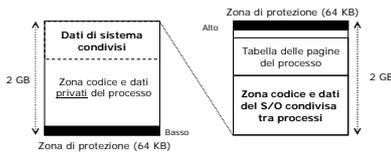


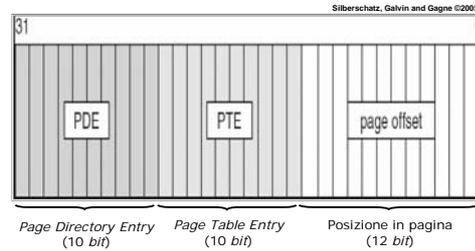
## Gestione della memoria – 1

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



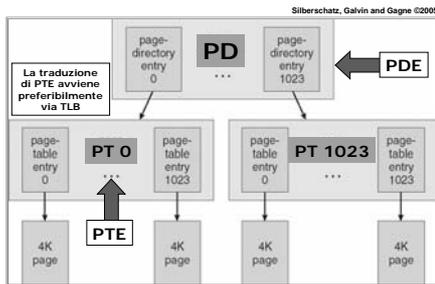
Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 365/389

## Indirizzamento virtuale – 1



Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 366/389

## Indirizzamento virtuale – 2



Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 367/389

## Indirizzamento virtuale – 3

- La **PD** è in una posizione nota a priori allineata al margine di una pagina
  - L'indirizzo della sua base è multiplo di 4 KB
  - Le sue *entry* sono ampie 32 bit
  - La *entry* indirizzata viene localizzata da
    - $Base_{PD} + (PDE \times 00_2)$ 
      - Perché il campo PDE indirizza unità ampie 4 B
- Una parte del valore trovato serve per localizzare la base della **PT** indirizzata
  - Come sopra per trovare la *entry* indirizzata in PT che fornisce la base della pagina riferita cui si somma il campo *page offset* per trovare il B di base della **parola** indirizzata
- Tutta questa traduzione può essere fornita dalla TLB

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 368/389

## Gestione della memoria – 2

- Zone di protezione (**indirizzi illegali**) usate per consentire la rilevazione di possibili errori di indirizzamento da parte del processo
- Codice e dati di S/O **replicati** nella memoria del processo consentono ai *thread* di passare in modo nucleo senza cambiare spazio di indirizzamento
  - Cambia solo lo *stack* (da utente a nucleo)

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 369/389

## Gestione della memoria – 3

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

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 370/389

## Gestione della memoria – 4

- Ogni pagina assegnata può essere rimossa temporaneamente dalla RAM
  - Le pagine codice e *file* mappati in memoria hanno una posizione nota (un *file*) su disco
  - Le aree di lavoro **non** la hanno
- MS Windows associa loro un *page file* ma solo nel momento del bisogno
  - Per evitare di tener impegnate vaste aree del disco
  - Fino a 16 *page file* a massima dimensione creati (ma non assegnati!) a tempo di inizializzazione del sistema

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 371/389

## Gestione della memoria – 5

- Più processi possono condividere l'accesso a pagine di uno stesso *file mappato in memoria*
  - Un **DLL** 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*)
  - Altri *file* dati velocizzano la comunicazione tra processi
  - 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 a 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

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 372/389

## Gestione della memoria – 6

- Ogni processo può gestire direttamente la propria memoria virtuale tramite **Win32 API**
- Le relative chiamate di sistema operano su **regioni** di pagine **contigue**
  - Agendo sulla memoria virtuale del chiamante
- Le pagine possono essere acquisite (per assegnazione o prenotazione), rilasciate, protette, rese inamovibili, etc.
- Chiamate di sistema sono anche disponibili per la gestione dei *file* mappati in memoria

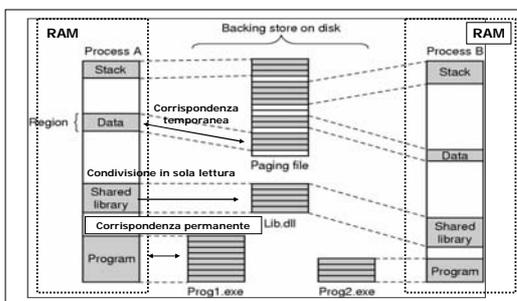
Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 373/389

## Gestione della memoria – 7

- **Paging-on-demand** con pagine ampie 4 KB (Pentium) fino a un massimo di 64 KB
  - Il S/O può usare pagine ampie 4 MB per ridurre la dimensione della propria tabella delle pagine
  - **Gestione per processi**
    - Un descrittore (**Virtual Address Descriptor**) raccoglie le informazioni di controllo dello spazio di indirizzamento virtuale del processo
      - Suddiviso per regioni di pagine virtuali
    - A ogni regione effettivamente in uso corrisponde una lista delle pagine che la compongono, il cui puntatore è posto nel **VAD**

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 374/389

## Gestione della memoria – 8



Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 375/389

## Gestione della memoria – 9

- L'assegnazione di pagine *on demand* avviene per **gruppi di pagine contigue** ( $\leq 8$ )
  - Per garantire maggior località
- **5** diverse situazioni possono verificarsi a seguito di un riferimento fallito
  - La **pagina riferita non è assegnata al processo**
    - Errore fatale → **BSOD**
  - La **pagina non può essere riferita**
    - Errore fatale (*protection fault*) → **BSOD**
  - La **pagina condivisa esiste ma non può essere scritta**
    - Copia locale assegnata al richiedente (*copy-on-write*)
  - Le **aree stack o dati devono crescere**
    - Assegnazione di una nuova pagina in RAM
  - La **pagina riferita è prenotata ma non ancora assegnata**
    - La pagina viene assegnata e posta in RAM

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 376/389

## Gestione della memoria – 10

- Il caricamento di una nuova pagina in RAM può richiedere il **rimpiazzo** locale di una 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  $Min_i \leq \# \{I_i\} \leq Max_i$
  - Il rimpiazzo avviene entro il **WS** del richiedente e **solo se**  $\# \{I_i\} = Max_i$ , altrimenti la pagina viene aggiunta
    - Politica di rimpiazzo locale
- Si ha **rimpiazzo globale** se e solo se un particolare processo deve scambiare proprie pagine tra RAM e disco troppo spesso

## Gestione della memoria – 11

- Anche il S/O stesso è visto come un processo con un proprio **WS** con pagine rimpiazzabili
  - Solo alcune pagine del S/O sono **inamovibili**
- Un **daemon** di **kernel** con periodo 1 s. accerta che vi siano sufficienti pagine libere
- Se insufficienti il **daemon** attiva un **thread** del **Memory manager** che esamina con una euristica gli WS dei processi per rilasciarne pagine
  - 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

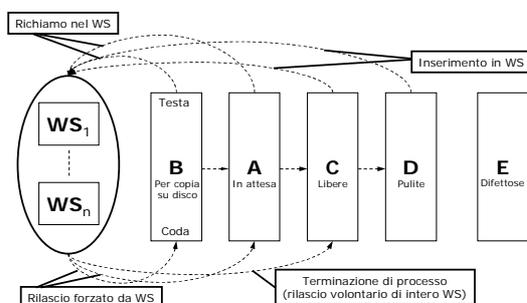
## Gestione della memoria – 12

- Ciascuna **page frame** in RAM può essere
  - **In uso** e appartenere a 1 WS ( $\geq 1$  se condivisa)
  - **Rilasciata** e appartenere a 1 sola lista
    - **[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** per consentire riassegnazione **senza travaso di informazione privata**
    - **[E] Difettosa**: pagina che non può più essere utilizzata a causa di difetti nella zona di memoria fisica

## Gestione della memoria – 13

- 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]**
  - La riscrittura di **file** mappati su disco può richiedere nuovi blocchi
  - La loro assegnazione richiede di aggiornare la mappa dei blocchi e per farlo possono servire nuove pagine di lavoro in RAM
    - Questa situazione è sorgente di **stallo potenziale**
  - Il 2o **daemon** può però **sempre** creare spazio in RAM per pagine di lavoro perché il **file** delle pagine è preallocato e dunque sempre disponibile
- 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]**

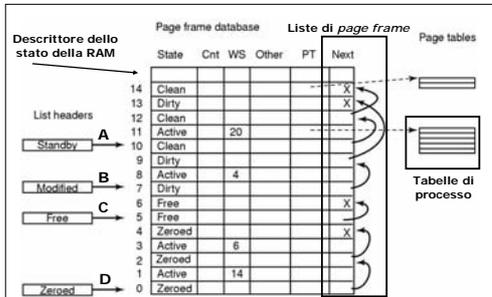
## Gestione della memoria – 14



## Gestione della memoria – 15

- 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 – 15



Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 383/389

## Gestione dell'I/O – 1

- Architettura progettata per massima flessibilità
  - Per supportare agilmente l'arrivo di nuove periferiche
- Sotto la responsabilità dell'**I/O manager** coadiuvato dal **Plug-&-play manager**
- Il **p&p m** interroga ogni *bus* per determinare la presenza di periferiche
  - A tempo di inizializzazione per le interfacce **statiche**
  - **Periodicamente** per le interfacce **dinamiche** (p. es.: USB)
- $\forall$  periferica registrata viene caricato e installato il corrispondente gestore (come per GNU/Linux) cui si associa uno specifico **oggetto di controllo**

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 384/389

## Gestione dell'I/O – 2

- **Dispositivi di ingresso**
  - Tastiera, *mouse*, *touchpad*, *cloche (joystick)*, telecamera, microfono per comandi vocali, lettore codice a barre, ...
- **Dispositivi di uscita**
  - Schermo, stampante, *plotter*, masterizzatore, schede suono, ...
- **Dispositivi di memorizzazione**
  - Dischi magnetici flessibili (*floppy*), ad alta densità (ZIP), fissi (*hard*) di tipo IDE o SCSI, e *flash hard*, disco ottico di tipo CD-ROM o DVD, nastro
- Diverse caratteristiche di comportamento e di interfacciamento per ciascuna tipo di periferica

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 385/389

## Gestione dell'I/O – 3

- Il **Power manager** si preoccupa di contenere il consumo energetico del sistema
  - Chiedendo all'**I/O manager** di cambiare lo stato dei gestori dei dispositivi in relazione all'uso delle periferiche
- La richiesta di dati di FS è inviata al **Cache manager**
  - Se necessario girata all'**I/O manager** che la indirizza al gestore appropriato
  - Il FS viene visto come gestore di periferica di I/O

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 386/389

## Gestione dell'I/O – 4

- Ogni gestore di periferica **conforme** deve esibire caratteristiche **fissate** di comportamento
  - Trattare richieste di servizio codificate in forma di pacchetto standard (IRP, **I/O request packet**)
  - Avere e usare rappresentazione a **oggetti** di tipo Win32 API
  - Prevedere il trattamento di periferiche rimovibili
  - Obbedire a richieste di cambio di stato da parte del **Power manager**
  - Aderire alle direttive di configurazione emesse dall'**I/O manager** (nessun prerequisito del gestore è immodificabile)
  - Permettere più esecuzioni concorrenti  $\rightarrow$  **procedure rientranti**
  - Essere utilizzabile **anche** in ambiente Windows 98

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 387/389

## Gestione dell'I/O – 5

- Il **Plug-&-play manager** interroga ogni periferica rilevata per identificarla e localizzarne sul FS il *software* di gestione
  - Caricamento dinamico automatico
  - Altrimenti finestra di dialogo per richiedere l'inserzione di un disco con il *software* necessario
- Ogni gestore deve fornire alcune procedure con profilo e comportamento predefiniti
  - Servizi localizzati a partire dall'**oggetto** associato alla periferica
    - Inizializzazione del gestore, registrazione della periferica, configurazione del vettore delle interruzioni, ...

Il Sistema Operativo MS Windows (parte 2) Sistemi Operativi - T. Vardanega Pagina 388/389

## Gestione dell'I/O – 6

- Oltre a demandare a **HAL** il trattamento uniforme delle caratteristiche specifiche dell'*hardware* del sistema, alcuni gestori prevedono una struttura a livelli
  - **Livello alto**
    - **Gestione funzionale** della periferica
      - Analoga alla logica d'uso dei *file* in UNIX
  - **Livello basso**
    - **Gestione dei protocolli** di comunicazione fisica verso la periferica
      - Per esempio: *bus* di tipo PCI, USB, SCSI