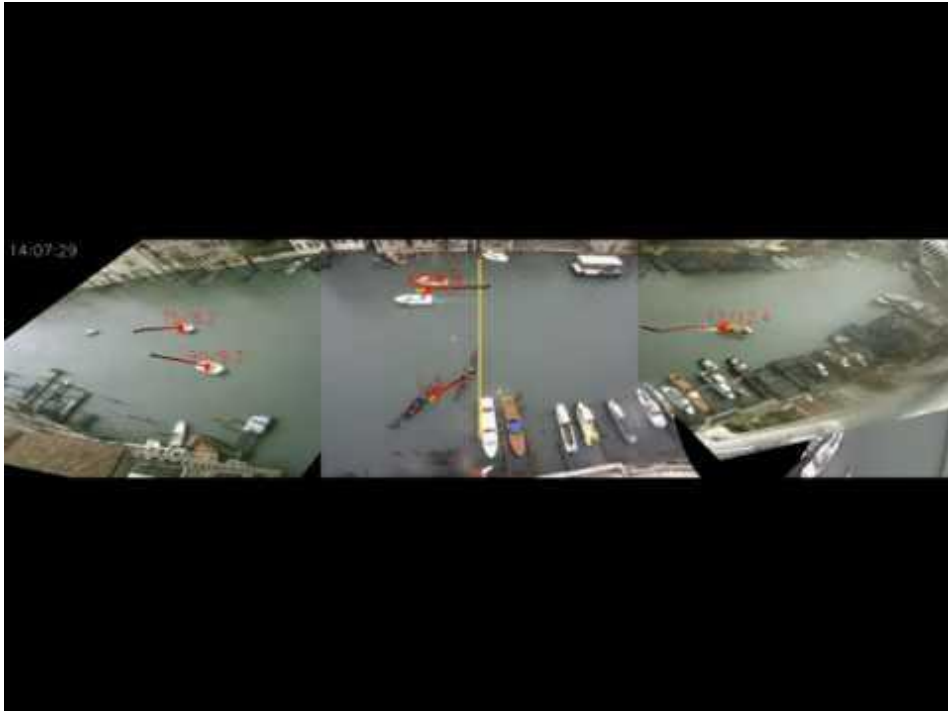
# Domenico Daniele Bloisi

- 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

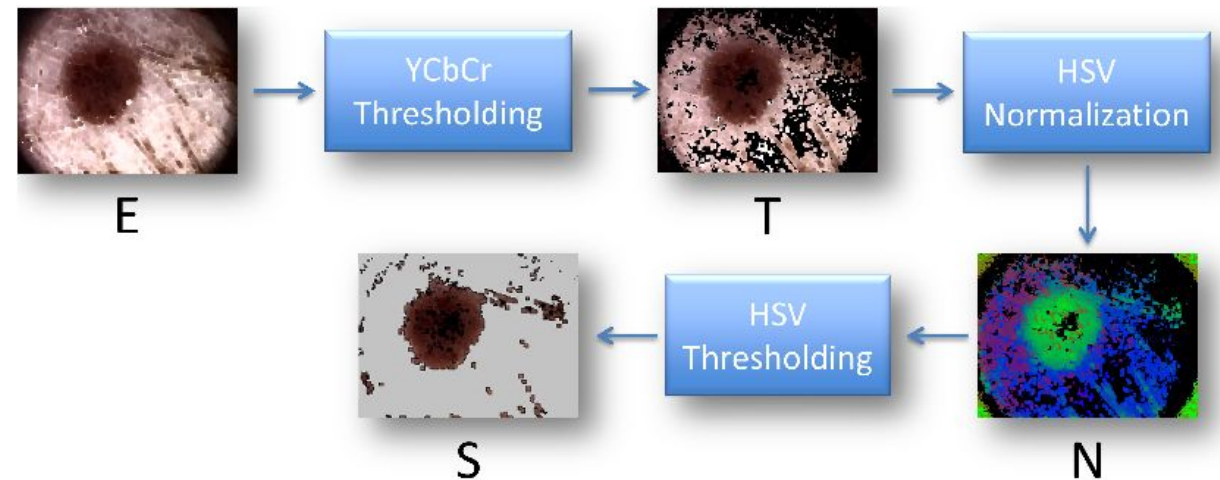
- Intelligent surveillance
- Robot vision
- Medical image analysis



[https://youtu.be/9a70Ucgbi\\_U](https://youtu.be/9a70Ucgbi_U)



<https://youtu.be/2KHNZX7UIWQ>





# 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](mailto:domenico.bloisi@unibas.it)



# Domanda 1

---

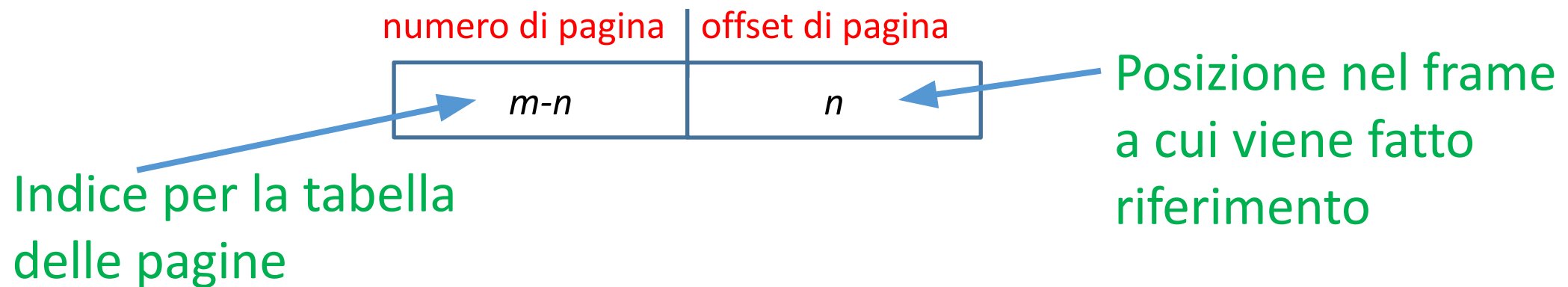
Si supponga di avere un sistema con uno spazio di indirizzi di indirizzi a 14 bit.

Ipotizzando pagine da 2KB quante entry avrà la tabella delle pagine?

# Tabella delle pagine

---

Se la dimensione degli indirizzi logici è  $2^m$  e la dimensione di una pagina è  $2^n$  byte, allora gli  $m-n$  bit più significativi di un indirizzo logico indicano il numero di pagina e gli  $n$  bit meno significativi l'offset di pagina





# Risposta Domanda 1

---

Si supponga di avere un sistema con uno spazio di indirizzi di indirizzi a 14 bit  $\square m = 14$

Ipotizzando pagine da 2KB  $\square 2KB = 2^{11}B \square n = 11$

quante entry avrà la tabella delle pagine?

$$2^{m-n} = 2^{(14-11)} = 2^3 = 8$$

# Domanda 2

---

Si supponga di avere un sistema con uno spazio di indirizzi di indirizzi a 14 bit. Ipotizzando pagine da 2KB, un processo con registro base 0 e limite 10468, quante pagine occuperà?

# Risposta Domanda 2

---

Si supponga di avere un sistema con uno spazio di indirizzi di indirizzi a 14 bit. Ipoteizzando pagine da 2KB

□ ogni pagina può contenere  $2\text{KB} = 2^{11}\text{B} = 2048 \text{ byte}$

un processo con registro base 0 e limite 10468, quante pagine occuperà? □ Ipoteizzando di occupare 5 pagine,

avremo  $2048 \times 5 = 10240 \text{ byte} < 10468$  □ Occupando

6 pagine, avremo  $2048 \times 6 = 12288 \text{ byte} > 10468$

# Bit di validità

Avendo una tabella delle pagine da 8 entry e solo 6 pagine occupate, servirà (per proteggere la memoria) rendere inaccessibili i puntatori delle ultime due entry della tabella delle pagine attraverso un bit di validità

12.287	pagina 5
10.468	pagina 4
	pagina 3
	pagina 2
	pagina 1
00000	pagina 0

numero del frame		bit di validità/ non validità
0	2	v
1	3	v
2	4	v
3	7	v
4	8	v
5	9	v
6	0	i
7	0	i

tabella delle pagine

0	
1	
2	pagina 0
3	pagina 1
4	pagina 2
5	
6	
7	pagina 3
8	pagina 4
9	pagina 5
	⋮
	pagina <i>n</i>

Figura 9.13 Bit di validità (v) o non validità (i) in una tabella delle pagine.

# Domanda 3

---

Spiegare la differenza tra indirizzi logici e indirizzi fisici

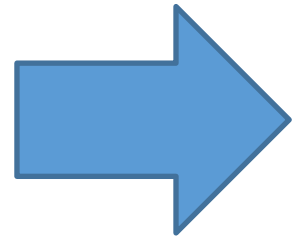


# Risposta Domanda 3

---

Gli indirizzi logici sono gli indirizzi ideali generati dai programmi utenti relativi alla locazione 0 in memoria

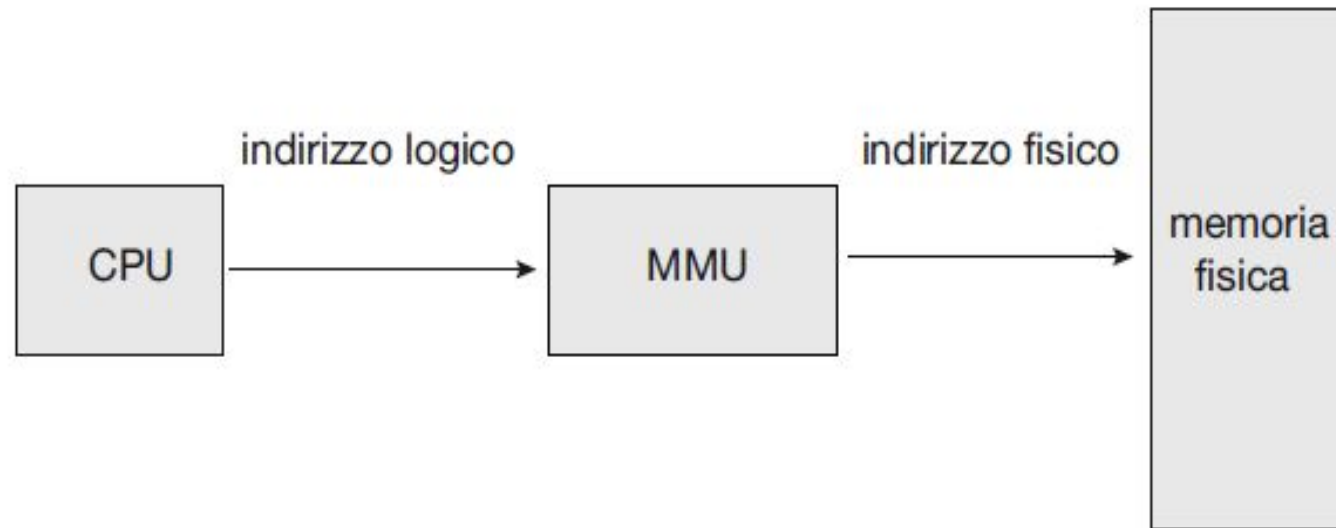
Gli indirizzi fisici sono i reali indirizzi usati per recuperare e immagazzinare dati nella memoria del calcolatore



# Risposta Domanda 3

---

La figura sottostante mostra come avviene la traduzione da indirizzo logico a indirizzo fisico nei moderni sistemi di elaborazione



**Figura 9.4** Unità di gestione della memoria (MMU).

# Domanda 4

---

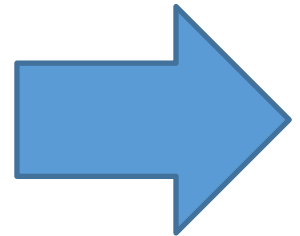
Spiegare la differenza tra frammentazione interna e frammentazione esterna

# Risposta Domanda 4

---

La frammentazione interna si verifica quando un processo  $p$  non occupa interamente la porzione di memoria che gli è stata allocata.

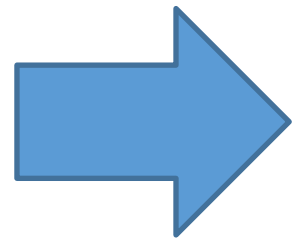
Tale spazio rimane inutilizzabile da altri processi fino al rilascio della memoria allocata a  $p$ .



# Risposta Domanda 4

---

La frammentazione esterna si verifica quando c'è una quantità totale di memoria libera sufficiente a soddisfare la richiesta di memoria da parte di un nuovo processo, tuttavia non sono disponibili porzioni di memoria contigue così grandi da contenere il nuovo processo.

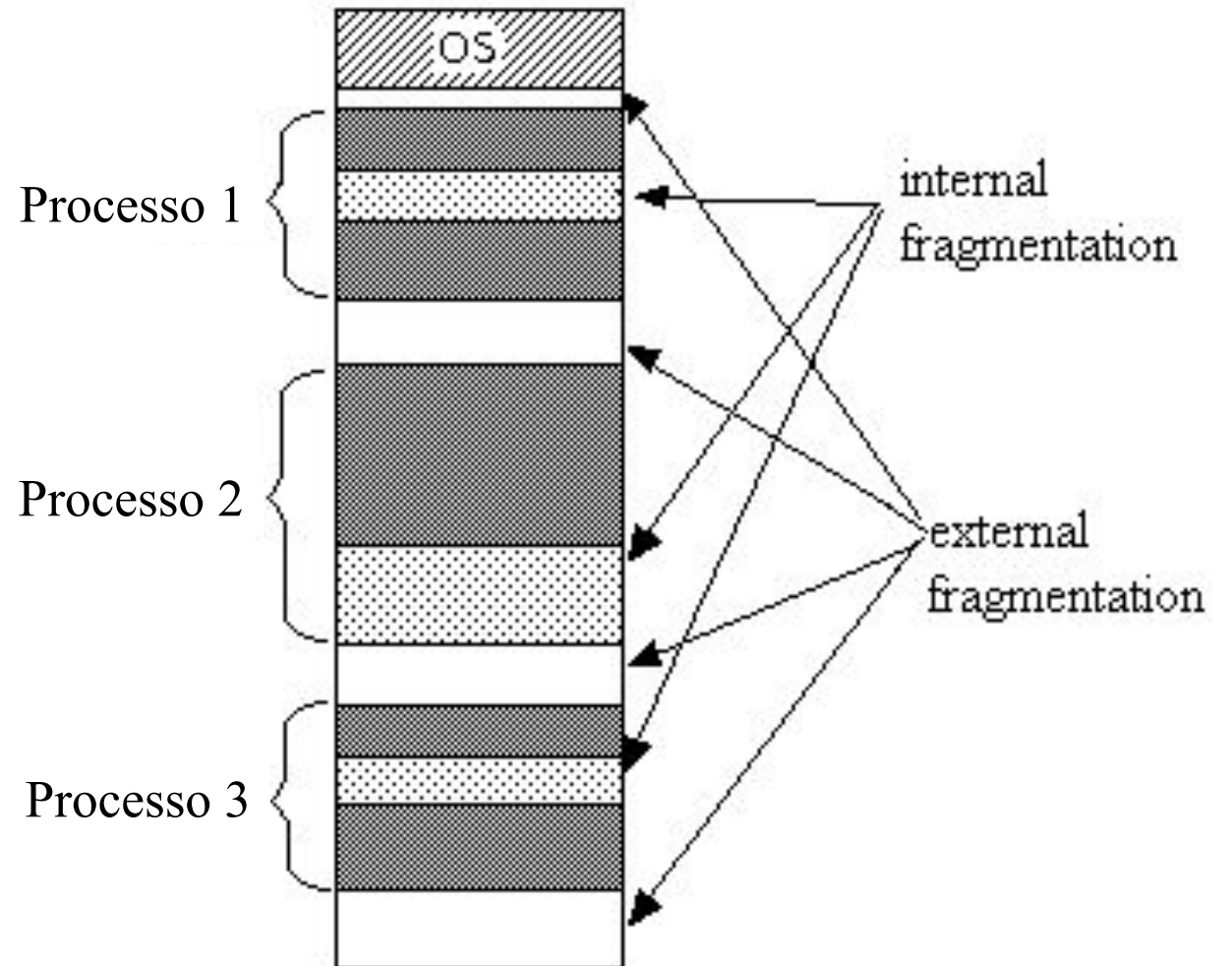




# Risposta Domanda 4

---

La figura a lato mostra visivamente la differenza tra frammentazione interna e frammentazione esterna



# Domanda 5

---

Fornire una descrizione dei seguenti algoritmi di allocazione:

- First-fit
- Best-fit
- Worst-fit

# Risposta Domanda 5

---

- First-fit: si scorre la lista delle locazioni di memoria disponibili fino a trovare il primo blocco che sia sufficientemente grande
- Best-fit: si scandisce tutta la lista delle locazioni di memoria disponibili e si alloca il più piccolo blocco che sia sufficientemente grande
- Worst-fit: si scandisce tutta la lista delle locazioni di memoria disponibili e si alloca il più piccolo blocco che sia sufficientemente grande. La politica worst-fit produce le parti inutilizzate più grandi, le quali possono essere più utili rispetto alle piccole porzioni create da best-fit.

# Domanda 6

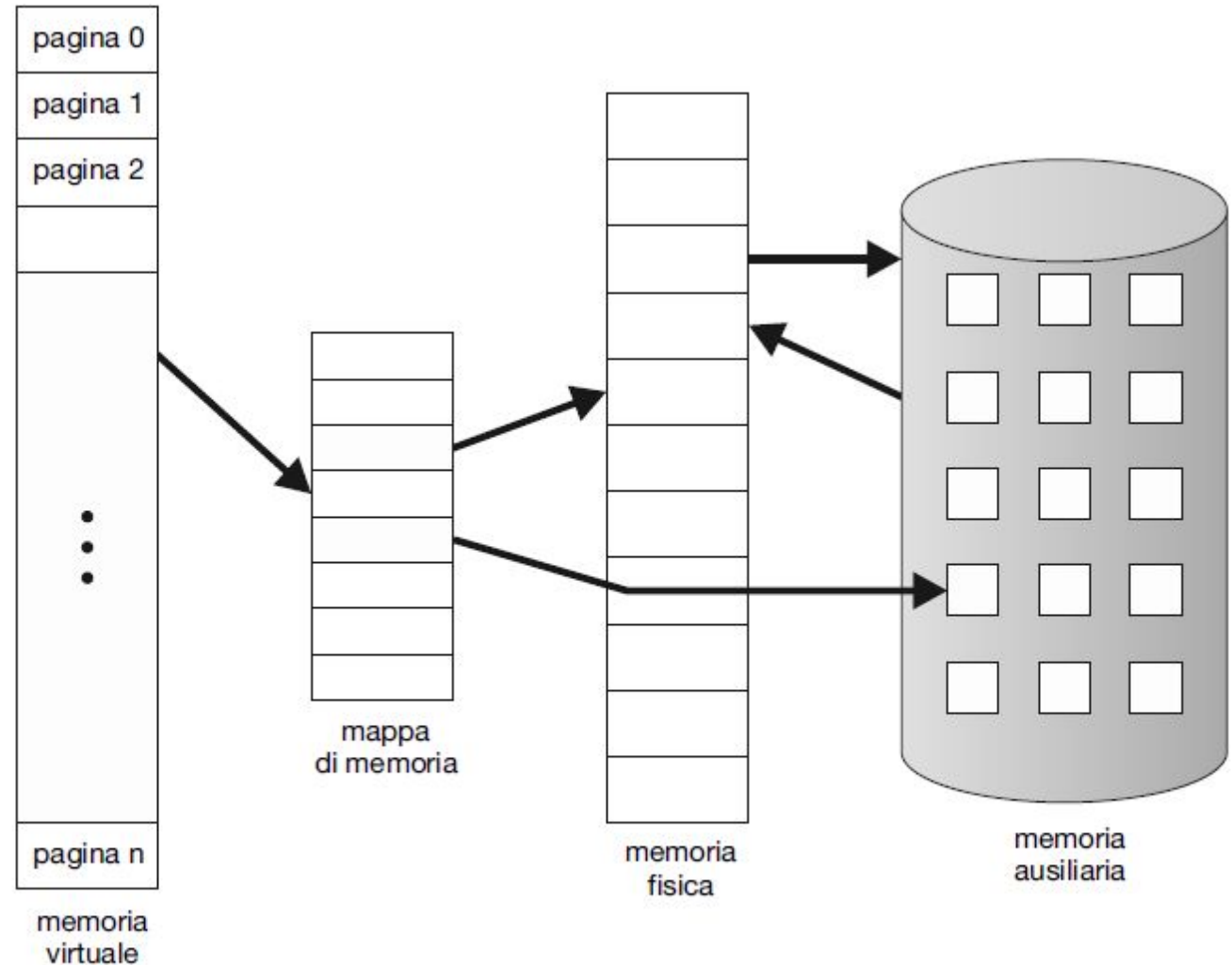
---

Che cos'è la memoria virtuale? Integrare la risposta con un opportuno schema grafico

# Risposta Domanda 6

La **memoria virtuale** è una tecnica che permette di eseguire processi che possono anche non essere completamente contenuti in memoria

La **memoria virtuale** facilita la programmazione, poiché il programmatore non deve preoccuparsi della quantità di memoria fisica disponibile, ma può concentrarsi sul problema da risolvere con il programma



**Figura 10.1** Schema che mostra una memoria virtuale più grande di quella fisica.



# Esercizio 1

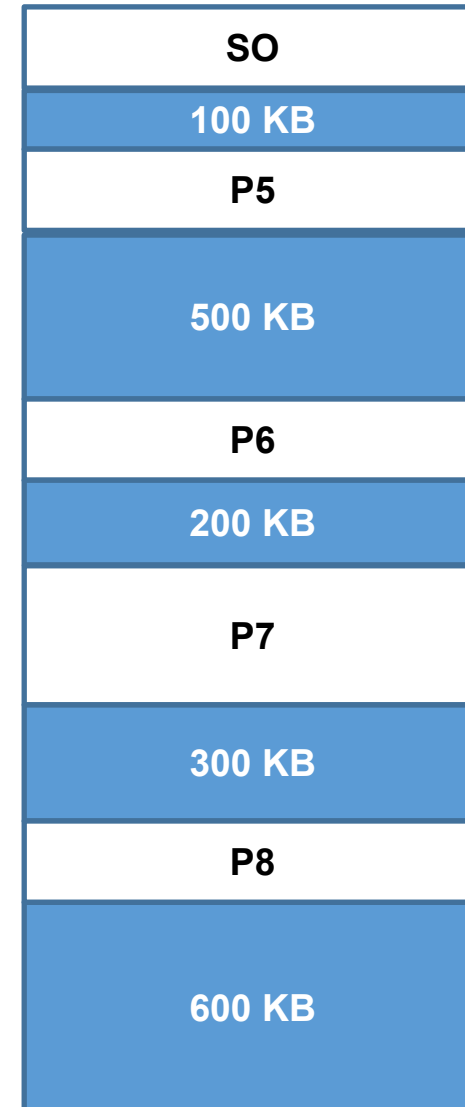
---

Si assuma di avere:

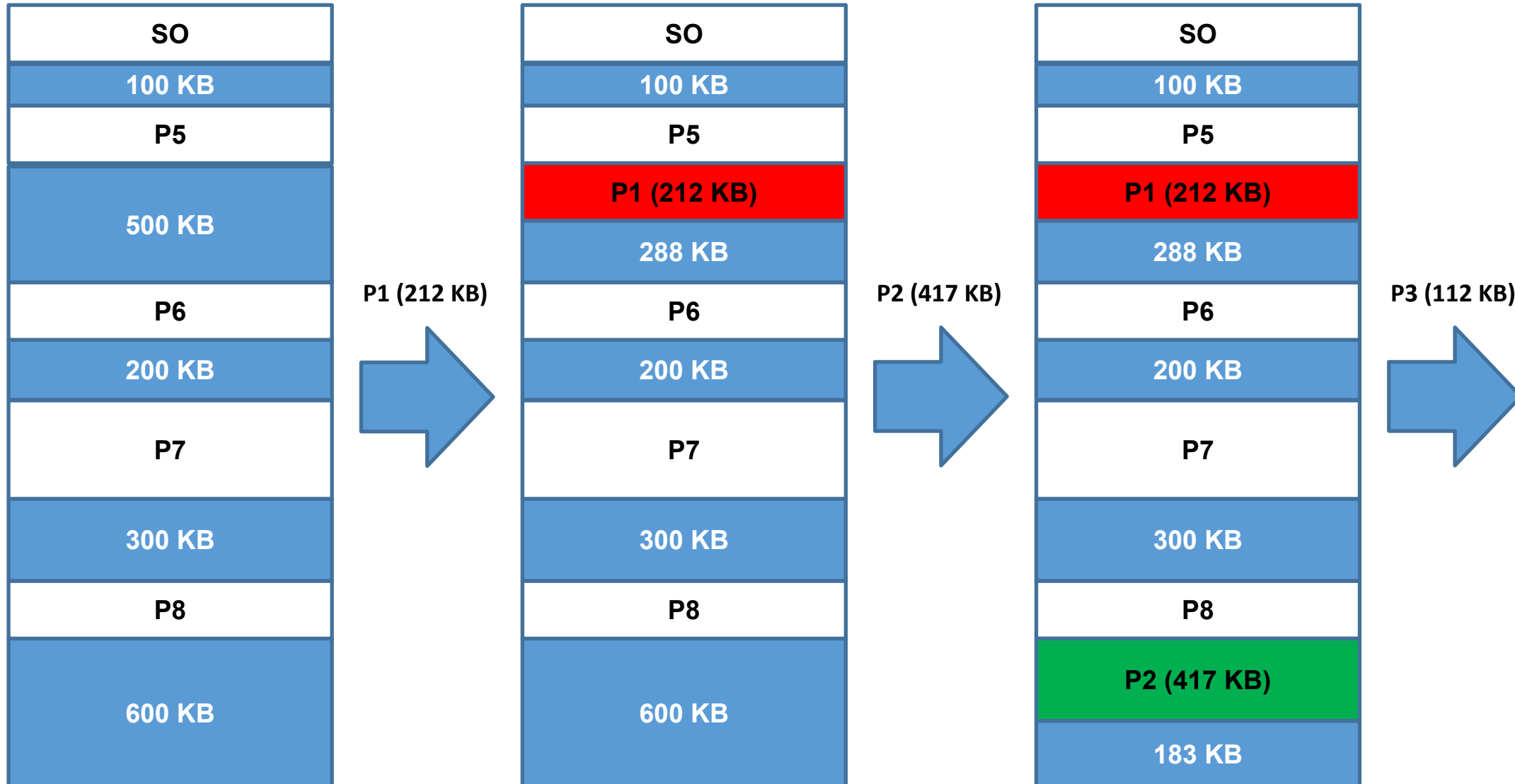
- un sistema di allocazione contigua dei processi in memoria
- la memoria nella situazione illustrata a lato

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

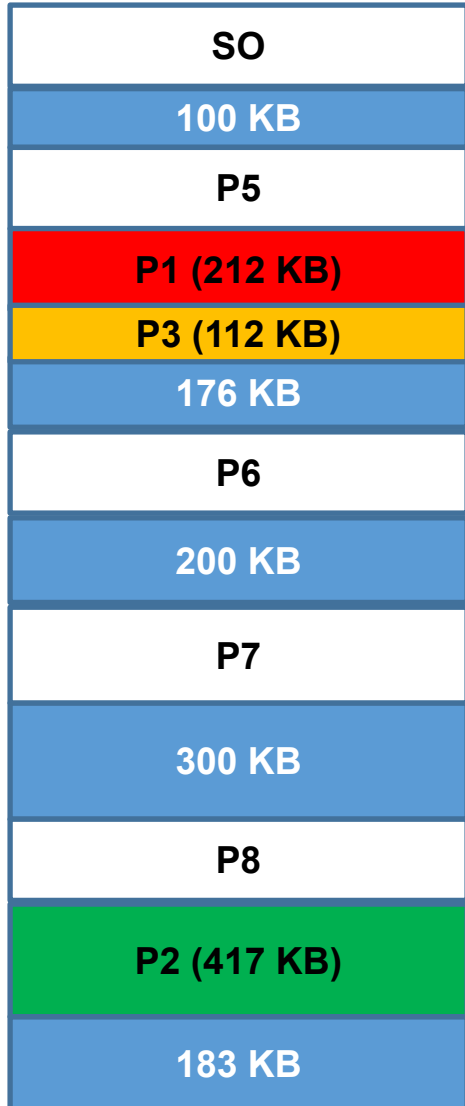


# Soluzione Esercizio 1

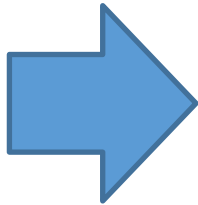


# Soluzione Esercizio 1

---



P4 (426 KB)



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

# Esercizio 2

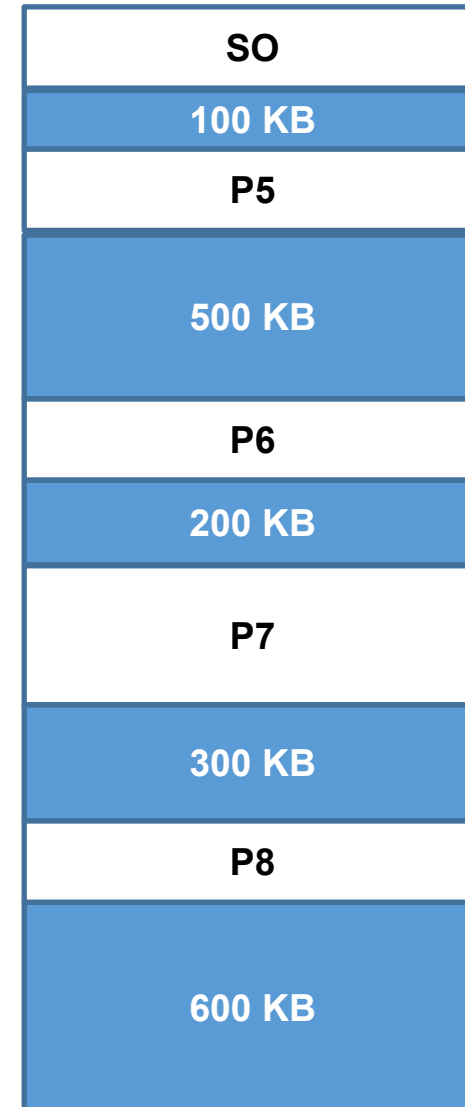
---

Si assuma di avere:

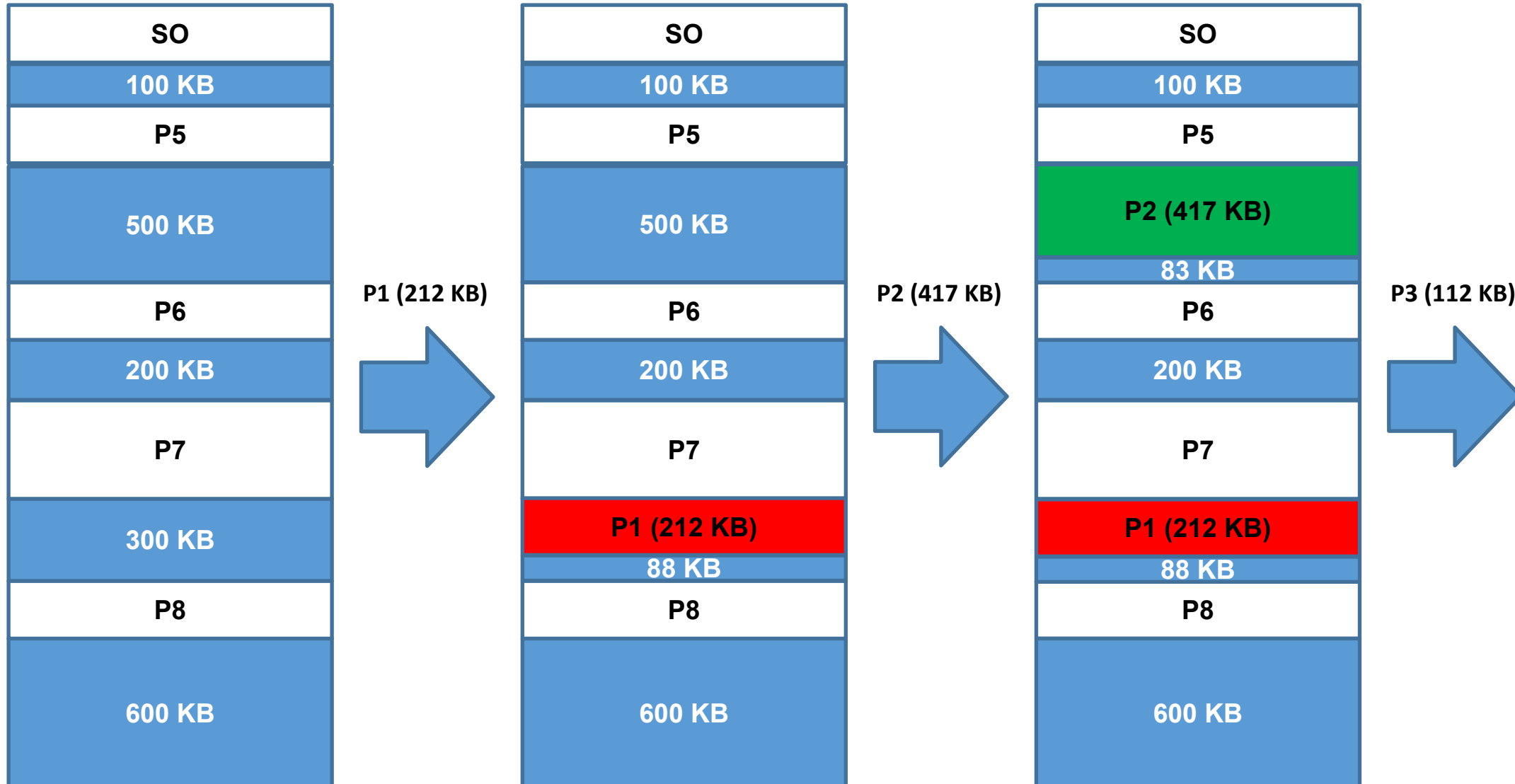
- un sistema di allocazione contigua dei processi in memoria
- la memoria nella situazione illustrata a lato

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

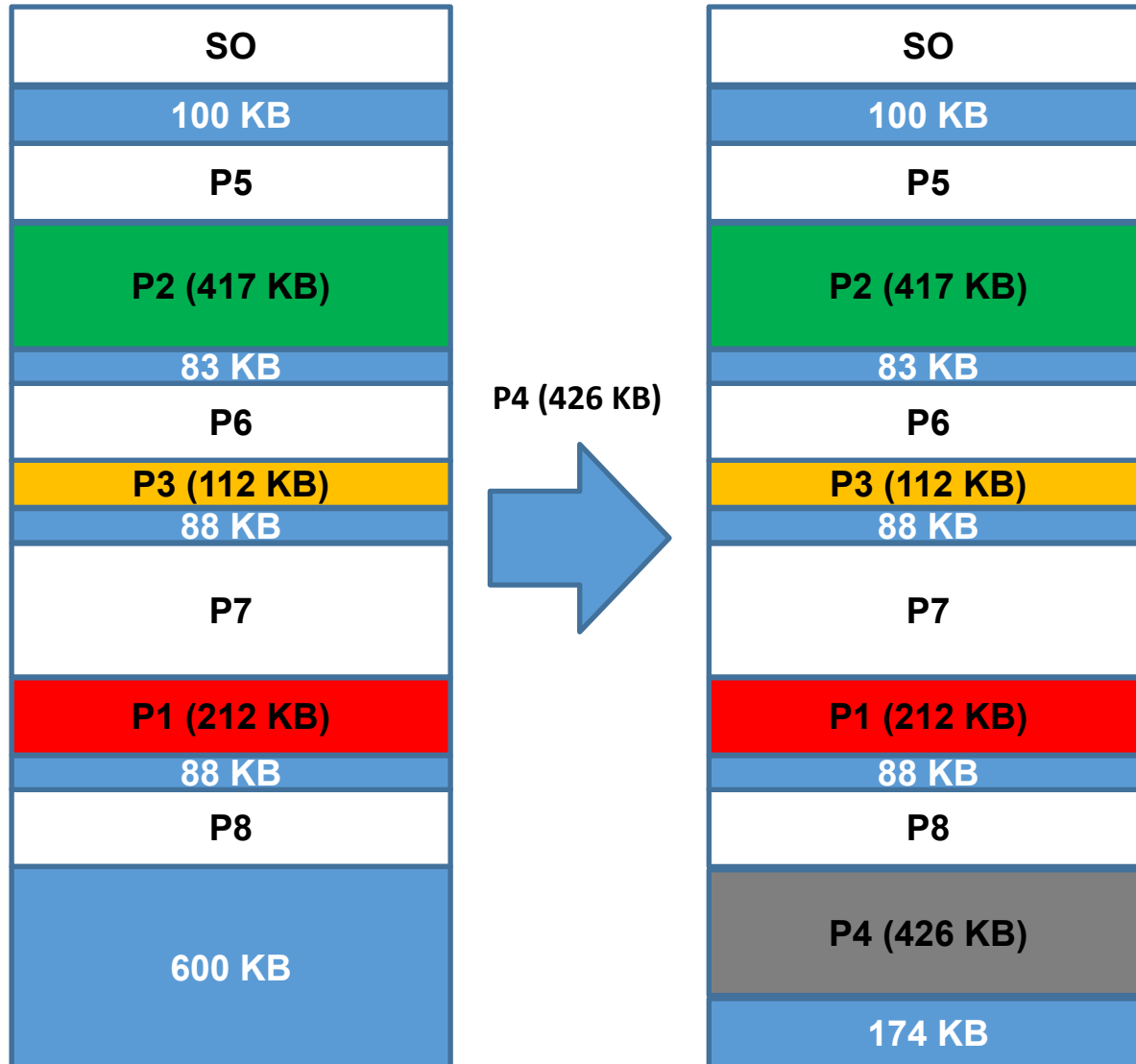
- P1 richiede 212 KB
- P2 richiede 417 KB
- P3 richiede 112 KB
- P4 richiede 426 KB



# Soluzione Esercizio 2



# Soluzione Esercizio 2



Tutti i processi in attesa sono stati allocati

# Esercizio 3

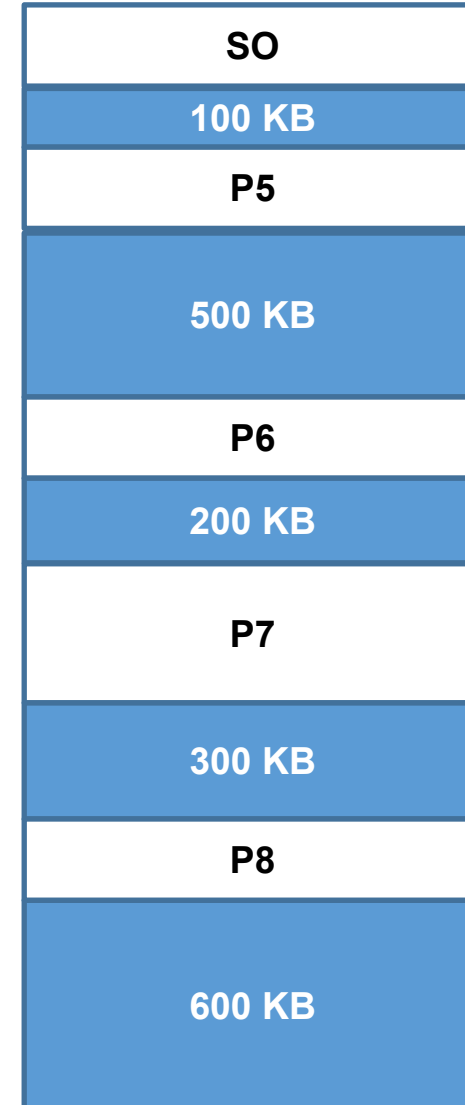
---

Si assuma di avere:

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

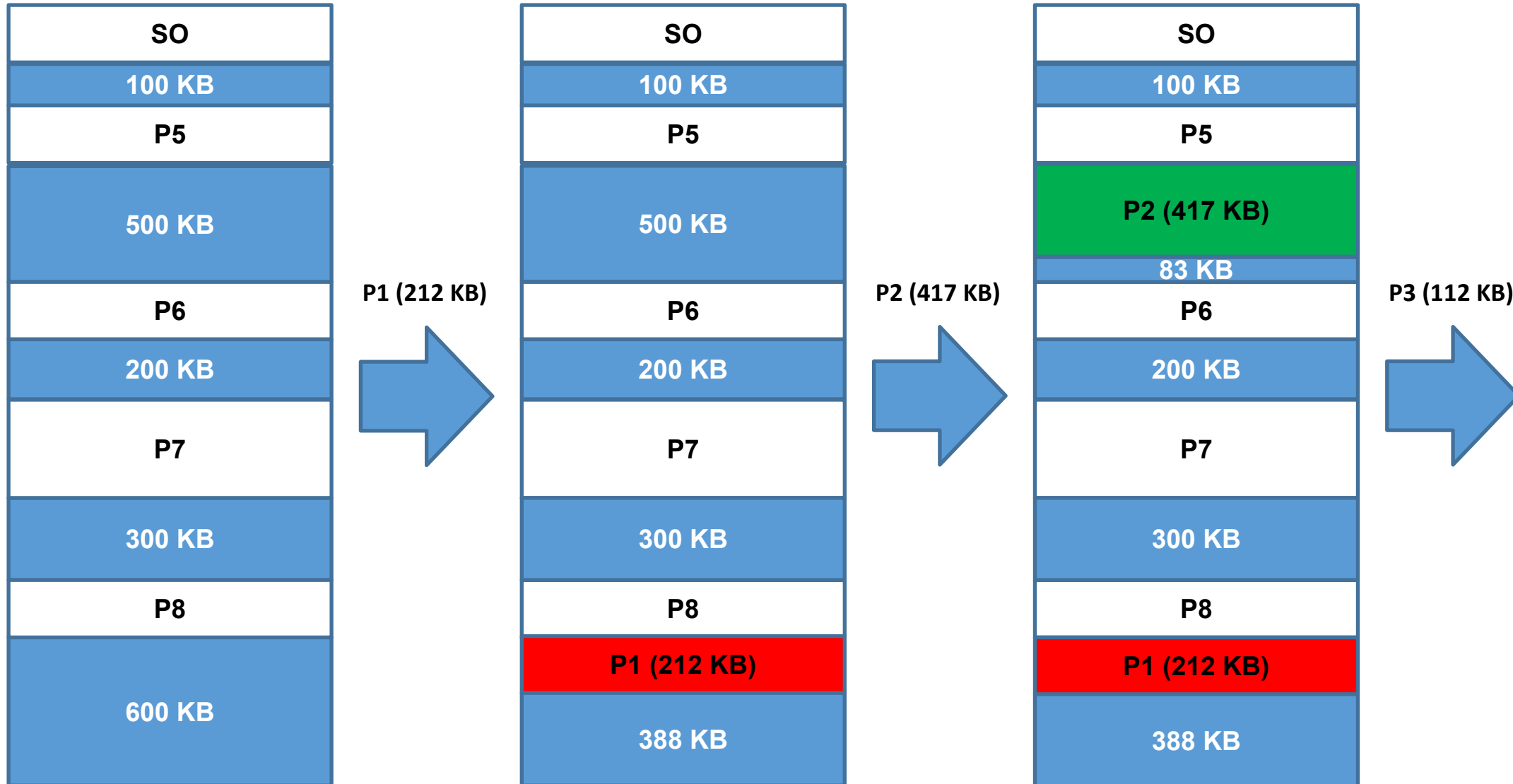
Utilizzando un approccio **worst-fit**, come verranno allocati i processi seguenti?

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



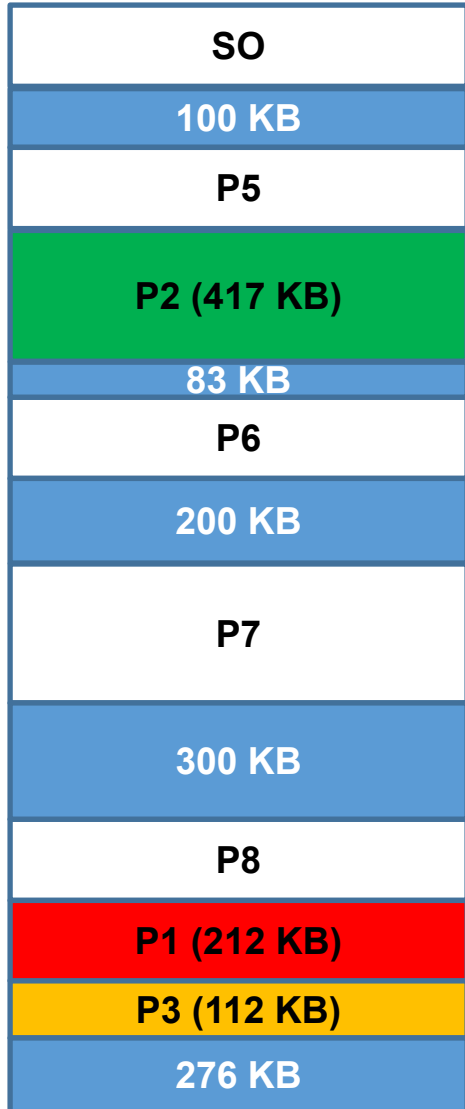


# Soluzione Esercizio 3

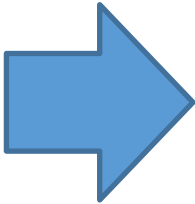


# Soluzione Esercizio 3

---



P4 (426 KB)



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

# Esercizio 4

---

Si assuma di avere un sistema con

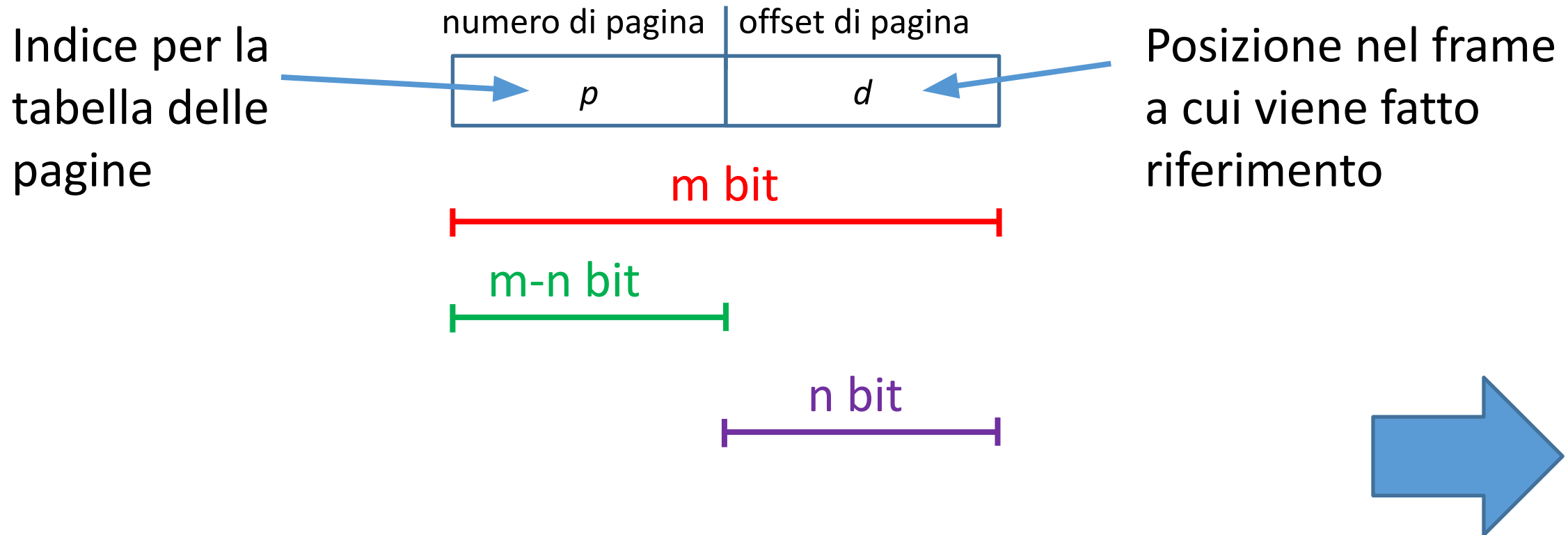
- una dimensione di pagina pari a 1KB
- uno spazio degli indirizzi logici di 15 bit

Di quante pagine può disporre il sistema?

# Soluzione Esercizio 4

---

Ogni indirizzo (**logico**) generato dalla CPU è diviso in:



# Soluzione Esercizio 4

1

spazio degli  
indirizzi logici  
di 15 bit

numero di pagina | offset di pagina



15 bit

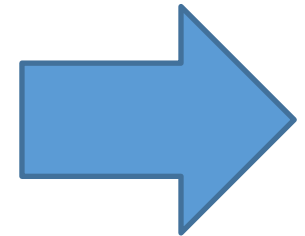
10 bit

5 bit

2 dimensione di  
pagina pari a 1KB

$$1\text{KB} = 2^{10}\text{byte}$$

3



# Soluzione Esercizio 4

---

- Una dimensione di pagina pari a 1KB porta ad occupare 10 bit per l'indirizzamento poichè  $1\text{KB} = 2^{10}\text{byte}$
- Avendo uno spazio degli indirizzi logici di 15 bit, restano 5 bit da sfruttare.
- Quindi, il sistema potrà disporre di  $2^5$  pagine

# Esercizio 5

---

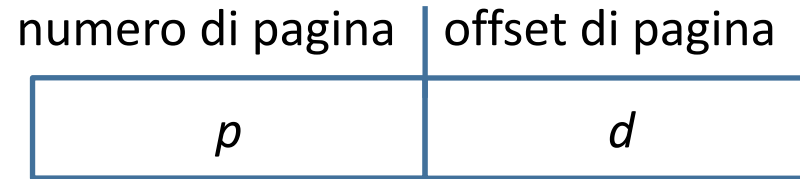
Si assuma di avere un sistema con

- uno spazio degli indirizzi logici di 15 bit
- 8 pagine

Quanto sono grandi le pagine del sistema?

# Soluzione Esercizio 5

1 spazio degli indirizzi logici di 15 bit



15 bit

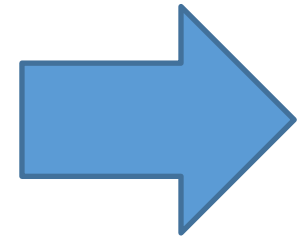
2 numero di pagine pari a 8

$8 = 2^3$

3 bit

12 bit

3





# Soluzione Esercizio 5

---

- Il sistema dispone di  $2^3$  pagine
- Avendo uno spazio degli indirizzi logici di 15 bit, restano 12 bit da sfruttare.
- Una dimensione di pagina pari a 4KB porta ad occupare 12 bit per l'indirizzamento poichè  $4KB = 2^{12} \text{byte}$

In conclusione, il sistema avrà pagine da 4KB

# Esercizio 6

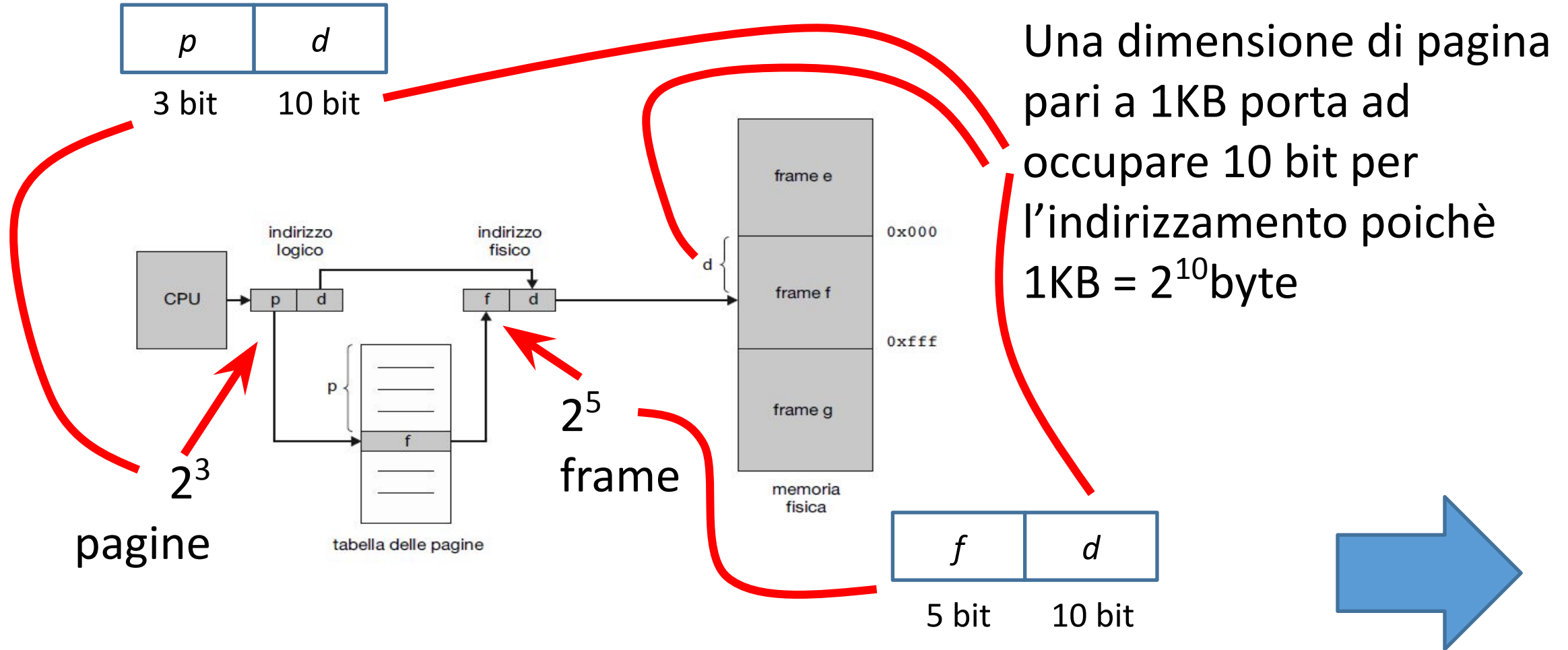
---

Si consideri uno spazio di indirizzi logici di 8 pagine, ciascuna da 1KB, mappata su una memoria fisica da 32 frame.

Quanti bit servono per gli indirizzi logici?

Quanti bit servono per gli indirizzi fisici?

# Soluzione Esercizio 6



# Soluzione Esercizio 6

---

- Il sistema dispone di  $2^3$  pagine
- Una dimensione di pagina pari a 1KB porta ad occupare 10 bit per l'indirizzamento poichè  $1\text{KB} = 2^{10}\text{byte}$
- Il sistema dispone di  $2^5$  frame

In conclusione, il sistema avrà uno spazio degli indirizzi logici da 13 bit e uno spazio degli indirizzi fisici da 15 bit.

# Esercizio 7

---

Sia dato un sottosistema di memoria con paginazione, caratterizzato dalle seguenti dimensioni:

- frame 4 MB
- memoria fisica indirizzabile 128 GB

Si calcoli il numero di bit minimo per indicizzare tutte le pagine

# Soluzione Esercizio 7

---

- Data la dimensione di ogni pagina pari a 4 MB, saranno necessari 22 bit per indicizzare un elemento all'interno della stessa perché  $4 \text{ MB} \square 2^2 \times 2^{20} = 2^{22} \text{ byte}$
- La memoria fisica, invece, necessita di almeno 37 bit perché  $128 \text{ GB} \square 2^7 \times 2^{30} = 2^{37} \text{ byte}$
- Il numero di bit minimo per indicizzare tutte le pagine è quindi pari a  $37 - 22 = 15 \text{ bit}$

# Esercizio 8

---

Sia dato un sottosistema di memoria con paginazione, caratterizzato dalle seguenti dimensioni:

- frame 16 MB
- memoria fisica indirizzabile 16 GB

Si calcoli il numero di bit minimo per indicizzare tutte le pagine

# Soluzione Esercizio 8

---

- Data la dimensione di ogni pagina pari a 16MB, saranno necessari 24 bit per indicizzare un elemento all'interno della stessa perché  $16 \text{ MB} \square 2^4 \times 2^{20} = 2^{24} \text{ byte}$
- La memoria fisica, invece, necessita di almeno 34 bit perché  $16 \text{ GB} \square 2^4 \times 2^{30} = 2^{34} \text{ byte}$
- Il numero di bit minimo per indicizzare tutte le pagine è quindi pari a  $34 - 24 = 10 \text{ bit}$



# Esercizio 9

dimensione della memoria logica: 16 byte

dimensione di una pagina: 4B

memoria fisica: 32 byte

Che indirizzo fisico corrisponde  
all'indirizzo logico 4?

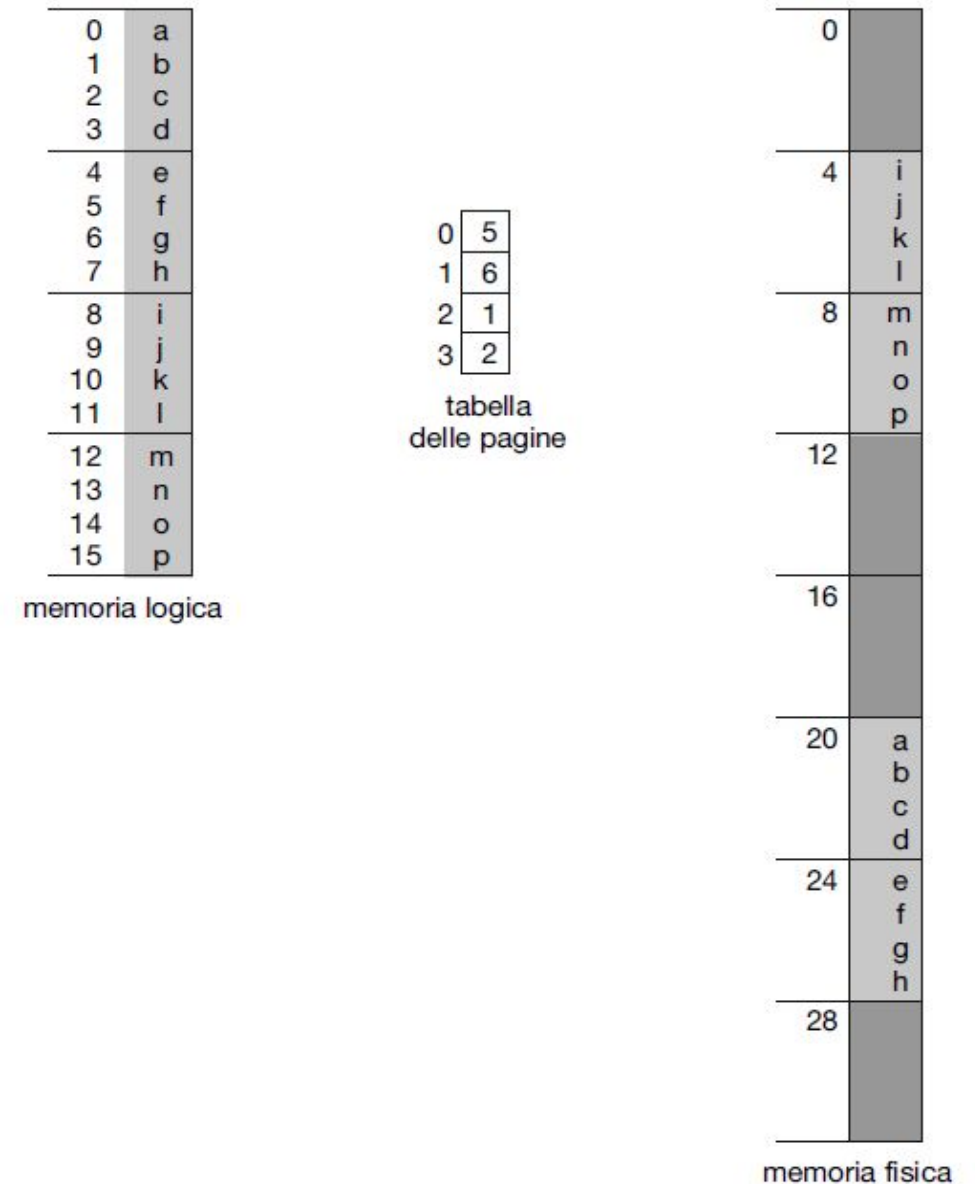


Figura 9.10 Esempio di paginazione per una memoria di 32 byte con pagine di 4 byte.

# Soluzione Esercizio 9

Che indirizzo fisico corrisponde all'indirizzo logico 4?

indirizzo logico 4

pagina	offset
1	0

indirizzo fisico 24

frame	offset
6	0

$$(6 \times 4) + 0 = 24$$

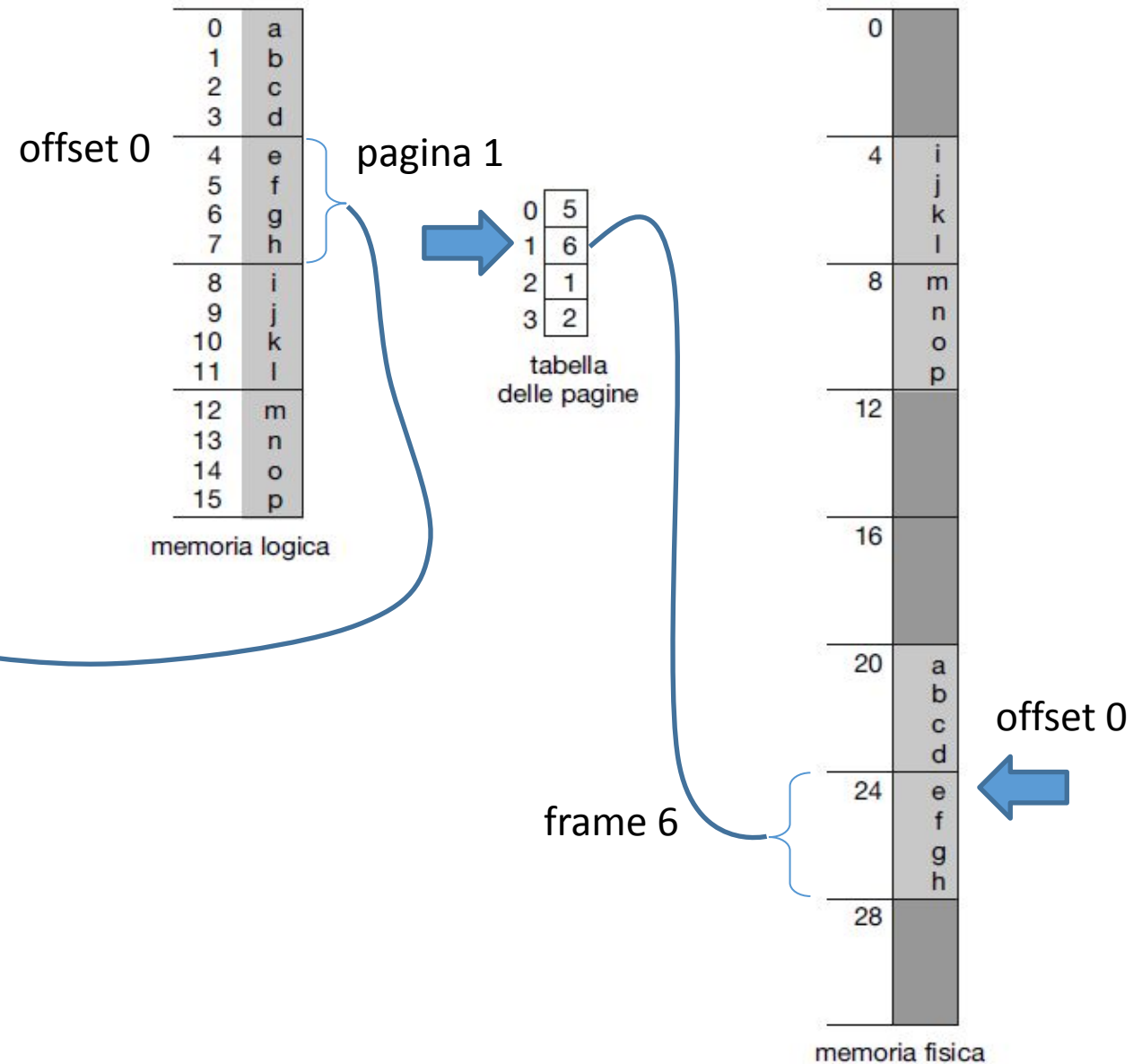


Figura 9.10 Esempio di paginazione per una memoria di 32 byte con pagine di 4 byte.

# Esercizio 10

dimensione della memoria logica: 16 byte

dimensione di una pagina: 4B

memoria fisica: 32 byte

Che indirizzo fisico corrisponde  
all'indirizzo logico 13?

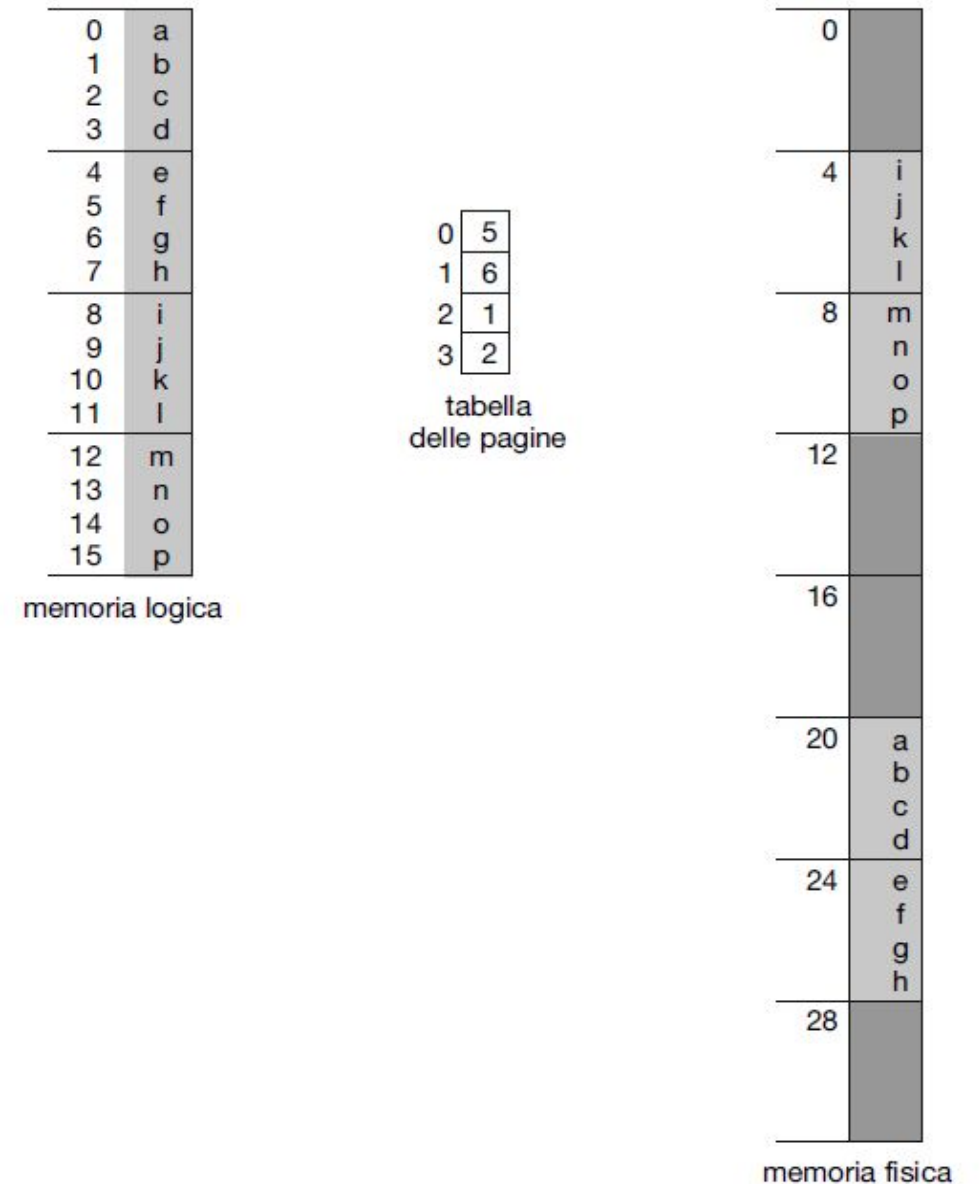
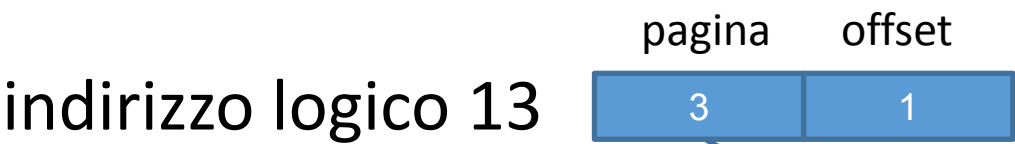


Figura 9.10 Esempio di paginazione per una memoria di 32 byte con pagine di 4 byte.

# Soluzione Esercizio 10

Che indirizzo fisico corrisponde all'indirizzo logico 13?



$(2 \times 4) + 1 = 9$

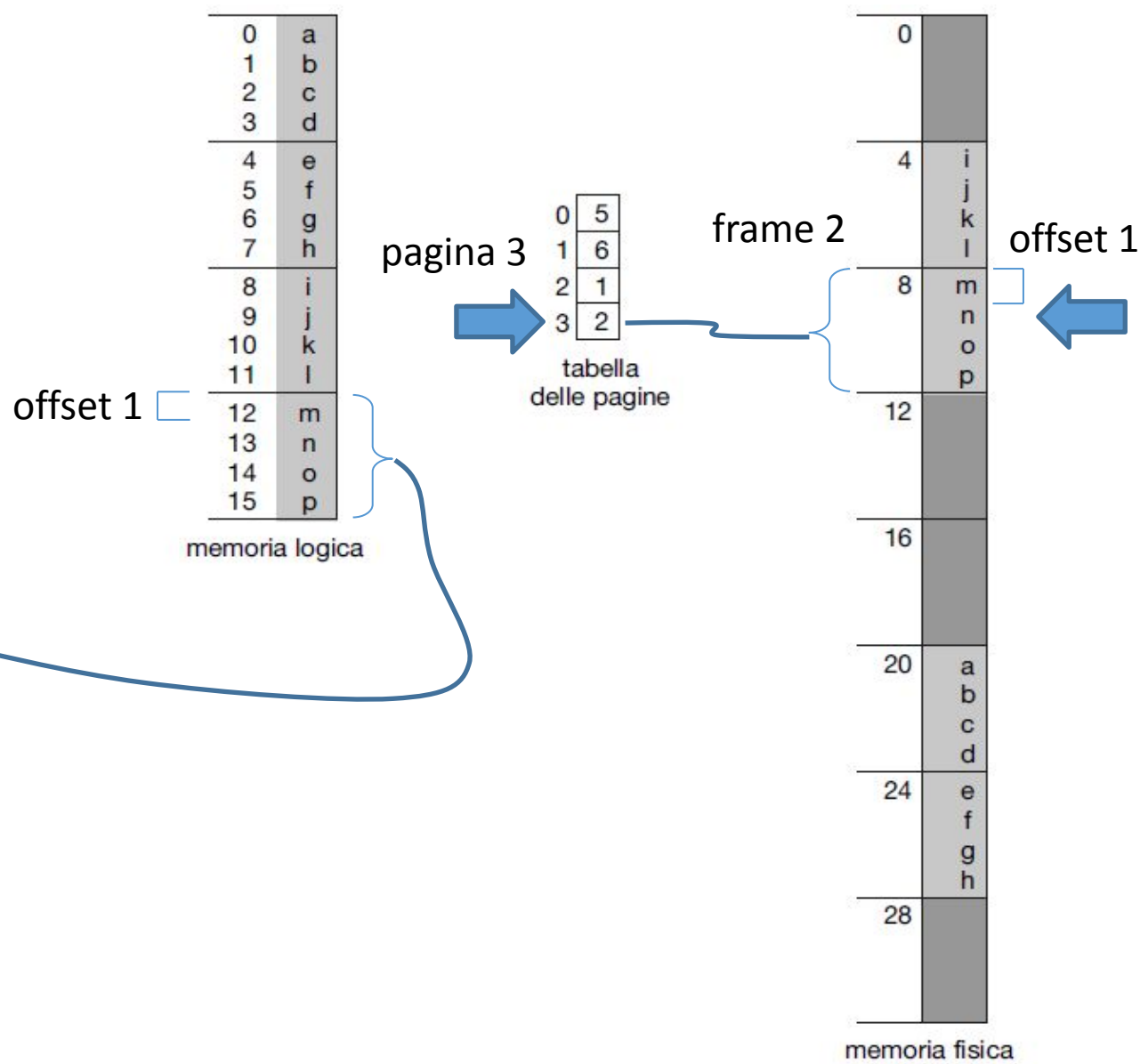


Figura 9.10 Esempio di paginazione per una memoria di 32 byte con pagine di 4 byte.

# Esercizio 11

---

Sia dato un sistema di paginazione con una tabella delle pagine che risieda in memoria

Se per un accesso in memoria occorrono 200 nanosecondi, quanto tempo occorrerà per ottenere il dato relativo a un indirizzo logico?

# Soluzione Esercizio 11

---

Tempo di accesso =  $200 + 200 = 400$  ns

Per ottenere il numero del frame relativo ad un indirizzo logico occorreranno 400 nanosecondi, poichè occorrono 200 ns per accedere alla tabella delle pagine in RAM e 200 ns per accedere al dato in memoria

# Esercizio 12

---

Sia dato un sistema di paginazione con una tabella delle pagine che risieda in memoria, avente un TLB con un hit ratio del 90%

Se per un accesso in memoria occorrono 200 nanosecondi, quanto tempo occorrerà per ottenere il dato relativo a un indirizzo logico?

Si supponga per semplicità che cercare una entry nella tabella delle pagine richieda tempo pari a 0 ns

# Soluzione Esercizio 12

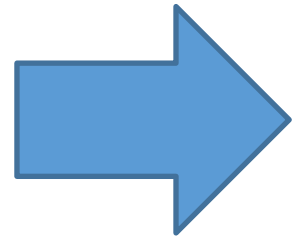
---

Caso 1: il numero di pagina desiderato si trova nel TLB

Tempo di accesso =  $0 + 200 = 200$  ns

Caso 2: il numero di pagina desiderato NON si trova nel TLB,  
sono quindi necessari 2 accessi in memoria

Tempo di accesso =  $200 + 200 = 400$  ns





# Soluzione Esercizio 12

---

Il tempo effettivo di accesso alla memoria sarà quindi

tempo effettivo

$$\text{di accesso} = 0.9 \times 200\text{ns} + (1-0.9) \times 400\text{ns} = 220\text{ns}$$

# Esercizio 13

---

Si consideri un sistema a memoria virtuale con

- indirizzi virtuali a 32 bit
  - indirizzi fisici a 24 bit
  - pagine di 4K byte
- a) di quante pagine sono costituiti rispettivamente lo spazio di indirizzamento virtuale e quello fisico?
- b) di quanti bit è costituito l'offset?
- c) di quante righe è costituita la tavola delle pagine?
- d) quante pagine occupa un processo da 9734 byte?

# Soluzione Esercizio 13

---

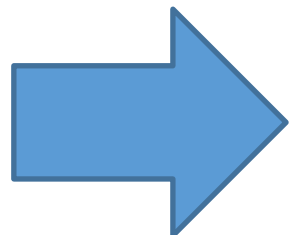
a) di quante pagine sono costituiti rispettivamente lo spazio di indirizzamento virtuale e quello fisico?

- Essendo gli indirizzi virtuali a 32 bit lo spazio virtuale è costituito da  $2^{32}$  byte
- Le pagine sono da 4 K byte =  $2^{12}$  byte, pertanto il numero di pagine virtuali è di:

$$2^{32}/2^{12} = 2^{20} = 1 \text{ M pagine}$$

- Essendo gli indirizzi fisici a 24 bit lo spazio fisico è costituito da  $2^{24}$  byte
- Le pagine sono da 4 K byte =  $2^{12}$  byte, pertanto il numero di pagine fisico è di:

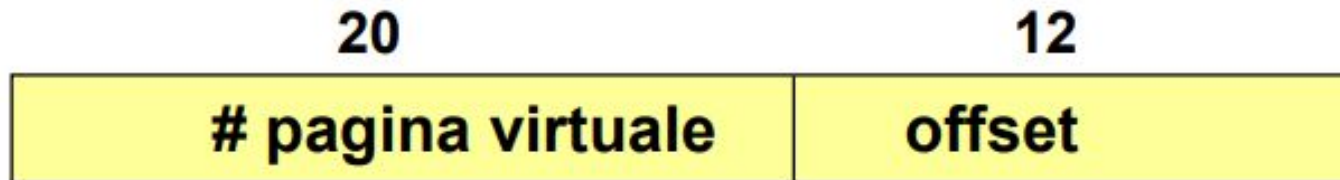
$$2^{24}/2^{12} = 2^{12} = 4 \text{ K pagine}$$



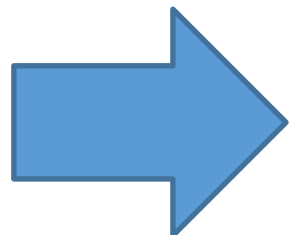
# Soluzione Esercizio 13

b) di quanti bit è costituito l'offset?

- L'offset dipende solo dalla dimensione delle pagine
- Dato che le pagine sono da 4 K byte =  $2^{12}$  byte, l'offset è di 12 bit
- Struttura dell'indirizzo virtuale:



- Struttura dell'indirizzo fisico

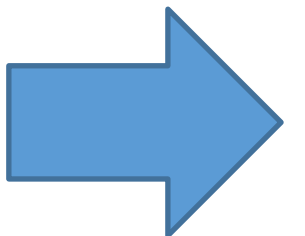


# Soluzione Esercizio 13

---

c) di quante righe è costituita la tavola delle pagine?

- La tavola delle pagine ha un elemento per ogni pagina dello spazio di indirizzamento virtuale. Le pagine virtuali sono  $2^{20}$



# Soluzione Esercizio 13

---

d) quante pagine occupa un processo da 9734 byte?

- Il processo viene allocato in pagine da 4 Kbyte = 4096 byte.
- Il numero di pagine occupate dal processo è dato da:

$$P = \lceil 9734/4096 \rceil = 3$$

- L'ultima pagina è solo parzialmente occupata. C'è uno sfrido pari a:

$$S = 3 \cdot 4096 - 9734 = 2554 \quad (\text{pari al } 2554/4096 \cong 62\%)$$

- Questo spreco corrisponde al fenomeno della *frammentazione interna*



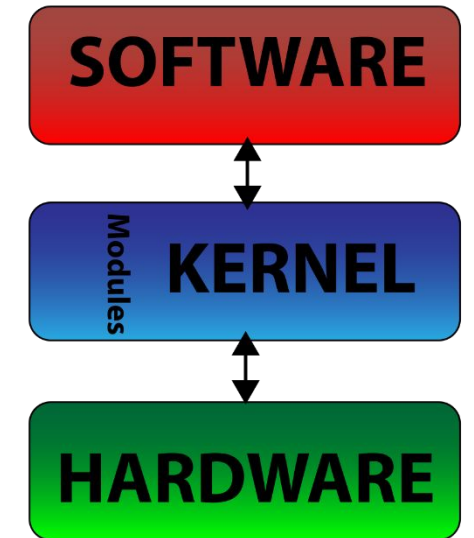
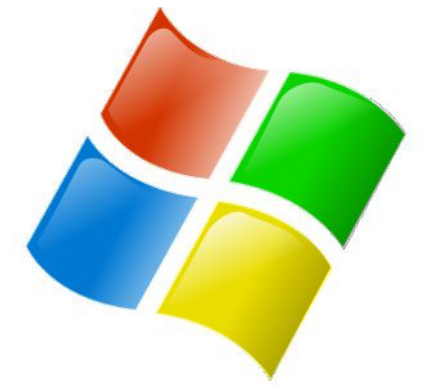


**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Sistemi Operativi*

# Esercitazione

## Memoria centrale



Docente:  
Domenico Daniele  
Bloisi

