

Sistemi Operativi

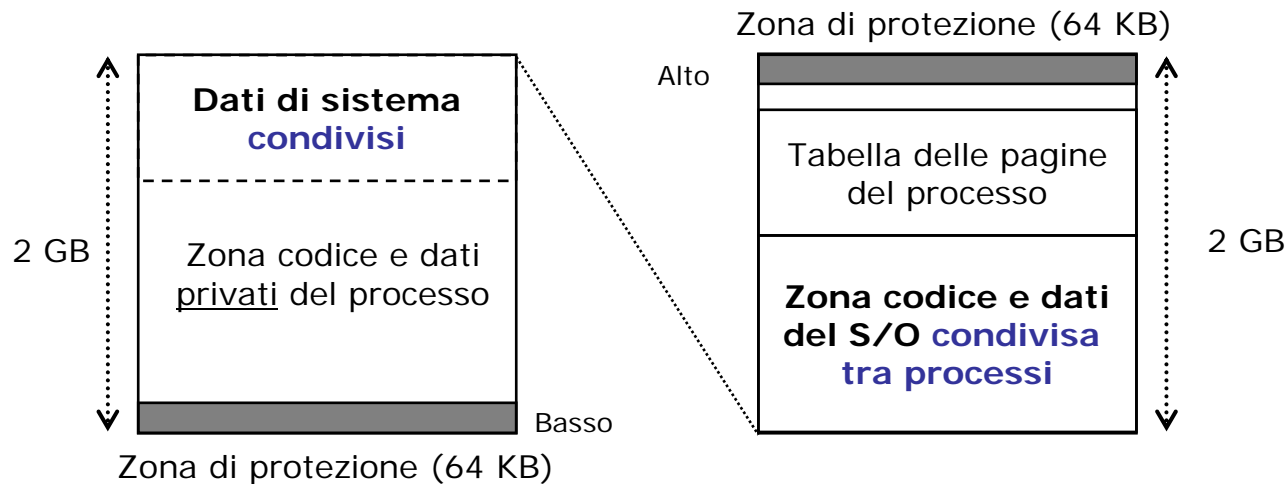
Il Sistema Operativo Windows (parte 2)

Docente: Claudio E. Palazzi
cpalazzi@math.unipd.it

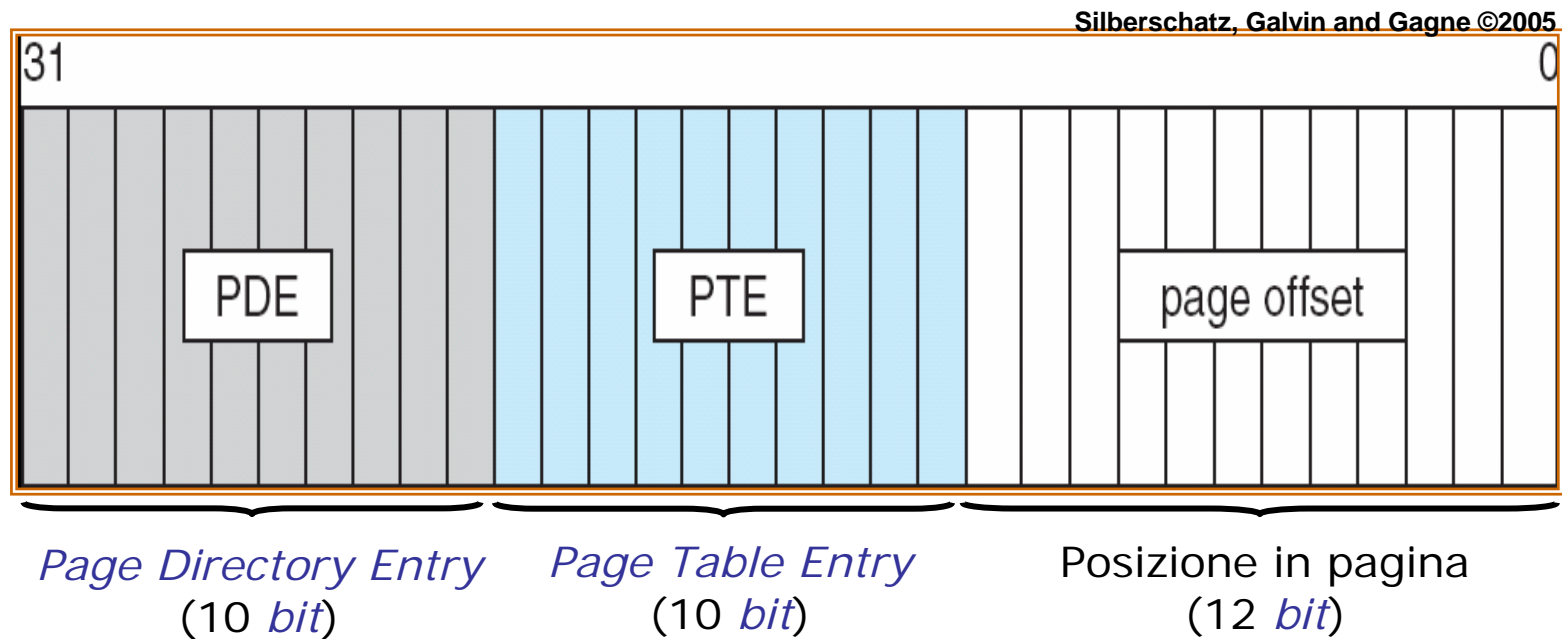
Crediti per queste slides al Prof. Tullio Vardanega

Gestione della memoria – 1

- Ogni processo dispone di uno spazio di indirizzamento virtuale **paginato** ampio 4 GB e suddiviso in 2 zone adiacenti ampie 2 GB ciascuna
 - Indirizzi virtuali espressi su 32 *bit*

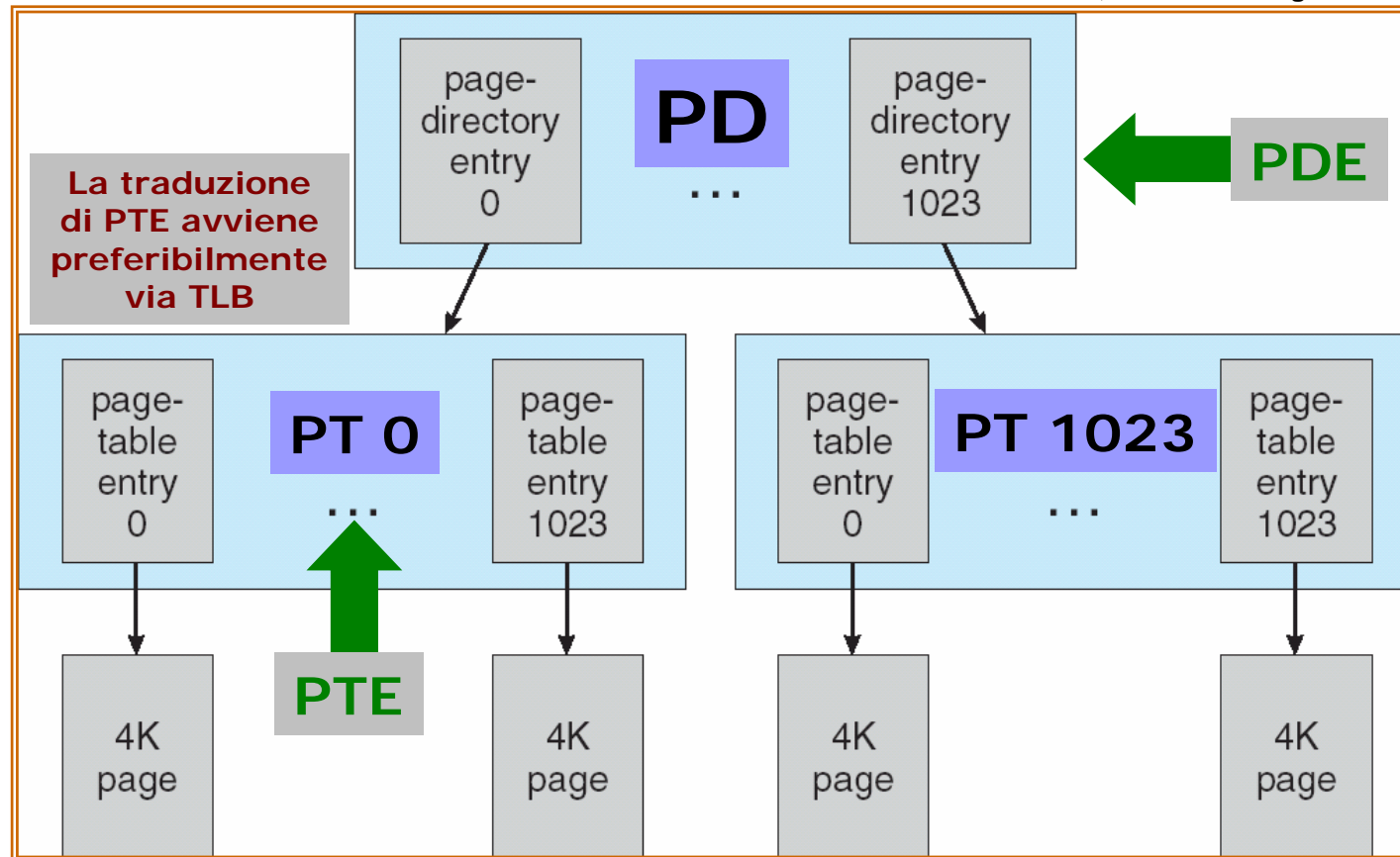


Indirizzamento virtuale – 1



Indirizzamento virtuale – 2

Silberschatz, Galvin and Gagne ©2005



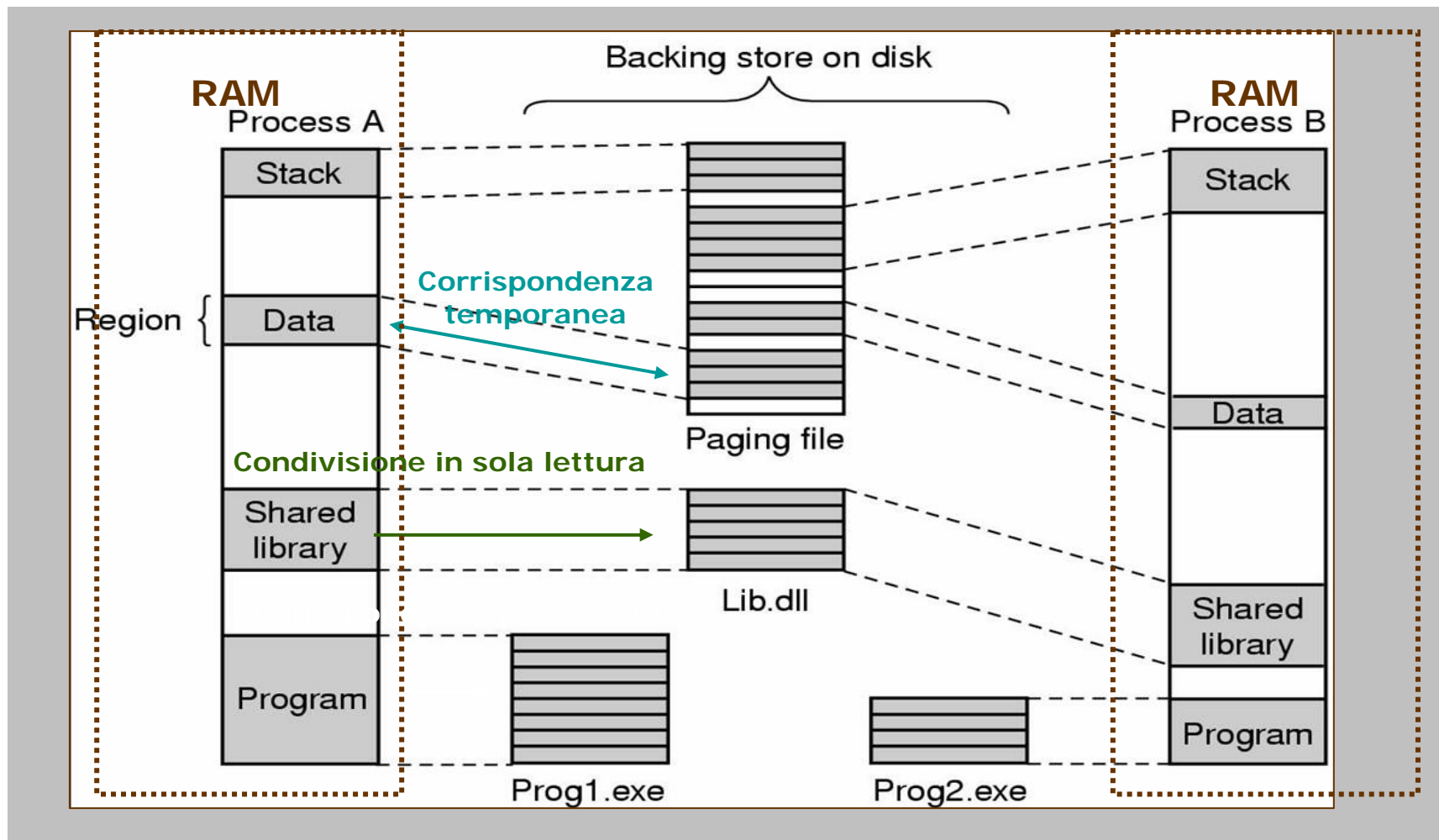
Gestione della memoria – 3

- Una pagina virtuale può essere
 - **R** (lettura) / **W** (scrittura) / **E** (esecuzione)
 - **Libera (*free*)**: non riferita da alcun PTE
 - Tutte le pagine di un processo sono inizialmente libere (*paging-on-demand*)
 - Page fault
 - **Assegnata (*committed*)**: in uso per codice o dati
 - Viene riferita tramite indirizzo virtuale e caricata da disco ove non fosse già presente in RAM
 - **Prenotata (*reserved*)**: non ancora in uso, ma **non libera**
 - Per agevolare l'assegnazione di pagine contigue a processi
 - Alla creazione di un nuovo processo 1MB è riservato per lo stack

Gestione della memoria – 4

- Più processi possono condividere l'accesso a pagine di uno stesso **file mappato in memoria**
 - Un libreria condivisa **DLL** (Dynamic Link Library) è un tipico esempio di **file** mappato in memoria
 - Codice **condiviso** in sola lettura
 - Dati statici R/W **copiati** per ciascun processo (*copy-on-write*)
 - Ogni processo che accede a un **file** possiede specifici diritti di accesso che il S/O si preoccupa di far rispettare
- La stessa posizione nel **file** può corrispondere ad indirizzi virtuali **diversi** per processi distinti
 - Gli indirizzi riferiti nel codice condiviso di **DLL** devono pertanto essere espressi in modo **relativo**
 - A cura del compilatore

Gestione della memoria – 5



Gestione della memoria – 6

- Il caricamento di una nuova pagina in RAM può richiedere il **rimpiazzo** locale di un pagina “vecchia”
 - Solo se non vi sono abbastanza pagine libere
 - Il sistema mantiene una lista delle pagine libere
 - A ogni processo i si associa l'insieme I_i delle sue pagine attualmente in RAM (**Working Set**)
 - L'ampiezza del WS I_i può variare solo entro limiti prefissati

Gestione della memoria – 7

- Anche il S/O stesso è visto come un processo con un proprio **WS** con pagine rimpiazzabili
 - Min set iniziale nell'ordine di 20-50 pagine
 - Max set iniziale nell'ordine di 45-345 pagine
 - Solo alcune pagine del S/O sono **inamovibili**
- Un *daemon* di **kernel** con periodo 1 s accerta che vi siano sufficienti pagine libere
 - *Balance set manager*
- Se insufficienti il *daemon* attiva un *thread* del **Memory manager** che esamina con una euristica i WS dei processi per rilasciarne pagine
 - *Working set manager*
 - Processi non recentemente attivi con WS ampi vengono scrutinati prima degli altri
 - Le pagine necessarie si prelevano dagli WS di ampiezza vicina al massimo e con scarso uso recente

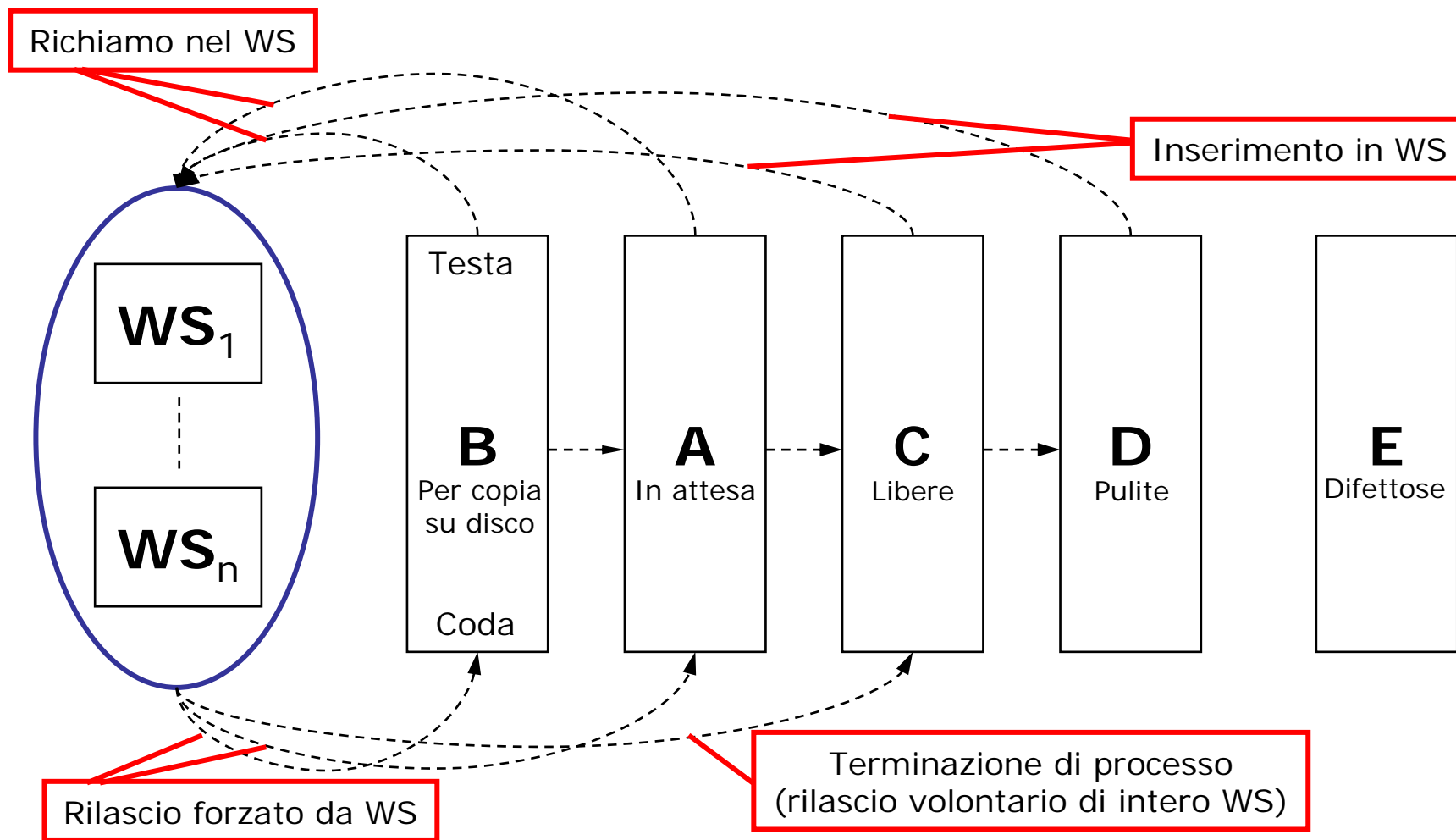
Gestione della memoria – 8

- Ciascuna *page frame* in RAM può essere
 - **In uso** e appartenere a 1 WS (≥ 1 se condivisa)
 - **Rilasciata** e appartenere a 1 e 1 sola lista tra:
 - [A] **In attesa**: pagina recentemente rimossa dal WS di un processo ma ancora associata a esso e **non** modificata
 - Può essere riassegnata e sovrascritta senza problemi
 - [B] **Da copiare su disco**: ~ A ma se rimpiazzata deve essere riportata su disco
 - [C] **Libera**: ~ A ma non più associata ad alcun processo
 - [D] **Azzerata**: ~ C ma con contenuto **obliterato** a zero per consentire riassegnazione **senza travaso di info privata**
 - [E] **Difettosa**: pagina che non può più essere utilizzata a causa di difetti nella zona di memoria fisica

Gestione della memoria – 9

- Lo **swapper thread** (*daemon*) del **Memory manager** porta in [A] o [B] le pagine dello *stack* dei processi i cui *thread* siano stati tutti recentemente inattivi
- Altri 2 *daemon* assicurano che vi siano abbastanza pagine in [C] salvando su disco quelle in [B] e poi accodandole in [A]
- Un WS che cresce preleva pagine libere da [C] se le sovrascrive interamente (senza conservare dati precedenti) da [D] altrimenti
 - Un *daemon* dedicato che opera per conto del **kernel** azzerava periodicamente il contenuto di pagine in [C] e le pone in [D]

Gestione della memoria – 10



Gestione della memoria – 11

- Euristiche complesse e **non garantite** governano le scelte effettuate dalle varie attività di gestione delle liste [A] – [D]
 - L'amministratore di sistema può influenzare alcune euristiche mediante parametri di configurazione
- Lo stato della RAM viene mantenuto in una tabella dedicata acceduta per indice di pagina fisica (*page frame database*)
 - Pagina valida/invalida, contatore dei riferimenti, WS di appartenenza, lista di appartenenza, etc.

Gestione della memoria – 12

