

Università di Roma Tor Vergata
Corso di Laurea triennale in Informatica
Sistemi operativi e reti
A.A. 2016-17

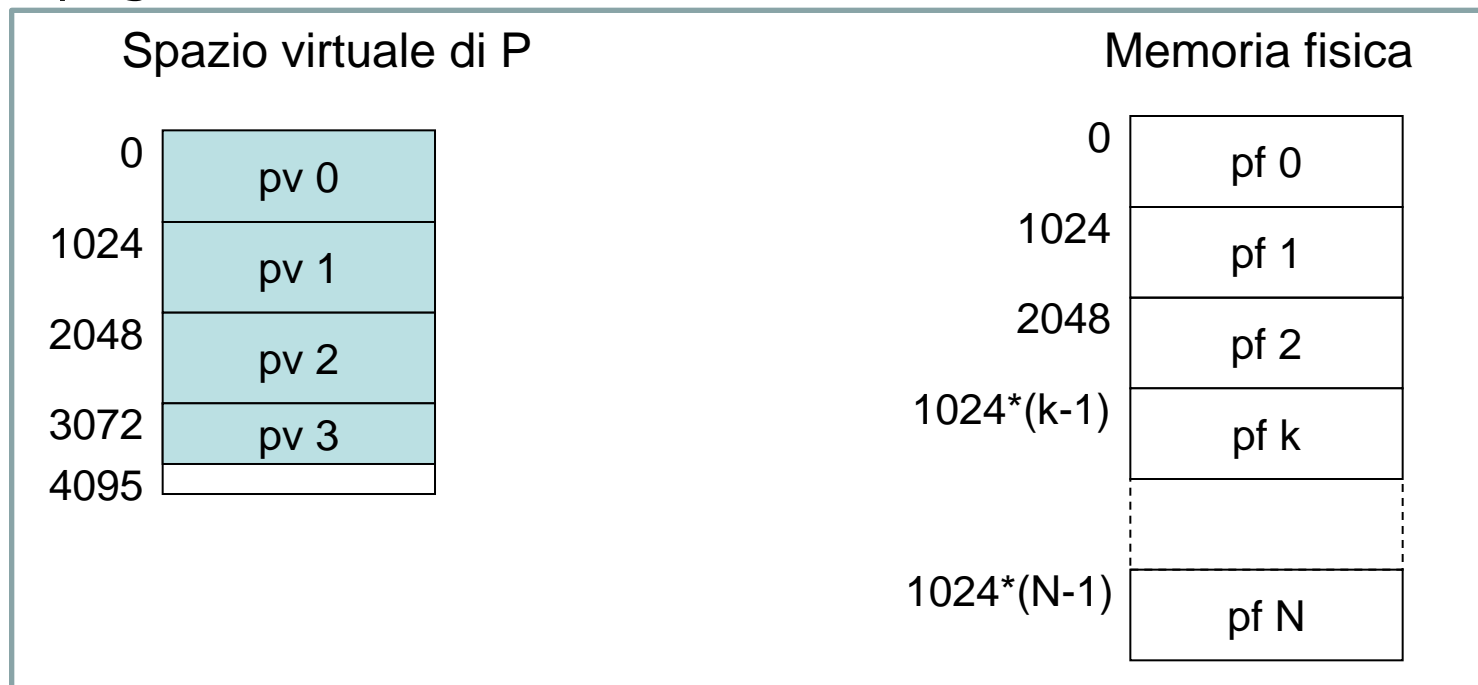
Pietro Frasca

Lezione 13

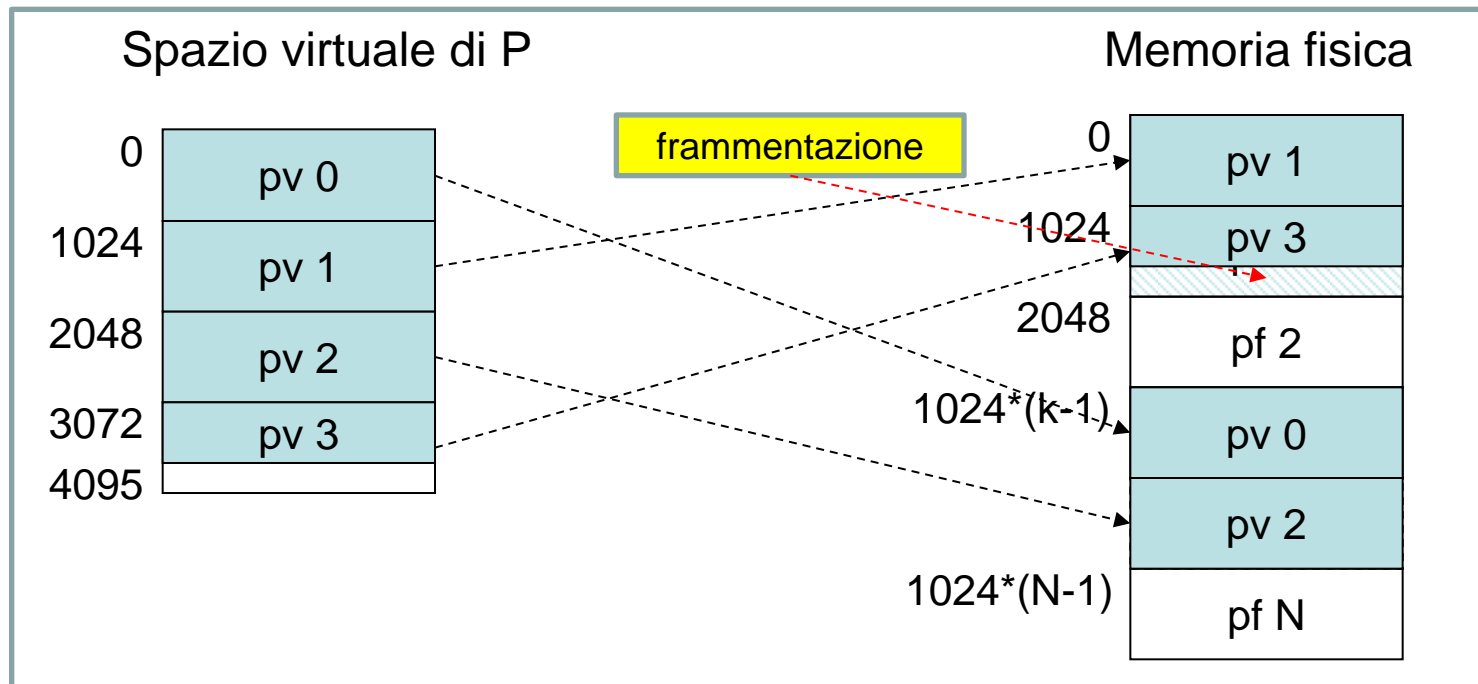
Giovedì 24-11-2016

Memoria paginata

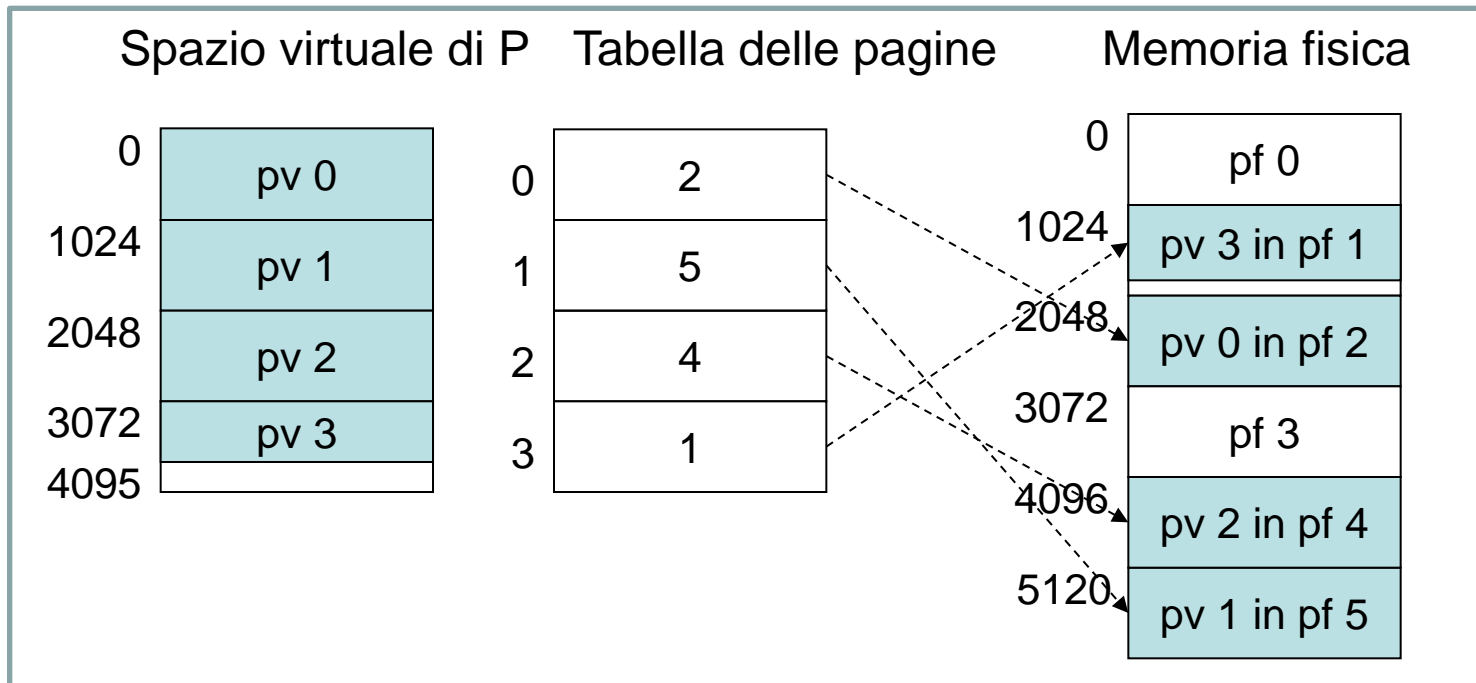
- La gestione della memoria fisica mediante partizioni ha lo svantaggio di produrre la frammentazione della memoria.
- La tecnica della **memoria paginata** consiste nel suddividere lo spazio virtuale del processo in blocchi di dimensione fissa dette **pagine virtuali**. Le pagine hanno dimensione di **potenza di 2** (esempio $2^{10}=1024$).
- Anche la memoria fisica è partizionata in **pagine fisiche** (dette anche **frame**) della **stessa dimensione** delle pagine virtuali.



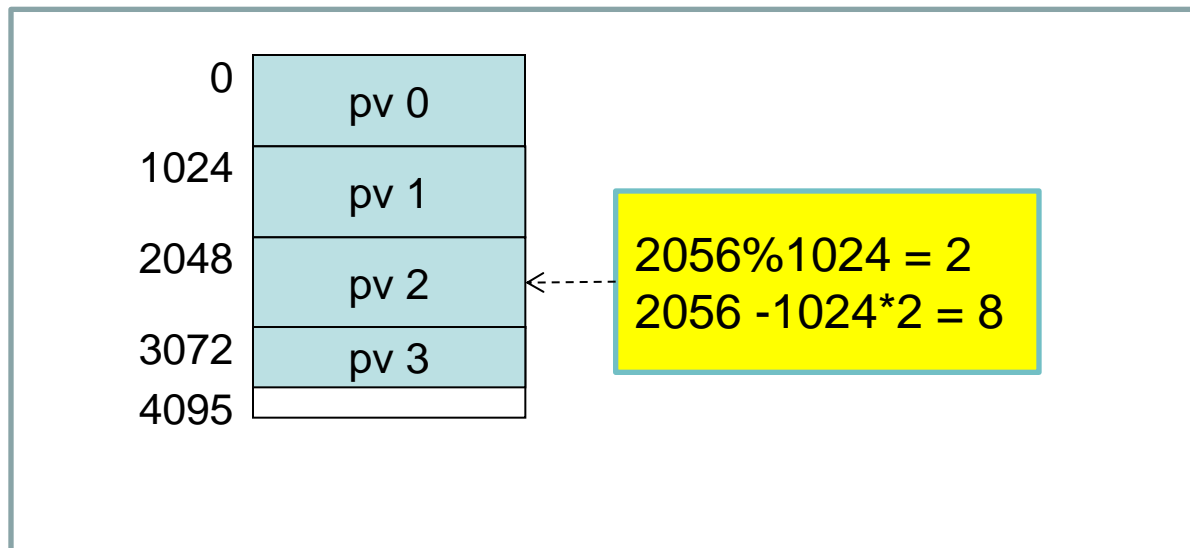
- In tal modo ciascuna pagina virtuale di un processo può essere allocata in memoria in qualsiasi pagina fisica e in modo indipendente.
- La pagina fisica, allocata per l'ultima pagina virtuale del processo potrebbe non essere completamente utilizzata. Infatti è molto difficile che la dimensione dello spazio virtuale di un processo sia un multiplo intero della dimensione della pagina. In tal caso si produce frammentazione interna.



- Per **ogni processo**, il kernel, gestisce una **tabella delle pagine** che realizza la funzione di traduzione degli indirizzi tra le pagine virtuali e le pagine fisiche nelle quali le pagine virtuali sono caricate.
- L'indirizzo della tabella delle pagine e il numero di pagine di cui è costituito lo spazio virtuale di un processo sono memorizzate in campi del descrittore del processo.



- La dimensione della tabella delle pagine dipende dalla dimensione dello spazio virtuale del processo e dalla dimensione delle pagine.
- In pratica, per tradurre un indirizzo virtuale **IV** è necessario sapere a quale pagina virtuale esso appartiene. Dividendo l'indirizzo virtuale per la dimensione **D** della pagina si ha che:
 - **Il quoziente** della divisione è **l'indice** della pagina virtuale;
 - **Il resto** della divisione è **l'offset** dell'indirizzo all'interno della pagina.
- Ad esempio se la pagina è di dimensioni di 1024 byte, **l'indirizzo 2056** appartiene alla **pagina 2** ed ha **offset pari a 8 (la numerazione delle pagine inizia da 0)**.

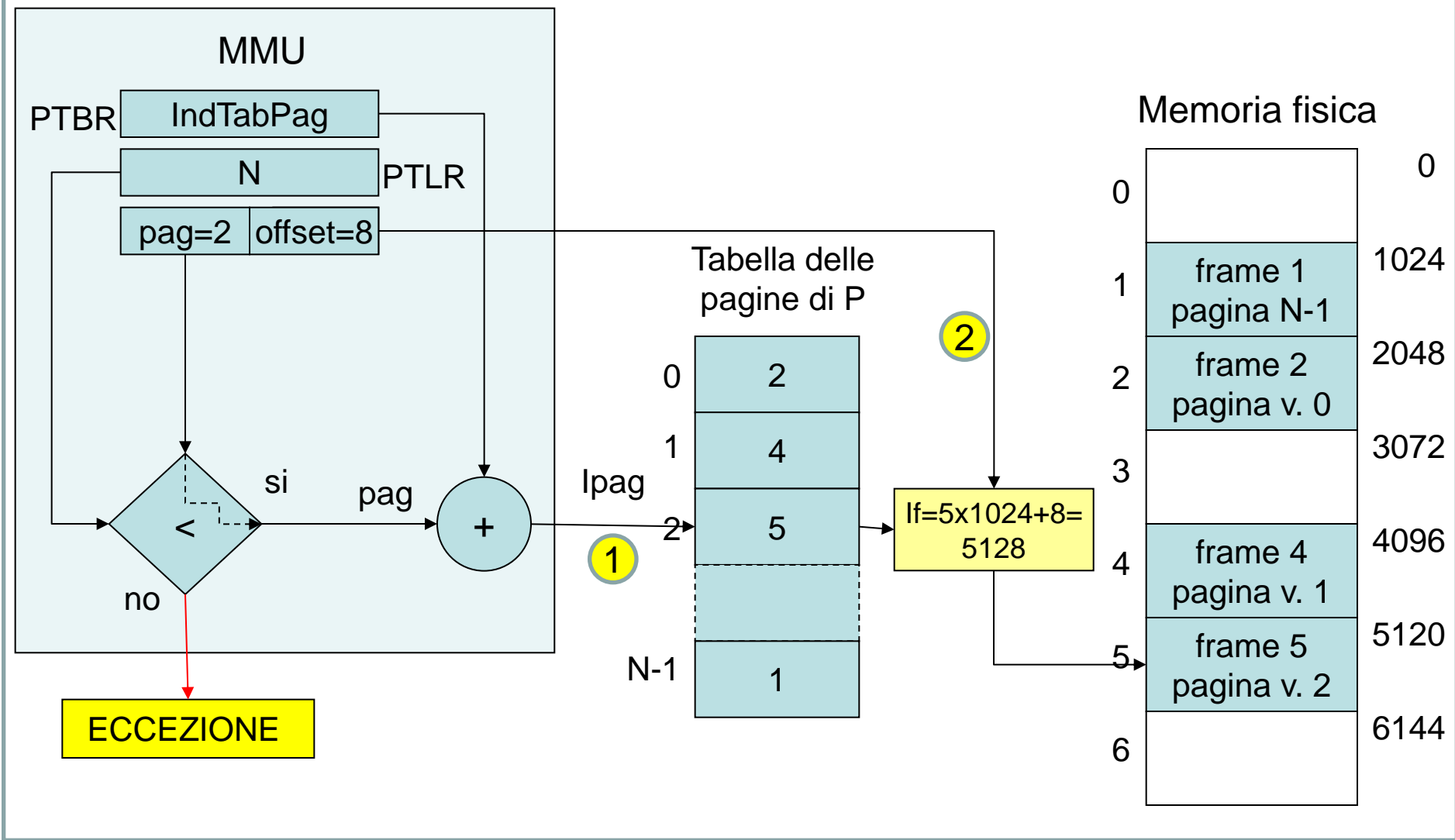


- Per mantenere lo stato di allocazione della memoria fisica, il gestore della memoria mantiene aggiornata una tabella, detta **tabella delle pagine fisiche**, che ha un numero di righe pari al numero delle pagine fisiche in cui è suddivisa la memoria.
- L'elemento della tabella indica se la pagina fisica è libera o occupata. Se la pagina è occupata l'elemento contiene due campi: **il pid del processo** a cui appartiene la pagina e **l'indice pv** della pagina virtuale corrispondente.

	pid	pagina
0	P1	1
1	P2	6
2	free	
3	P3	4
k	free	
N-1	P2	10

Tabella delle pagine fisiche

- L'indirizzo fisico della tabella delle pagine del processo in esecuzione viene recuperato dal suo descrittore e caricato nel registro PTBR della MMU (**Registro Puntatore alla Tabella delle Pagine**).
- Per ridurre il numero di accessi alla tabella delle pagine, come per la memoria segmentata, nella MMU è presente una **TLB (Translation Lookaside Buffer)** contenente un numero limitato di registri associativi che risolvono la corrispondenza tra pagine virtuali e pagine fisiche relative agli ultimi riferimenti.
- Quando la cpu genera un indirizzo virtuale se nell'MMU sono già presenti i dati per la traduzione dell'indirizzo non è necessario accedere alla tabella in memoria. Altrimenti, la traduzione avviene in base allo schema mostrato nella figura seguente, e alla fine della traduzione la coppia **(pv,pf)** viene salvata in un registro della TLB. In genere il buffer è gestito in modo circolare e quindi vengono sovrascritti continuamente i registri, in modalità FIFO.



Traduzione degli indirizzi nella memoria paginata

- Come già descritto per la segmentazione, per accedere ad una locazione di memoria sono ora **necessari due accessi**: il primo accesso alla tabella delle pagine e il secondo alla locazione vera e propria.
- Quando non sono più disponibili pagine fisiche libere (o sono poche), lo scheduler a medio termine che esegue la funzione di `swap_out`, libera un numero di pagine necessarie per il processo in esecuzione, revocandole a uno o più processi.
- il **descrittore del processo** contiene, per quanto riguarda la memoria paginata, due campi:
 - **l'indirizzo della tabella delle pagine**
 - **il numero di pagine virtuali** di cui è costituito il suo spazio virtuale.
- Un parametro importante è dato dalla **dimensione** delle pagine. Più è piccola la dimensione delle pagine più sono grandi le dimensioni delle **tabelle delle pagine dei processi**.

- Se la dimensione della pagina è troppo grande si verifica una frammentazione interna, dato che difficilmente la dimensione di uno spazio virtuale è un multiplo intero della dimensione delle pagine.
- Tipici valori della dimensione delle pagine sono compresi tra 512 Byte e 16MB.
- Per quanto riguarda la **protezione** e la **condivisione**, restano validi i concetti descritti per la segmentazione, anche se, nella paginazione c'è una corrispondenza semantica tra moduli logici del programma e **gruppi** di pagine, dato che un modulo logico può essere composto da varie pagine.

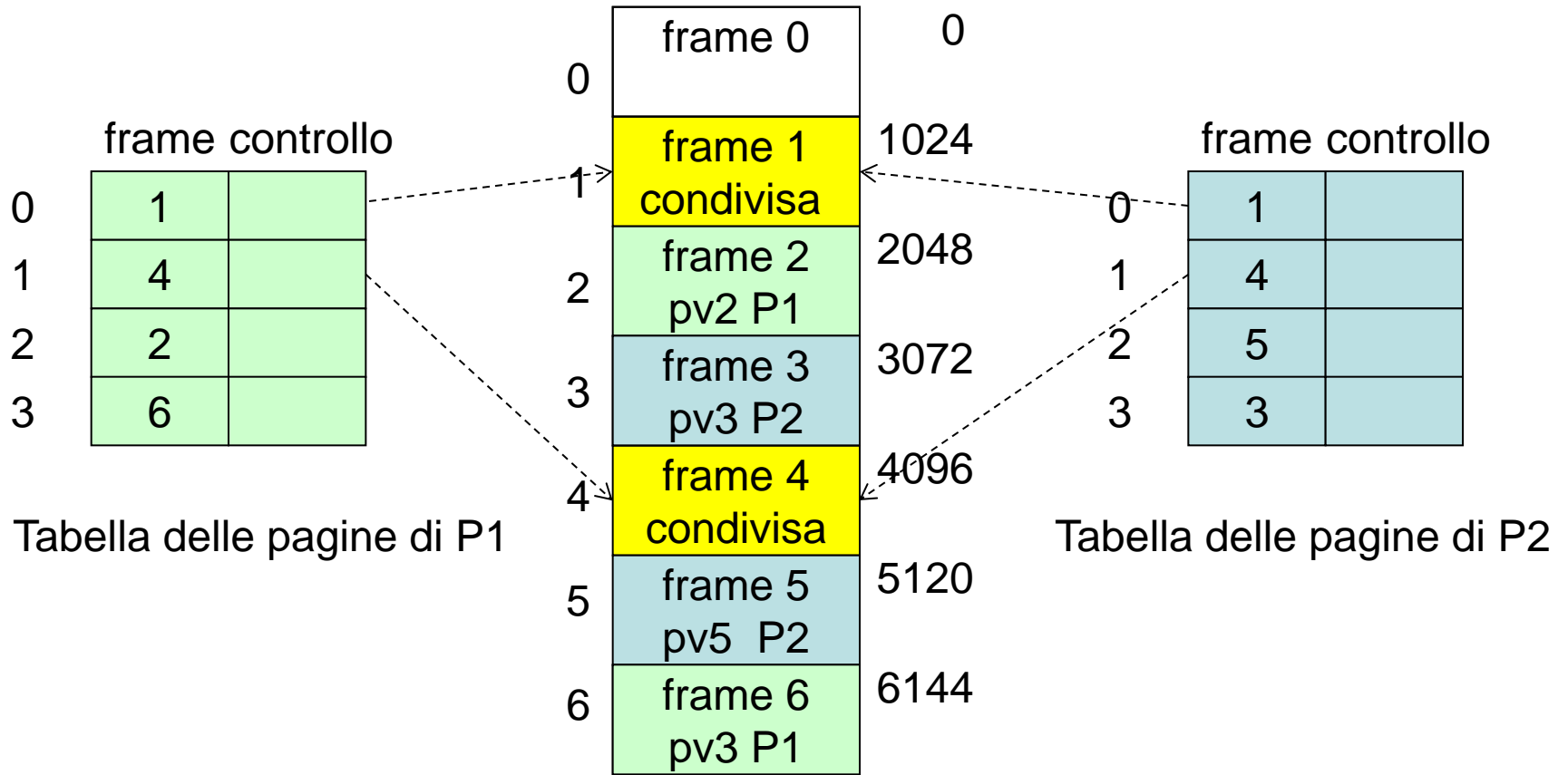
- E' possibile associare ad ogni pagina virtuale specifici diritti di accesso, associando ad ogni elemento della *tabella delle pagine* un campo di controllo in modo analogo a quanto visto per la segmentazione.

campo pagina fisica		Controllo	
Indice pagina fisica in cui è stata allocata la p.v.		R	W

Descrittore della tabella delle pagine

- Le stesse considerazioni possono essere fatte per la condivisione. In questo caso è possibile consentire a più processi di condividere delle pagine.

Memoria fisica



Condivisione di pagine

Segmentazione su richiesta

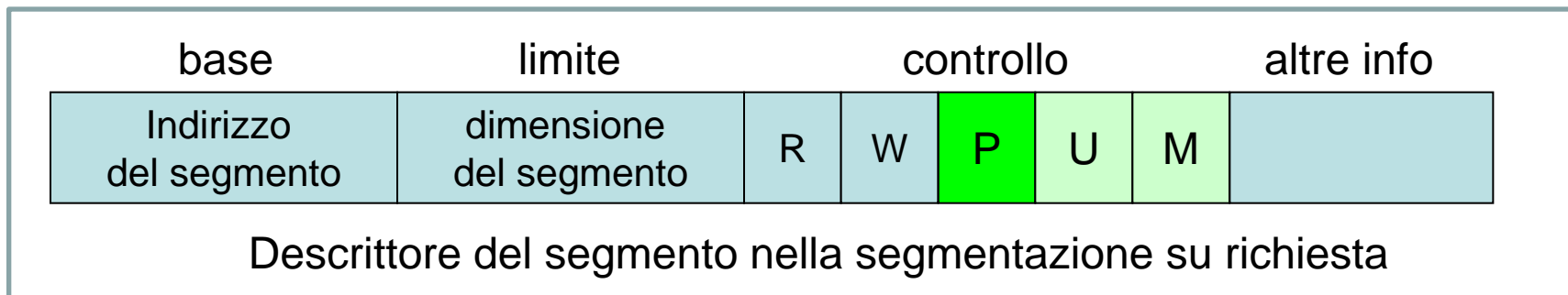
- Per evitare i problemi di memoria dovuti alla modalità di caricare interamente lo spazio virtuale di un processo, si ricorre ad una più complessa funzione di rilocazione **$If=f(Iv)$** in modo tale che, durante l'esecuzione di un processo, sia possibile mantenere in memoria soltanto la parte del suo spazio virtuale che in quel momento è assolutamente necessario. Per realizzare questo funzionamento è necessario che la funzione di rilocazione elabori l'indirizzo virtuale

$Iv = \langle seg, offset \rangle$

in modo tale che:

- se il segmento **seg** è presente in memoria calcoli il corrispondente indirizzo fisico **$If=f(Iv)$** ;
- altrimenti se non è presente in memoria generi un segnale di interruzione , detto **segment-fault**.

- La funzione che gestisce il segment-fault ha il compito di cercare un segmento libero in memoria fisica e caricarvi il corrispondente segmento virtuale e riattivare l'esecuzione del processo. E' necessario modificare la struttura del descrittore del segmento:



- Oltre ai bit di controllo R e W sono ora presenti altri flag di controllo:
 - Presenza (P)**
 se **P=1** indica che il segmento si trova in memoria fisica e i campi base e limite specificano la partizione di memoria fisica in cui il segmento è caricato.
 Se **P=0** il segmento non si trova in memoria fisica, nel campo **altre info** è memorizzato l'indirizzo del settore su disco da dove inizia la memorizzazione del segmento virtuale. In questo caso si genera **un'interruzione di segment-fault**.

- Se non è disponibile sufficiente quantità di memoria fisica per allocare il segmento che contiene l'indirizzo virtuale che ha generato il segment-fault, si procede all'operazione di **rimpiazzamento** che consiste nel selezionare dalla memoria ram qualche segmento, appartenente o meno al processo attuale, e portarlo sulla swap-area, liberando memoria fisica.
- Per rendere più efficiente l'implementazione di algoritmi di rimpiazzamento dei segmenti, nel descrittore del segmento sono aggiunti i bit di controllo indicati con il nome di bit d'**uso (U)** e **modifica (M)**.

base	limite	controllo					altre info
Indirizzo del segmento	dimensione del segmento	R	W	P	U	M	

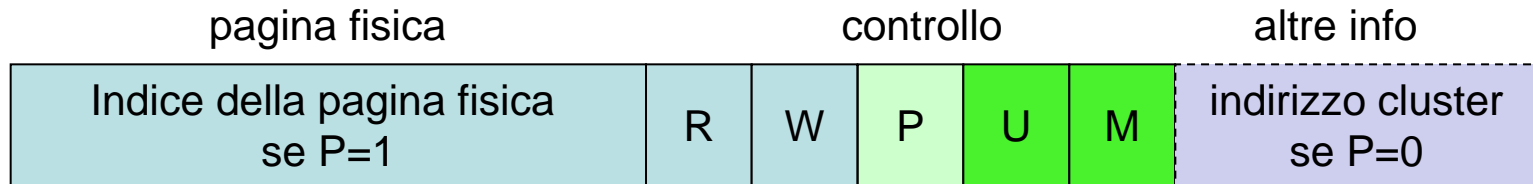
Descrittore del segmento nella segmentazione su richiesta

- Il bit **U** (uso) e **M** (modifica) sono **settiati via hardware**. Il bit **U** è posto a **1** ogni volta che si riferenzia il segmento e il bit di modifica **M** detto **dirty bit**, è settato quando si modifica il contenuto del segmento.
- I bit **U** e **M** possono essere letti e resettati via software.
- La presenza del bit M di modifica consente di scegliere come segmento da rimpiazzare un segmento che ha il bit **M=0**, in quanto tale segmento non è stato modificato dopo che è stato allocato in memoria e quindi la partizione che lo contiene può essere liberata senza dover riscrivere il contenuto del segmento nella swap-area.
- Il bit d'uso **U** è usato per verificare se un segmento è stato indirizzato dal momento in cui il bit **U** è stato azzerato. Ciò consente di valutare la **frequenza d'uso** di ogni segmento e quindi di implementare algoritmi di scelta del segmento da rimpiazzare sulla base della frequenza d'uso dei segmenti presenti in memoria.

Paginazione su richiesta

- Con la tecnica della paginazione un processo deve essere caricato completamente in memoria per essere eseguito.
- Come già descritto per la segmentazione su richiesta, anche per la paginazione, ricorrendo ad architetture di microprocessori più complesse, si può fare in modo che un processo possa essere eseguito caricando in memoria solo una parte del suo spazio virtuale.
- Nel campo controllo dell'elemento della ***tabella delle pagine*** è inserito un bit di presenza **P** che indica se la pagina virtuale relativa è presente o meno in memoria. Analogamente sono usati il bit d'uso **U** e di modifica **M**.

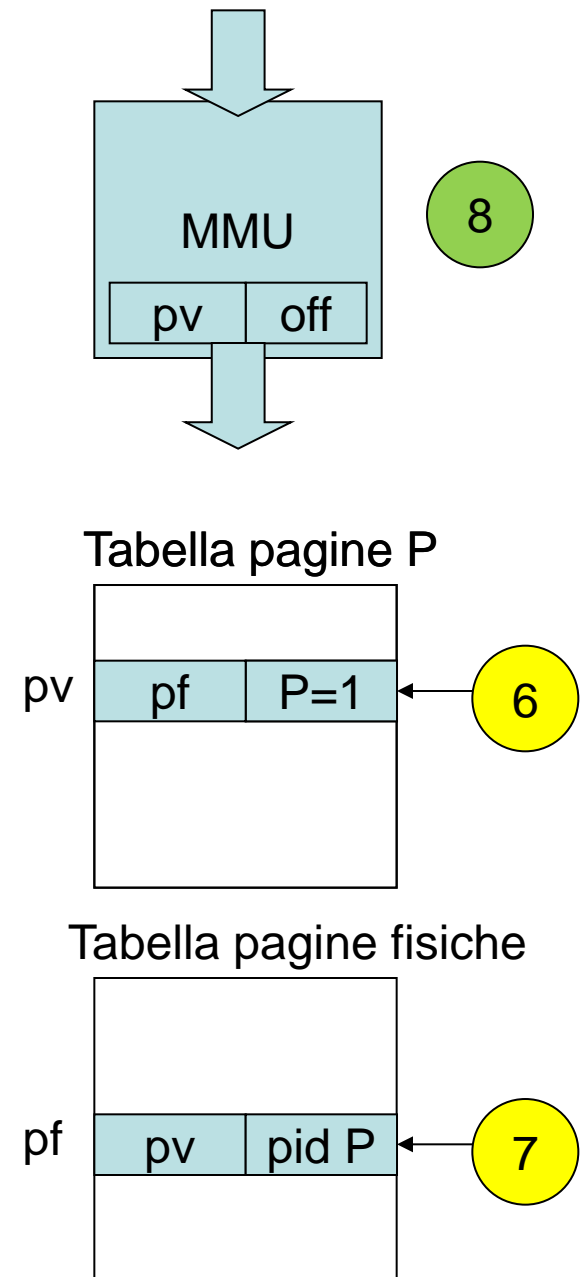
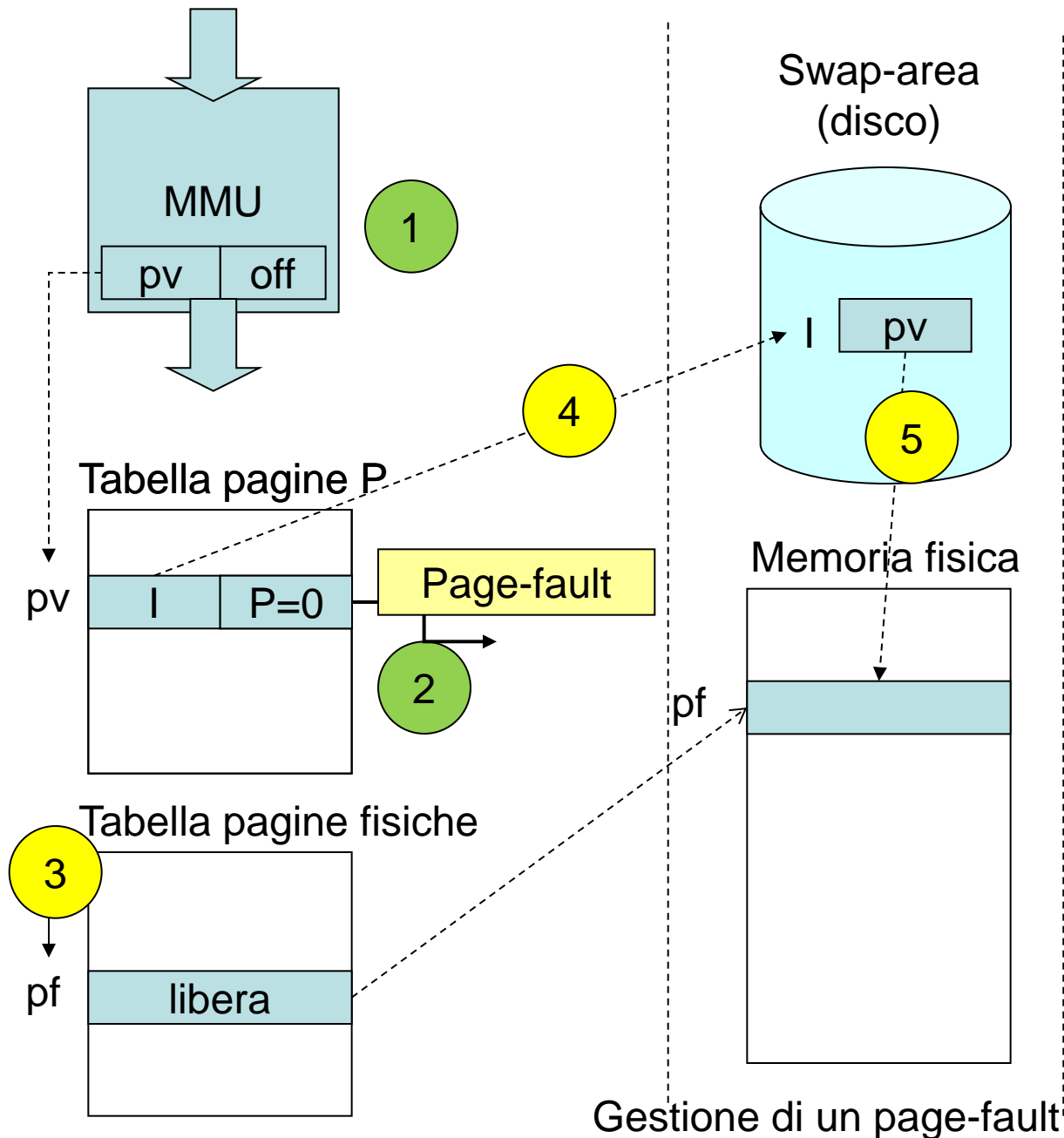
- L'elemento della tabella delle pagine dipende dall'architettura hardware. Tipicamente ha il formato come in figura:



Elemento della tabella delle pagine

- Generalmente, il campo altre info non è presente nella tabella delle pagine ma è memorizzato in tabelle esterne, associate al descrittore del processo.
- Il disco di sistema è formattato in blocchi logici (cluster) di dimensione pari alla dimensione delle pagine.
- Quando un processo è creato, alcune pagine del suo spazio virtuale sono copiate in altrettanti cluster della swap-area (o del file di paging) e sono caricate in memoria solo poche pagine (pre-paging) o anche nessuna.

- La **tabella delle pagine** è creata nel modo seguente:
 - ponendo a 1 i bit P di presenza delle pagine caricate e a 0 quelle delle pagine non caricate;
 - Inserendo nel campo **pagina fisica** l'indice della pagina fisica se $P=1$ o nel campo **altre info** l'indirizzo della pagina su swap-area (indirizzo del cluster) delle pagine non caricate ($P=0$).
- Quando la MMU indirizza una pagina che non è caricata in memoria (**bit $P=0$**) essa genera un segnale di interruzione di **page-fault (errore di pagina)** che attiva una funzione del gestore della memoria che ha il compito di ricercare nella **tabella delle pagine fisiche**, una pagina libera in cui caricare la pagina virtuale richiesta.
- Se non è disponibile memoria libera, è necessario liberare la memoria salvando sull'area swap un certo numero di pagine, appartenenti o meno al processo in esecuzione. Questa operazione prende il nome di **rimpiazzamento delle pagine**.

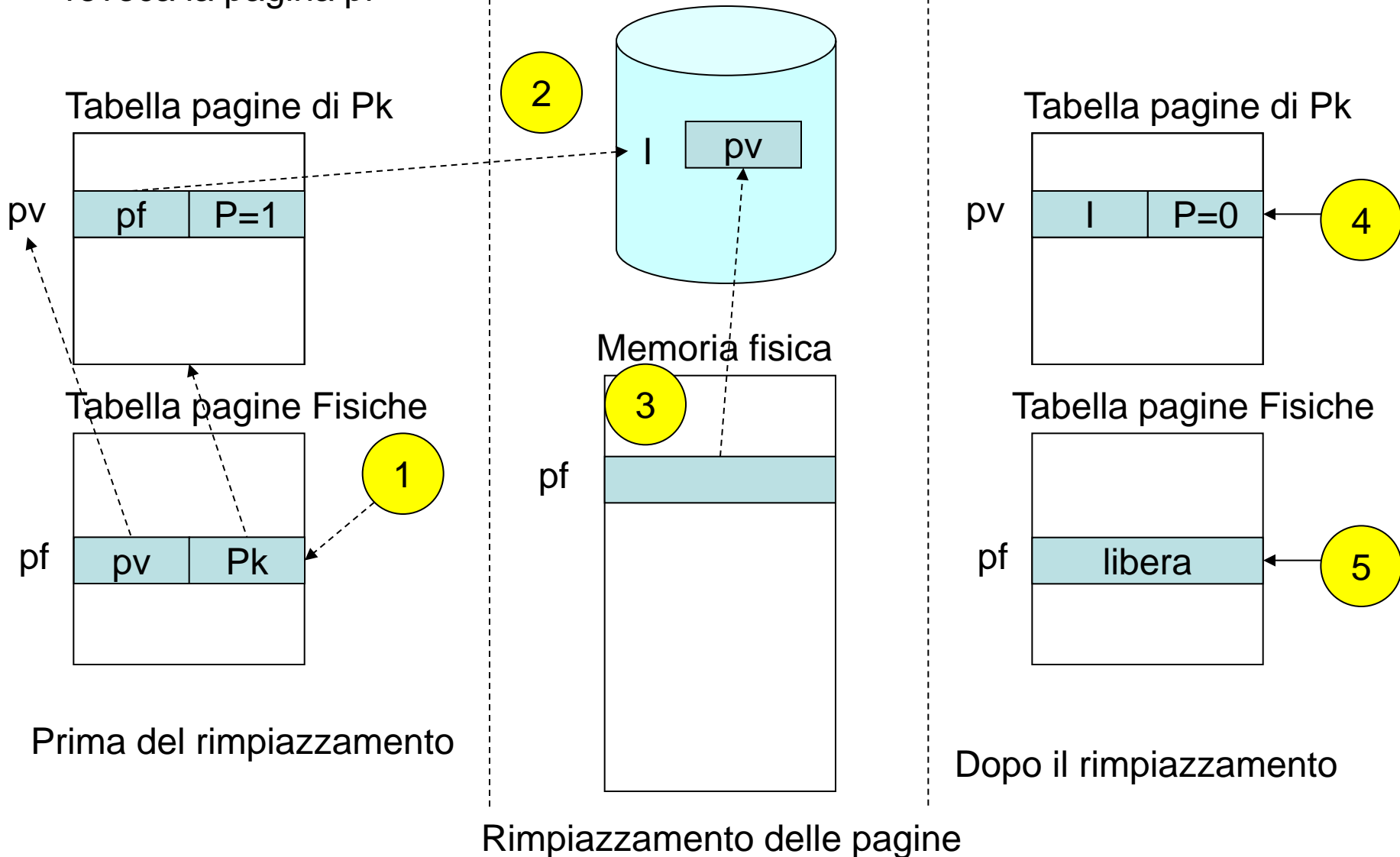


Gestione di un page-fault

Rimpiazzamento delle pagine

- Per effettuare il rimpiazzamento di una pagina fisica di indice **pf** il kernel esegue i seguenti passi:
 1. mediante un algoritmo di rimpiazzamento, ricerca nella tabella delle pagine fisiche la **pf** più idonea da rimpiazzare;
 2. trovata la pagina **pf**, recupera l'indice **pv** della pagina virtuale che vi è allocata e il processo **Pk** cui essa appartiene; seleziona l'elemento di indice **pv** della tabella delle pagine di **Pk** e ottiene l'indirizzo **I** del cluster su swap-area;
 3. salva la pagina fisica **pf** contenente la pagina virtuale **pv** di **Pk** su swap-area all'indirizzo **I**;
 4. aggiorna i valori dell'elemento **pv** della tabella delle pagine del processo **Pk** ponendo a **0** (reset) il bit di presenza **P**;
 5. aggiorna la tabella delle pagine fisiche rendendo **pf** libera.
- La copia su disco della pagina scelta non avviene se il bit di modifica è $M=0$.
- Al termine del rimpiazzamento il processo torna in esecuzione ed esegue di nuovo l'istruzione che ha generato il page-fault.

Pk: processo a cui si revoca la pagina pf



- L'algoritmo di rimpiazzamento, deve escludere dalla selezione le **pagine fisiche che sono coinvolte in un trasferimento dati**. Ad esempio se una pagina fisica **pf** contiene la pagina virtuale **pv** di un processo **P** che ha attivato un trasferimento di dati da una periferica a un suo buffer contenuto nella propria pagina virtuale **pv**, tramite un canale DMA, **pf** non deve essere scelta.
- Infatti, anche se il processo **P** è bloccato, il DMA trasferisce dati nel buffer e se **pf** viene sostituita, il DMA scriverà dati in una pagina virtuale diversa.
- Per questo motivo l'elemento della tabella delle pagine fisiche contiene anche un **campo di lock** che **se posto ad 1** significa che la corrispondente pagina fisica deve essere ignorata dall'algoritmo di rimpiazzamento. Il blocco di una pagina spesso è chiamato pinning (appuntare).
- Un'altra soluzione è di gestire tutte le operazioni di I/O nei buffer del kernel e successivamente copiare il dati nelle pagine del processo utente.

- I page-fault riducono le prestazioni del sistema. E' quindi necessario **minimizzare il numero di page-fault** che un processo può generare durante la sua esecuzione.
- Il numero di page-fault, in parte, dipende da come è strutturato il programma che il processo esegue. Tanto più il programma è ben strutturato, tanto minore sarà il numero di page-fault che genererà.
- Quando ci sono molti processi il numero di page-fault aumenta notevolmente. Diventa fondamentale la scelta dell'algoritmo di rimpiazzamento per evitare il fenomeno del **trashing** che consiste nell'attività, da parte della CPU, a trasferire continuamente pagine tra la memoria fisica e la swap-area, riducendo fortemente l'efficienza del sistema.