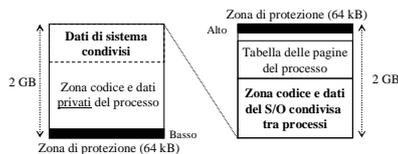


## Gestione della memoria – 1

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



Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 246

## Gestione della memoria – 2

- Zone di protezione (= indirizzi illegali) per rilevare possibili errori di indirizzamento nell'esecuzione del processo
- Codice e dati di S/O **replicati** nella memoria del processo consentono alle *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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 247

## Gestione della memoria – 3

- Una pagina virtuale (*frame*) può essere
  - R / W / E
  - Libera**: non in uso → non può essere riferita
    - 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 se non già presente in RAM
  - Prenotata**: non ancora in uso, ma **non libera**
    - Consente di assegnare pagine contigue a processi

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 248

## 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
- Windows associa loro un *page file* ma solo nel momento del bisogno
  - Per evitare di dover 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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 249

## Gestione della memoria – 5

- Più processi possono condividere l'accesso a **pagine** di uno stesso *file* mappato in memoria
  - Un **DLL** ne è un tipico esempio
    - 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 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**

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 250

## 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** in memoria virtuale
- 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

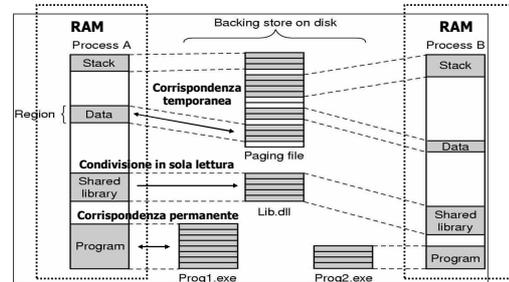
Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 251

## Gestione della memoria – 7

- **Paging-on-demand** con pagine ampie 4 kB (Pentium) fino ad 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
  - Ad 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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 252

## Gestione della memoria – 8



Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 253

## Gestione della memoria – 9

- L'assegnazione di pagine *on demand* avviene per **gruppi di pagine contigue** ( $\leq 8$ )
  - Per maggior località
- 5 situazioni possono verificarsi a seguito di un riferimento fallito
  - **La pagina riferita non è assegnata al processo**
    - Errore fatale
  - **La pagina non può essere riferita**
    - Errore fatale (*protection fault*)
  - **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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 254

## Gestione della memoria – 10

- Il caricamento di una nuova pagina in RAM può richiedere il **rimpiaccio 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**), la cui ampiezza può variare solo entro limiti consentiti
 
$$\text{Min}_i \leq \# \{I_i\} \leq \text{Max}_i$$
  - Il rimpiazzo avviene entro il **WS** del richiedente e **solo se**  $\# \{I_i\} = \text{Max}_i$ , altrimenti la pagina viene aggiunta
- Si ha **rimpiaccio globale** solo se un particolare processo deve troppo spesso salvare proprie pagine su disco

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 255

## Gestione della memoria – 11

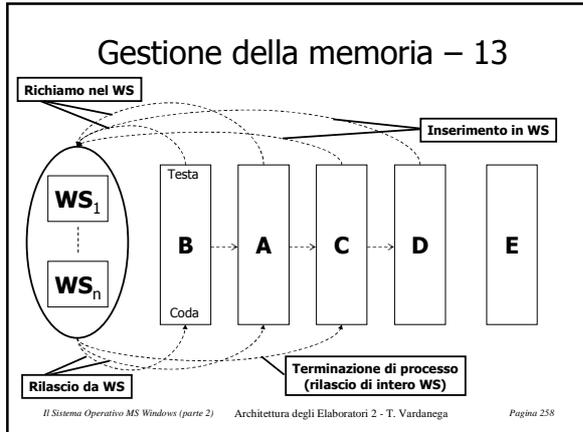
- Il S/O è visto come un processo con un suo **WS** con pagine rimpiazzabili
  - Tranne alcune che sono configurate come **inamovibili**
- Un **daemon** di **kernel** con periodo 1 s. accerta che vi siano sufficienti pagine libere
- Se insufficienti, il **daemon** attiva una **thread** del **Memory manager** che esamina gli WS dei processi per rilasciare pagine
  - Processi non recentemente attivi con WS ampi vengono presi in esame prima degli altri
  - Le pagine necessarie si prelevano dagli WS di ampiezza vicina al massimo e con scarso uso recente

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 256

## Gestione della memoria – 12

- Ciascuna pagina in RAM può essere
  - **In uso**, e appartenere a 1 WS ( $\geq 1$  se condivisa)
  - **Rilasciata**, e appartenere a 1 ed 1 sola lista tra:
    - **A: In attesa**: pagine recentemente rimosse dal WS di un processo, ma ancora associate a esso, e **non** modificate
      - Possono essere riassegnate e sovrascritte senza problemi
    - **B: Da copiare su disco**:  $\sim A$ , ma, se rimpiazzate, devono essere riportate su disco
    - **C: Libere**:  $\sim A$ , ma non più associate ad alcun processo
    - **D: Azzerate**:  $\sim C$ , ma con contenuto obliterato per consentire riassegnazione **senza travaso di informazione privata**
    - **E: Difettose**: pagine che non possono essere più utilizzare per difetti nel corrispondente banco di memoria

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 257



### Gestione della memoria – 14

- La *swapper thread* del *Memory manager* porta in **A** o **B** le pagine dello *stack* dei processi tutte le cui *thread* siano state recentemente inattive
- 2 *daemon* del *kernel* assicurano che vi siano abbastanza pagine in **C**, salvando su disco quelle in **B** (nei *file* mappati e nei *file* delle pagine) ed accodandole in **A**
  - La riscrittura su disco di *file* mappati può richiedere nuovi blocchi, la cui assegnazione richiede di aggiornare in RAM la mappa dei blocchi, per cui possono servire nuove pagine libere → **stallo potenziale**
  - Il 2o *daemon* può però sempre creare spazio in RAM, perché il *file* delle pagine è preallocato (e dunque non cresce!)
- Un WS che cresce preleva pagine libere da **C** se le sovrascrive interamente, da **D** altrimenti
  - Un *daemon* dedicato del *kernel* azzerava periodicamente il contenuto di pagine in **C** e le pone in **D**

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 259

### Gestione della memoria – 15

- Euristiche complesse e non garantite governano le scelte effettuate dalle varie attività di gestione delle liste
  - L'amministratore di sistema può influenzare talune 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.

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 260

### Gestione della memoria – 15

Descrittore dello stato della RAM	Page frame database					Next	Page tables
	State	Cnt	WS	Other	PT		
14	Clean						
13	Dirty					X	
12	Clean						
11	Active	20					
10	Clean						
9	Dirty					X	
8	Active	4					
7	Dirty						
6	Free					X	
5	Free						
4	Zeroed					X	
3	Active	6					
2	Zeroed						
1	Active	14					
0	Zeroed						

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 261

### 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-and-play manager*
- Il *p&pm* 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)
- ∇ 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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 262

### 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*, puntatore laser (*beamer*), 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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 263

### 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** e, se necessario, da questi girata all'**I/O manager** che la indirizza al gestore appropriato
  - FS visto come gestore di periferica di I/O

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 264

### 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 ad oggetti
  - 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 immodificabile)
  - Permettere più esecuzioni concorrenti → **procedure rientranti**
  - Essere utilizzabile **anche** in ambiente Windows 98

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 265

### 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) Architettura degli Elaboratori 2 - T. Vardanega Pagina 266

### 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
  - **Livello basso**
    - **Gestione dei protocolli** di comunicazione fisica verso la periferica
      - P.es.: *bus* di tipo PCI, USB, SCSI

Il Sistema Operativo MS Windows (parte 2) Architettura degli Elaboratori 2 - T. Vardanega Pagina 267