

## Lezione 14

### Windows XP

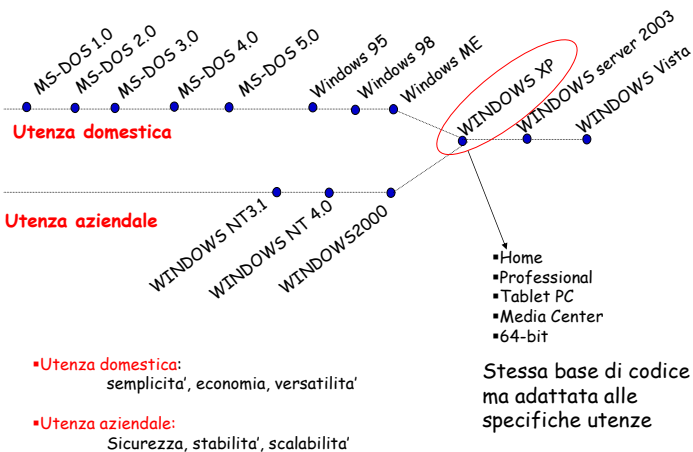
- Introduzione
- Architettura del kernel
- Gestione dei processi
- Gestione della memoria
- File system
- I/O
- WOW: Arriva Windows Vista



## Evoluzione

- 1981 - MS-DOS 1.0. s.o. a 16 bit. Solo **1MB di memoria centrale**
- 1983 - MS-DOS V.2.0 con driver per **hard disk** e gestione del **file system**
- 1984 - MS-DOS V.3.0 con supporto di **rete**
- 1985 - Windows 1.0 prima **interfaccia grafica**
- 1987 - Windows 2.0 con funzionalita' di **multitasking**
- 1988 - DOS V.4.0 con shell e gestione della **memoria > 640KB**
- 1991 - DOS V.5.0 gestione piu' **efficiente della memoria**
- 1992 - **Windows 3.1.** oltre 3 milioni di copie in due mesi: **supporto di rete**
- 1993-94 - Windows NT 3 e NT4. **S.O. object oriented** per server di alto livello. Stessa interfaccia grafica di Windows 3.1 ma kernel completamente riscritto
- 1995 - **Windows 95:** S.O. per **applicazioni a 32 bit.** Non piu' una interfaccia grafica come Windows 3.1 ma un completo S.O.
- 1996-1998 - Windows CE 3.0. S.O. per **computer palmari**
- 1998 - Windows 98. **browser** integrato, supporti **Java** e **HTML** e vari supporti hardware (USB, firewire, DVD,... )
- 2000 - Windows 2000 con supporto molto efficiente per le **connessioni di rete**
- 2001 - Windows XP: supporto a **64 bit**
- 2003 - windows server 2003 - **evoluzione di XP** per le aziende
- 2007 - Windows Vista

## Linee di sviluppo



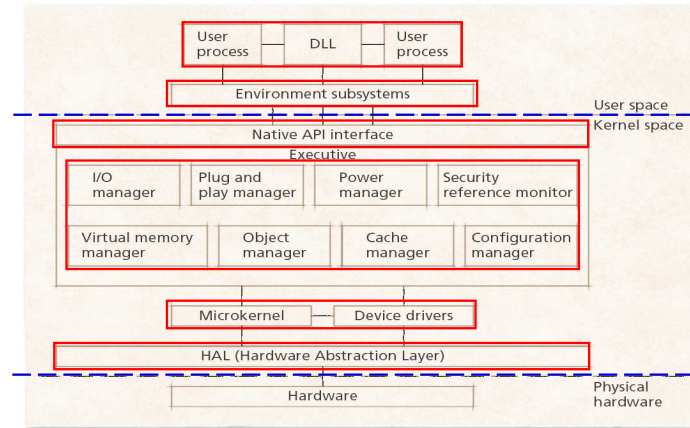
## Obiettivi

- **Fondere le linee di sviluppo domestico e aziendale**
  - Stesso codice base con moduli intercambiabili
- **Sicurezza**
  - Lista di controllo degli accessi, firewall
- **Scalabilita'**
  - Piattaforme parallele /distribuite
- **User-friendly**
  - Nuova interfaccia grafica, supporto multimediale
- **Boot veloce**
  - 30 secondi dall'accensione, 5 secondi da standby

## Architettura del sistema

- Architettura a microkernel ibrido:
  - Kernel compatto che fornisce pochi servizi di base
  - Sottosistemi eseguiti in modalita' kernel
  - Organizzazione stratificata
- Livelli software (dal basso)
  - Hardware Abstraction Layer (HAL)
  - Microkernel e driver dei dispositivi
  - Executive (servizi generali del s.o.)
  - Interfaccia nativa
  - Sottosistemi di ambiente
  - Processi utente

## Architettura del sistema



## Hardware Abstraction Layer (HAL)

Molti sistemi, anche basati sullo stesso processore hanno **dettagli architetturali differenti** (caratteristiche della scheda madre, cache, dispositivi proprietari,...)



utilita' di uno **strato software** per **nascodere l'hardware** al resto del sistema (Strato di Astrazione dell'Hardware)



E' sufficiente **adattare solo l'HAL** da un sistema all'altro e i **moduli del kernel** possono essere sviluppati **senza tenere conto dei dettagli hardware**

## Microkernel e gestore dei dispositivi

- **Microkernel:**
  - Fornisce i **meccanismi base del sistema** (scheduling e sincronizzazione dei threads, gestione delle interruzioni)
- **Gestore dei dispositivi**
  - Contiene la parte del software di **I/O dipendente dall'hardware**



HAL, microkernel e gestore dei dispositivi permettono di **portare Windows XP senza modifiche** su numerose architetture

## Executive e interfaccia nativa

- L'insieme dei moduli che costituiscono il **s.o. vero e proprio**
- sono eseguiti in **modalita' kernel** e possono **accedere a tutta la memoria**
- **Fornisce i principali servizi:**
  - Gestore della memoria,
  - File system
  - I/O di alto livello
  - ...
- I processi utente accedono ai servizi executive mediante le **API native**, chiamate di sistema proprie degli executive

## Sottosistemi di ambiente

- Processi in modalita' utente interposti **tra gli executive e il resto dello spazio utente**
- Esportano una **API per uno specifico ambiente**
  - Win32 (installato di default)
  - POSIX
  - Service for Unix (SFU)
  - Windows over Windows64 (WOW64, disponibile per Windows XP 64 bit) per l'esecuzione di applicazioni a 32 bit



Miglioramento della portabilita' delle applicazioni

## Registro

- **Database centralizzato strutturato ad albero** che mantiene tutte le **informazioni di configurazione** degli utenti, delle applicazioni del sistema e dell'hardware
- **Accessibile da parte di tutti i processi** per conoscere le risorse e le impostazioni del sistema
- **Esempio:**
  - Configurazione **utente**: impostazioni desktop
  - Configurazione di **sistema**: software installato
  - Configurazione **hardware**: informazioni sui dispositivi installati
  - Configurazione **applicazioni**: valori di default per una applicazione
- Il **gestore della configurazione** e' il componente executive responsabile della gestione del registro

## Oggetti

- Tutte le **risorse hardware** (es. dispositivi) o **logiche** (es. processi) sono rappresentati mediante **oggetti**
- **Oggetti = strutture dati** con informazioni sulle caratteristiche e operazioni consentite sull'oggetto
- Ogni oggetto e' **definito da un tipo** e sono creati dagli executive:
  - **Es.** Il gestore dell'I/O crea l'oggetto di **tipo file**
- Il tipo specifica gli **attributi e le operazioni consentite**
  - **Es. attributi:** nome, proprietario, ...
  - **Es. operazioni:** open, close, read, ...

## Altri esempi

- Tipo thread:
  - Operazioni consentite: create, sleep, execute ,...
- Tipo semaforo:
  - Operazioni consentite: create, wait, signal
- Tipo Pipe:
  - Operazioni consentite: create, read, write

## Handle (maniglia)

- Struttura dati che permette ai processi di **manipolare gli oggetti**
- Attraverso un handle un processo puo' invocare una delle procedure dell'oggetto e modificarne le caratteristiche
- Per **creare un oggetto** un processo:
  1. Invoca il **gestore degli oggetti**
  2. Il gestore degli oggetti crea la struttura dati corrispondente
  3. Passa un handle al processo
- Gli handle **possono essere comunicati** tra processi
- Per ogni oggetto, il **gestore degli oggetti** tiene il conto degli **handle** che vi fanno riferimento (necessario per la cancellazione)

## Interruzioni

non hanno tutte la stessa importanza



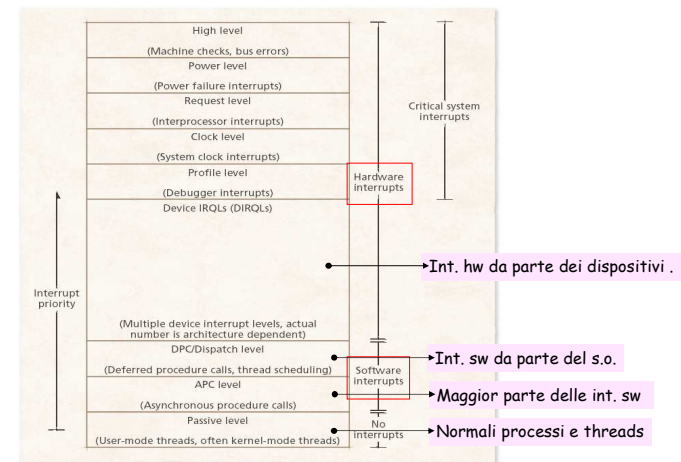
Windows XP definisce dei **livelli di priorit  di interruzioni** (Interrupt Request Level = IRQL)

Durante l'esecuzione, la CPU fa sempre riferimento ad un IRQL

Se arriva una interruzione con **IRQL maggiore** viene **interrotta l'esecuzione corrente**

Se arriva una interruzione con **IRQL inferiore**, essa **viene ritardata**

## Livelli di priorit  (IRQL)



## Interruzioni software: APC e DPC

- **Asynchronous procedure calls (APC)**
  - Invocazioni di routine di servizio che possono essere accodate per l'esecuzione da parte di un altro thread
  - Es: I/O asincrono.
    - Un thread inizia una richiesta e prosegue con altro lavoro
    - Al completamento dell'I/O il sistema invia al thread una APC
  - Disponibili in modalita' kernel e modalita' utente
- **Deferred procedure calls (DPC)**
  - Invocazioni di routine di servizio eseguite nello stesso contesto del thread che le invoca
  - Usate generalmente per le interruzioni da parte del s.o.
  - Es: scheduling dei thread

14. Windows XP

17

marco lapegna

## Processi

- **3 strutture dati per ogni processo :**
  - **Blocco EPROCESS** (usato dai componenti executive). Contiene
    - Process ID, tabella degli handle, dimensione del working set.
    - Puntatore al PEB
    - Organizzati secondo una lista nello spazio kernel
  - **Blocco KPROCESS** (usato dal microkernel). Contiene
    - Parte del blocco EPROCESS
    - Informazioni per lo scheduling (priorita', quanto )
    - Informazioni sulla sincronizzazione (semafori )
  - **Process environment block (PEB):**
    - Residente nello spazio utente
    - Informazioni utili al processo stesso
    - Es. Librerie dinamiche collegate, sezioni critiche acquisite

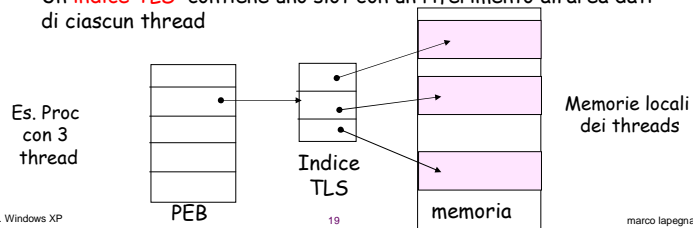
14. Windows XP

18

marco lapegna

## Thread

- **3 strutture analoghe per i thread:**
  - Blocco ETHREAD
  - Blocco KTHREAD
  - Thread environment block (TEB)
- I thread condividono lo spazio di indirizzamento del processo , ma possono avere una **propria memoria locale dei thread (Thread Local Storage TLS)**.
- Un **indice TLS** contiene uno slot con un riferimento all'area dati di ciascun thread

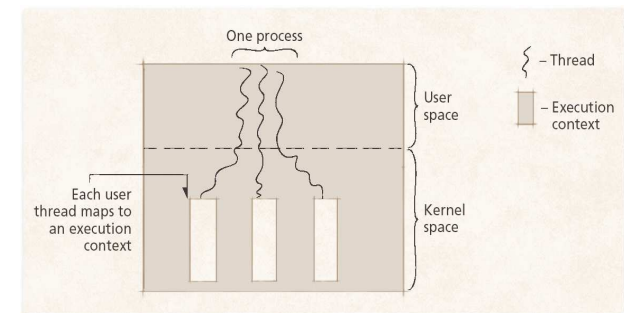


14. Windows XP

19

marco lapegna

## Implementazione dei thread



- **Implementazione 1 a 1**
  - 1 thread del kernel corrisponde a 1 thread utente
- **Unita' di elaborazione dello scheduler**

14. Windows XP

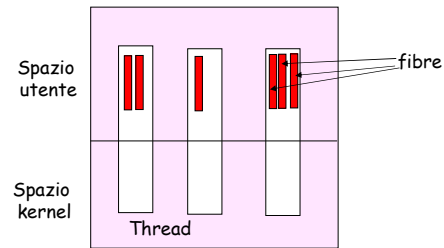
20

marco lapegna

## Fibre

- I thread possono creare le **fibre**
- Unità di elaborazione simili ai thread ma **gestiti in modalità utente** dal thread stesso

Il thread definisce lo scheduling  
(più aderente alle esigenze delle applicazioni)

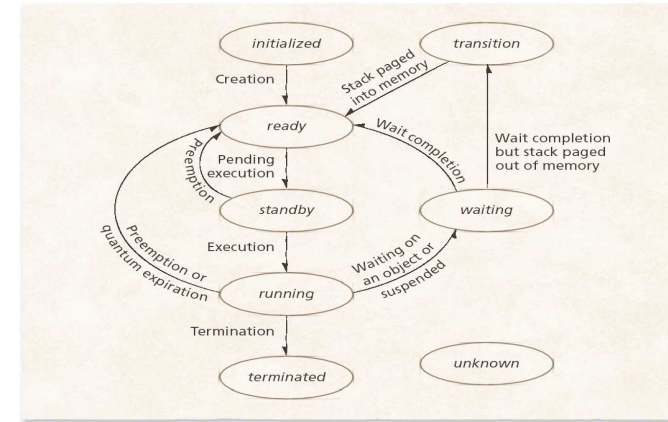


14. Windows XP

21

marco lapegna

## Scheduling dei thread



14. Windows XP

22

marco lapegna

## Scheduling dei thread

- Scheduling basato su **priorità con prelazione**
- **32 livelli di priorità**
  - 0 = priorità minore
  - 31 = priorità maggiore
- **32 ready queue** (una per ogni priorità)
- Ogni coda gestita **round robin**.
- **Due classi di thread**
  - Thread **real time** (priorità **statiche** 16 → 31)
  - Thread **normali** (priorità **dinamiche** 0 → 15)

14. Windows XP

23

marco lapegna

## Scheduling dei thread

### Classi di priorità

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

Livelli di priorità

- I thread normali sono divisi in **classi di priorità**
  - es. proc di sistema o processi utente
- All'interno di ogni classe la esistono **livelli di priorità**
  - Es. background/foreground

14. Windows XP

24

marco lapegna

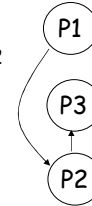
## Scheduling dei thread

- I thread normali hanno una **priorita' dinamica**
  - finestra che riceve input da tastiera (thread interattivi)  
→ **Aumento della priorita' +2**
  - Completamento I/O, ottenimento di una risorsa,  
→ **aumento della priorita' +1**
  - Scadenza del quanto  
→ **riduzione della priorita' -1**
- **comunque mai sopra highest o sotto lowest**

## 2 problemi

1. **Posticipazione indefinita**
2. **Inversione della priorita'**  
P1 priorita' alta in attesa di risorsa in possesso di P2  
P2 priorita' bassa  
P3 priorita' media

P3 ha la precedenza su P2 (ed anche su P1)



**SOLUZIONE**  
Aumento della priorita' dei thread  
in coda da molto tempo (aging)

## Sincronizzazione dei thread

- Windows definisce **numerosi strumenti per la sincronizzazione** chiamati **oggetti dispatcher**
  - Possono essere in uno stato
    - **signaled** (consentono la continuazione dell'elaborazione)
    - **unsignaled** (non consentono la continuazione dell'elaborazione)
1. **Oggetti mutex**
    - Semafori binari
    - **Signaled** quando il proprietario di un mutex lo rilascia

## Sincronizzazione dei thread

2. **Oggetti semafori**
  - Semafori contatori
  - **Signaled** quando il contatore diventa positivo
3. **Oggetti evento**
  - Usati principalmente per la sincronizzazione nelle operazioni di I/O
  - **Signaled** quando si verifica un evento
4. **Oggetti waitable timer**
  - Usati per le sincronizzazioni "periodiche" (es. salvataggio di file)
  - **Signaled** quando e' trascorso l'intervallo di tempo specificato
5. **Oggetti per sezioni critiche**
  - Analoghi ai mutex ma usati per sincronizzare thread di uno stesso processo
  - Piu' efficienti perche' non passano in modalita' kernel

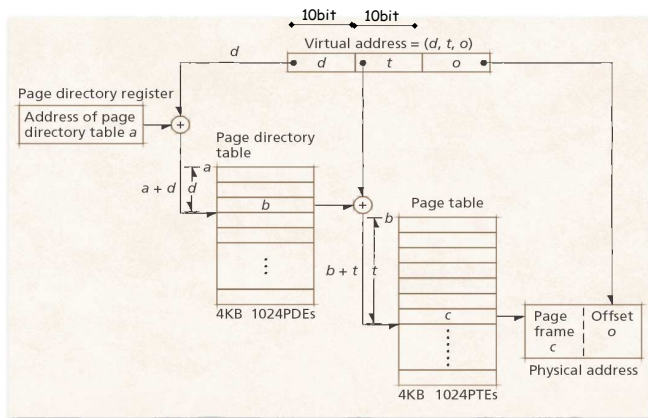
## Virtual Memory Manager (VMM)

- Executive responsabile della **gestione della memoria**
- Le pagine di un processo sono divise tra la memoria centrale e il disco, in 16 strutture chiamate **pagefile**
- 32 bit di indirizzamento → **4GB di spazio di indirizzamento virtuale**
  - **2 GB** di spazio **utente** (usato per le pagine del processo)
  - **2 GB** di spazio di **sistema** (usato per le tabelle delle pagine e altre strutture dati)
- **Frame di dimensione fissa di 4KB**

## VMM

- Tabella delle pagine a due livelli
  1. **Elenco delle tabelle delle pagine**
    - Un elenco delle tabelle per processo
    - Ogni elemento (Page Directory Entries) punta ad una tabella di secondo livello
    - La posizione dell'elenco e' memorizzata in un apposito registro della CPU durante il cambio di contesto
  2. **Tabella delle pagine**
    - Ogni elemento (Page Table Entries) punta ad un frame in memoria
  3. **Frame in memoria**
    - Contiene i dati delle pagine
- I registri associativi (TLB) possono accelerare la traduzione degli indirizzi

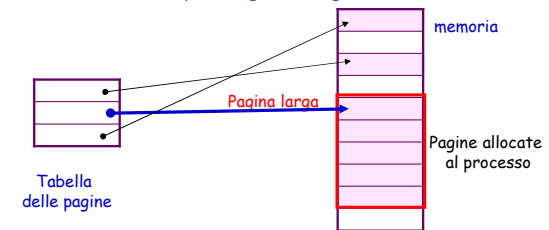
## Traduzione degli indirizzi virtuali



## Pagine larghe

- Quando possibile il VMM tenta di allocare **pagine consecutive come una unica grande pagina** di almeno 2 MB
  - Un **unico elemento** nella tabella delle pagine
  - Tale pagina avra' una **probabilita' maggiore di essere referenziata**

Persistenza del riferimento piu' lunga nei registri associativi





## Elementi della tabella delle pagine

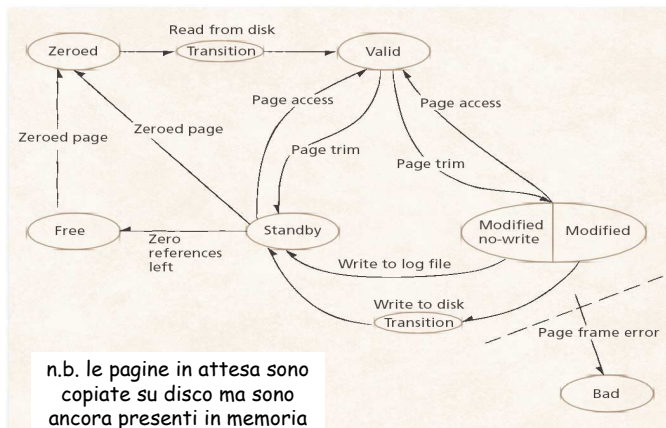
- **32 bit:**
  - 5 bit per la protezione
  - 20 bit per definire il frame in memoria
  - 4 bit per definire il pagefile a cui fare riferimento
  - 3 bit di stato della pagina
  
- **Bit di protezione**
  - Read
  - Write
  - Execute
  - Copy-on-write
  - Eccezione nell'accesso
  
- **Bit di stato**
  - Valida
  - Modificata
  - In transizione

## Stati delle pagine

Per tenere traccia dell'uso della memoria il sistema classifica i frame in **8 stati differenti** e mantiene **una lista per ogni stato**

- **Valido** la pagina memorizzata appartiene al working set di un processo
- **Transizione** la pagina memorizzata sta per essere trasferita sul disco
- **Modificato** la pagina memorizzata e' stata rimossa dal w.s. di un processo ma non e' ancora coerente con la copia sul disco
- **In attesa** il frame contiene una pagina che e' stata rimossa dal working set di un processo
- **Modificato ma da non scrivere** la pagina e' stata rimossa dal w.s. di un processo e non e' coerente con la copia sul disco. Per motivi di coerenza del file system e' necessario prima annotare la modifica.
- **Libero** il frame non contiene pagine valide
- **Azzerata** il frame ha tutti i bit azzerati e non fa parte di nessun w.s.
- **Bad** il frame ha generato un errore

## Localized LRU



## Sostituzione delle pagine

- Windows XP utilizza un modello ibrido di **sostituzione delle pagine**:  
Working set + Localized Least recently used (LLRU)
  
- **Vengono assegnati ad ogni processo**
  - Working set **minimo**
  - Working set **massimo**
  - Tali valori dipendono dal **carico del sistema**
  
- Quando un processo richiede piu' pagine del working set massimo il VMM sposta una pagina sul disco secondo la **politica LLRU**
  
- **Localized** → politica focalizzata sul singolo processo

## Localized LRU

- Elimina tutte le **pagine che eccedono la dimensione** del working set massimo
  - Le pagine eliminate sono dette anche ridotte
- Le pagine ridotte vengono spostate nella
  - Lista delle **pagine in attesa**
  - Lista delle **pagine modificate**
  - Lista delle **pagine modificate ma da non scrivere**
- La pagina viene spostata nuovamente nella lista delle **pagine valide** in caso di nuovi riferimenti (soft fault)
- La pagina viene spostata nella lista delle **pagine libere** se non viene piu' referenziata per un certo periodo di tempo
- Un thread di sistema a bassa priorita' la azzerata e la sposta nella lista delle **pagine azzerate**
  
- Le pagine per i nuovi processi vengono prese dalla lista delle pagine azzerate

14. Windows XP

37

marco lapegna

## Eccezioni nella sostituzione delle pagine

- Non tutte le pagine possono essere rimosse dalla memoria. **La memoria e' divisa in due segmenti**
  - **Memoria paginata**
    - Il VMM puo' rimuovere le pagine
  - **Memoria non paginata**
    - Il VMM non puo' rimuovere le pagine
    - Usato per
      - Driver dei dispositivi
      - Parte del kernel
      - Interrupt service routine

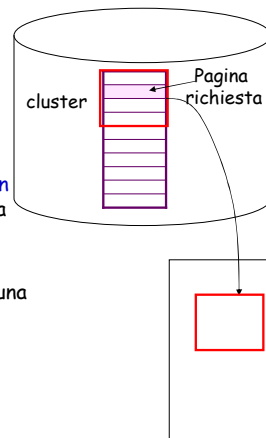
14. Windows XP

38

marco lapegna

## Ottimizzazione dell'algoritmo LLRU

- Il sistema impiega una **paginazione su richiesta**
- gruppi di **pagine sono memorizzati in cluster** nei pagefile
- In caso di page fault, Windows **carica in memoria tutto il cluster** che contiene la pagina richiesta
- **principio di localita' spaziale** (se serve una pagina, forse a breve serviranno anche quelle vicine)



Paginazione su richiesta clusterizzata

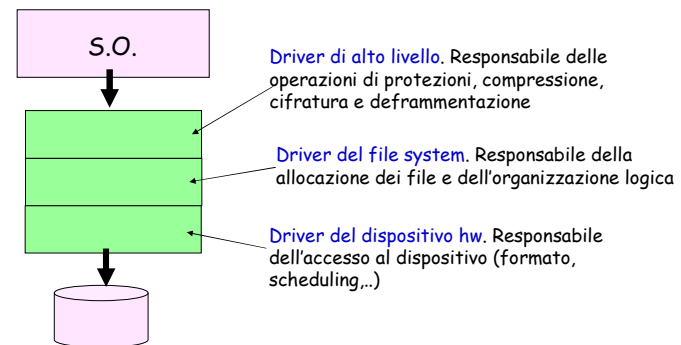
14. Windows XP

39

marco lapegna

## File system

- Il file system di windows XP sono divisi in tre livelli



14. Windows XP

40

marco lapegna

## File system NTFS

- Il primo file system di Windows era basato su una **allocazione tabellare** (File Allocation Table)
  - Una tabella con **una entry per ogni blocco fisico** del disco
- Principali problemi: **frammentazione, inefficienza**
- Esempio:** disco di 32 Gbyte
  - blocchi "piccoli" da 2 KB → oltre 16 milioni di blocchi → 64Mb per la tabella → **inefficienza**
  - blocchi "grandi" da 1 MB → **frammentazione**

Nei s.o. Windows NT, 2000, XP e Vista, il file system principale e'

**New Technology File System = NTFS**

14. Windows XP

41

marco lapegna

## FAT vs NTFS

	FAT16	NTFS
OS	DOS, Windows 95, 98	Windows NT, 2000, XP, Vista
Dim partizione	2GB	256 TByte
N.file	65000	> 4 miliardi
Dim file	2GB	16 TByte
Sicurezza	No	si
Compressione	No	si
Fault tolerance	Media	si

- Windows NT → NTFS 3
- Windows 2000 → NTFS 4
- Windows XP → NTFS 5
- Windows Vista → NTFS 6

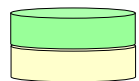
14. Windows XP

42

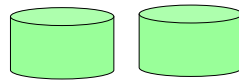
marco lapegna

## volumi

- La struttura fondamentale di NTFS e' il **volume**
  - Partizionamento logico** del disco (analogo alla partizione)
  - Creato dall'amministratore del sistema
  - Puo' occupare parte di un disco, un disco intero o comprendere piu' dischi



Disco con 2 volumi



2 dischi con 1 volume

- Nel caso di piu' dischi e' possibile utilizzare **tecniche RAID** per gestire il volume

14. Windows XP

43

marco lapegna

## cluster

- NTFS utilizza i **clusters** come unita' di allocazione del disco
  - Un cluster e' un **insieme di settori del disco** di dimensione compresa tra **2 e 8 KB**
  - L'indice del cluster e' chiamato **Logical Cluster Number (LCN)**



- Es. settori da 512 Byte e cluster di 2 KB

- Poiche' per dischi grandi la dimensione dei cluster e' molto minore di quella della FAT (32KB), la **frammentazione interna e' ridotta**

14. Windows XP

44

marco lapegna

## Metafile

- Tutte le informazioni che riguardano il volume, sono organizzate in 11 file regolari (metafile) che risiedono in una posizione prefissata (occupano i primi cluster, la **MFT zone**)
- metafile**
  - \$MFT** (elenco di tutti i file del file system compresi i metafile)
  - \$MFTMIRR** (copia di \$MFT, usata quando \$MFT si corrompe)
  - \$LOGFILE** (file che annota tutte le modifiche al file system)
  - \$VOLUME** (numero seriale, ora di creazione del volume,...)
  - \$ATTRDEF** (informazioni sugli attributi dei file)
  - \$BITMAP** (mappa dei cluster liberi)
  - .** (informazioni sulla directory radice)
  - \$BOOT** (contiene informazioni sul boot del volume)
  - \$BADCLUS** (elenco dei cluster danneggiati)
  - \$QUOTA** (info sull'uso del file system da parte degli utenti)
  - \$UPCASE** (Tabella di conversione carattere unicode)

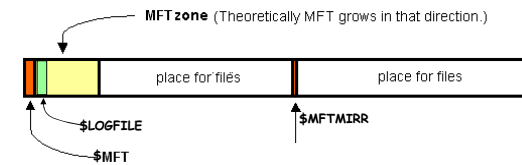
14. Windows XP

45

marco lapegna

## La MFT zone

- Dato che la **MFT** è un file residente nel file system esso può ingrandirsi e rimpicciolirsi, e quindi **frammentarsi**
- Il programma di deframmentazione non può agire sui metafile
- Per evitare la frammentazione, NTFS crea una **regione di cluster attorno alla fine della MFT** dove file e directory normali non possono essere memorizzati (**circa il 15% del disco**)
- Si noti che il file **\$MFTMIRR** è memorizzato al centro del volume



14. Windows XP

46

marco lapegna

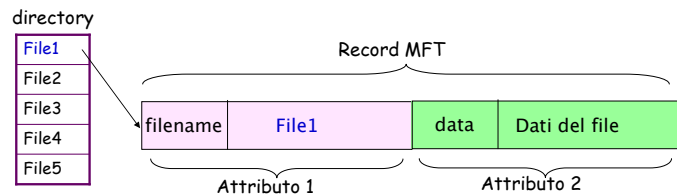
## \$MFT Master file table

- Composta da **record di dimensione fissa (1 KB)**
- Ogni file è rappresentato da **almeno un record nella tabella**
- Le informazioni di un file sono organizzate in **attributi**

Attributo = (intestazione : dati)

Esempi di attributi:

nome del file, data di creazione, proprietario, dati del file



14. Windows XP

47

marco lapegna

## Attributi

- Se il file è **sufficientemente piccolo** gli attributi del file sono memorizzati direttamente nei record della MFT

Attributi residenti

- Se il file è **molto grande** opportuni attributi puntano ai cluster che contengono i dati del file

Attributi non residenti



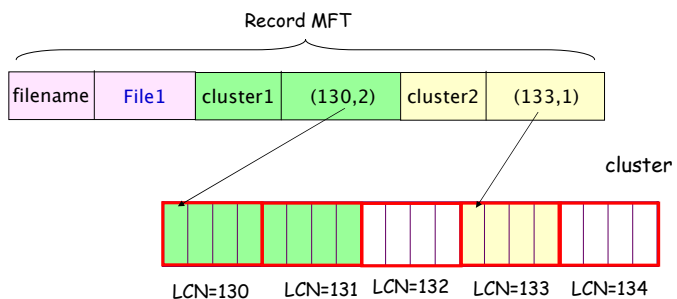
NTFS usa quindi un tipo di **allocazione indicizzata**

14. Windows XP

48

marco lapegna

## esempio



- filename e' sempre un **attributo residente**
- i dati del file risiedono nei cluster 130, 131 e 133 e sono **attributi non residenti**

14. Windows XP

49

marco lapegna

## \$LogFile

- Tutte le modifiche alle strutture dati sono eseguite come **transazioni**:
  - Prima della modifica di una struttura dati, il sistema scrive nel file di log un **record con tutte le informazioni sull'operazione da eseguire**
  - Dopo la modifica della struttura dati il **record** nel file di log viene **eliminato**
  - In caso di crash del sistema le strutture dati relative ad operazioni non completate **possono essere ripristinate a partire dal record** nel file di log

Journalized file system

14. Windows XP

50

marco lapegna

## Flussi di dati (stream)

- Il **contenuto di un file** e' memorizzato sotto forma di particolari attributi chiamati stream
- **Stream di default**
  - Attributo **senza nome**
  - Contenuto del file (testo, immagine, ..)
- **Stream alternativi**
  - Attributi **con nome**
  - Usato per altri metadati (titolo, oggetto, autore, data accesso,...)
  - Informazioni "right-click e proprieta'"

14. Windows XP

51

marco lapegna

## Directory in NTFS

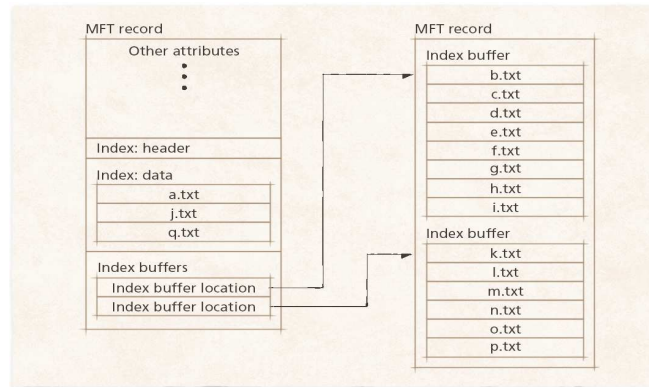
- **Anche le directory sono dei file ed hanno una entry nella MFT**
  - Ogni file directory contiene un **attributo chiamato index**
  - L'attributo index memorizza **l'elenco dei file** che si trovano nella directory
  - I nomi dei file sono memorizzati secondo un albero binario in **ordine alfabetico**
  - I dati del file directory sono i nomi dei file in essa contenuti
  - Ogni file della directory punta al record MFT relativo

14. Windows XP

52

marco lapegna

## Directory

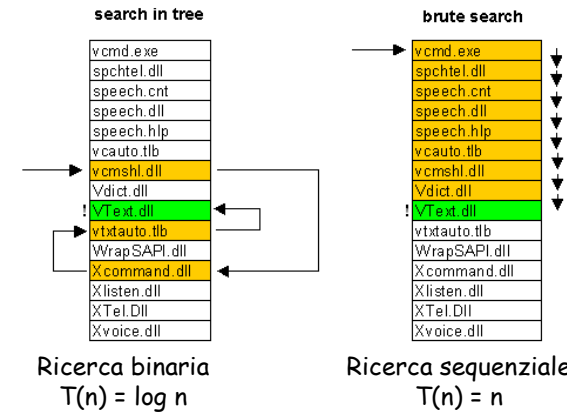


14. Windows XP

53

marco lapegna

## Ricerca di un file



14. Windows XP

54

marco lapegna

## Hard link

- Windows XP consente a piu' directory di fare riferimento allo stesso file
  - Si crea una nuova riga MFT per un file gia' esistente
- Nella riga MFT relativa al file esistente
  - Si aggiunge un nuovo attributo file\_name
  - Si incrementa il valore dell'attributo hard\_link
  - Si crea una nuova riga di directory per fare riferimento alla riga MFT del file esistente
- Il file non viene cancellato fino a quando l'attributo **hard\_link e' positivo**
- Limitazioni
  - Non sono consentiti **hard\_link a directory**
  - Non sono consentiti riferimenti a **file su altri volumi**

14. Windows XP

55

marco lapegna

## Punti di reparse

- Soluzione per **puntare a file e directory su altri volumi e file system**
- Punto di reparse = attributo di un file
  - Marcatore a 32 bit
  - 16 KB di dati
- Quando NTFS accede ad un file con un punto di reparse viene analizzato prima tale attributo
  - Si usa il **marcatore per definire il driver** del file system da usare
  - Il **driver elabora i dati** del punto di reparse

14. Windows XP

56

marco lapegna

## Esempi

- **File che viene spostato su dispositivo terziario**
  - I dati vengono cancellati
  - La riga MFT viene mantenuta
  - Viene aggiunto un punto di reparse con le indicazioni del dispositivo utilizzato
  - Trasparente all'applicazione
- **Montaggio di volumi**
  - Nella directory di montaggio viene aggiunto un punto di reparse con le indicazioni del volume da montare
- **Giunzioni di directory**
  - Permette i link a directory
  - Nel punto di reparse e' specificato il percorso della directory da collegare

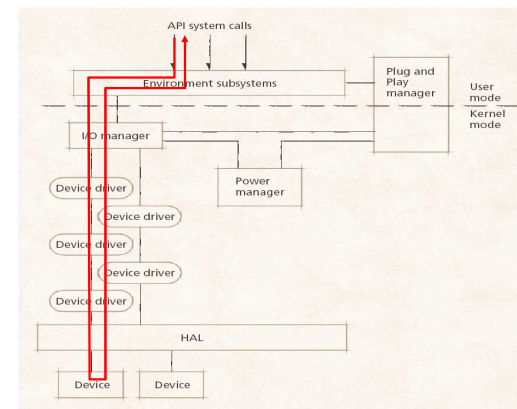
## Altre caratteristiche

- **Compressione dei file**
  - Trasparente alle applicazioni
  - Il file viene diviso in unita' di compressione (16 cluster) compresse separatamente (compressione segmentata)
  - Algoritmo di Lempel-Ziv
- **Cifratura dei file**
  - suddivisione in unita' di 16 cluster cifrati separatamente
  - Algoritmo a chiave pubblica/privata
- **File sparsi**
  - Memorizzazione dei soli elementi non nulli
  - Memorizzazione della posizione degli zeri

## Gestione dell'I/O

- La gestione dell'I/O coinvolge direttamente molte componenti del sistema operativo
1. **Sottosistema di ambiente**
    - Per permettere portabilita' delle applicazioni
  2. **Il gestore dell'I/O**
    - Executive (componente del kernel) che interagisce con i driver e verifica la disponibilita' del dispositivo
  3. **Driver dei dispositivi**
    - Organizzati secondo una pila
  4. **Gestore del Plug and Play**
    - Che riconosce e alloca dinamicamente i dispositivi
  5. **Gestore dell'alimentazione**
    - Utile nei portatili per spegnere dispositivi non utilizzati

## Gestione dell'I/O



I processi utente non interagiscono direttamente con il kernel

## Driver dei dispositivi

- **Driver dei dispositivi**
    - Organizzati secondo una pila
    - Ogni livello esporta funzioni verso l'alto
  - **Driver di alto livello**
    - Fornisce l'interfaccia ai processi utente
    - Es: NTFS
  - **Driver intermedio** (o di classe)
    - Implementa **servizi comuni** a dispositivi con caratteristiche simili
    - Es: modulo che implementa il tipo di allocazione dei file
  - **Driver di basso livello**
    - **Interagisce con HAL e controlla il dispositivo**
    - Es: driver di un disco di un particolare modello
- Possibili altri livelli intermedi

14. Windows XP

61

marco lapegna

## Gestore del Plug and Play (PnP)

- Necessaria la combinazione di
  - **Supporto hardware**
    - Capacita' dell'HW di riconoscere nuovi dispositivi e notificare eventi del dispositivi al s.o.
  - **Supporto software (driver PnP)**
    - Diviso in due parti
    - Sviluppata da Microsoft (configura e alloca le risorse)
    - Sviluppata dalle industrie (esporta le info del dispositivo)



Necessarie linee guida per lo sviluppo dei driver

14. Windows XP

62

marco lapegna

## Gestore dell'alimentazione

- **L'utente specifica le politiche** di utilizzo
  - Es: durata batteria vs prestazioni
- **Il gestore dell'alimentazione implementa le politiche**
- **Definisce:**
  - **4 livelli di alimentazione per i dispositivi**

D0, D1, D2, D3  
←—————→  
Piena alimentazione                      Dispositivo spento
  - **5 livelli di alimentazione del sistema**

S0, S1, S2, S3, S4, S5  
←—————→  
Piena alimentazione                      Standby                      sistema spento

14. Windows XP

63

marco lapegna

## Interprocess Communication

- **Strumenti tradizionali (orientati ai dati)**
  - Segnali
  - Pipes
  - Code di messaggi
  - Memoria condivisa
- **Strumenti orientati agli oggetti e alle procedure**
  - Chiamate di procedura locali e remote (RPC)
  - Microsoft COM
  - Clipboard
  - Caratteristica drag-and-drop della GUI

14. Windows XP

64

marco lapegna



## Esempio: Local and Remote Procedure Calls

- Segnali, pipe, code di messaggi e memoria condivisa permettono la **comunicazione di dati** tra processi
- LPC e RPC permettono ad un processo di comunicare con altri processi **invocando funzioni** eseguiti da questi ultimi
- **Local Procedure Call** se i processi risiedono sulla stessa macchina
- **Remote Procedure Call** se i processi risiedono su macchine differenti

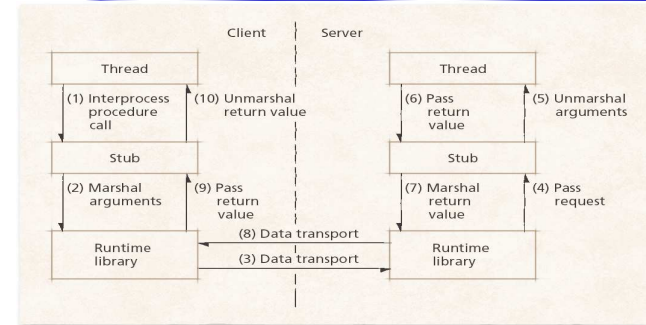


14. Windows XP

65

marco lapegna

## Implementazione



- Una RPC fa uso di **uno stub** che prepara i dati per la spedizione e traduce quelli in arrivo
- E di **una libreria** per spedire e ricevere la richiesta

14. Windows XP

66

marco lapegna

## WOW..... Arriva Windwos Vista



..... Uno "sguardo" a Windows Vista

14. Windows XP

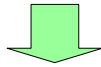
67

marco lapegna



## Da Windows XP a Windows Vista

- Windows Vista e' **l'evoluzione** di WindowsXP
- Non un sistema operativo "rivoluzionario"
- **Adeguamento e ottimizzazione degli executive** per i processi, la memoria, il file system ...
- L'utente apprezzerà l'**interfaccia 3D AERO**, ma anche il kernel ha subito modifiche



Windows Vista non e' WindowsXP con una nuova interfaccia grafica e qualche gadget in piu'

14. Windows XP

69

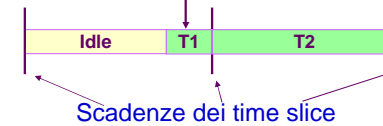
marco lapegna

## Processi e threads

- Lo scheduler di Windows XP contabilizza l'uso della CPU sulla base di un intervallo di clock multiplo di un tick (1 ms)
  - Tipicamente 10-15ms risoluzione
- **Soluzione non sempre equa**

### Esempio

T1 e T2 entrano nella coda ready; T1 inizia ma usa la CPU solo per una frazione ridotta del time slice



- **Inoltre** eventuali **cambi di contesto** (dovuti ad interruzioni) vengono contabilizzati a carico del thread

14. Windows XP

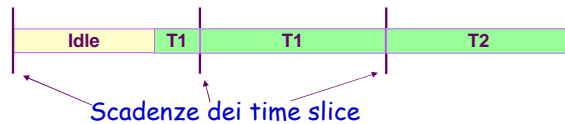
70

marco lapegna

## Miglioramenti in Windows vista

- Ogni thread acquisisce la CPU per **almeno un turno completo**

T1 e T2 entrano nella coda ready; T1 inizia e usa la CPU per almeno un intero intervallo



- **Inoltre**  
Uso della CPU da parte dei threads al netto delle interruzioni

Contabilizzazione piu' accurata

14. Windows XP

71

marco lapegna

## Classe di scheduler Multimedia

- Nuovo servizio che **da precedenza** a i thread delle **applicazioni multimediali**
  - Usato in **Windows Media Player 11**
- I thread si "**dichiarano multimediali**":
  - Viene definita la prioritá'
- Thread multimediali possono prendere fino **all' 80%** del tempo di CPU



14. Windows XP

72

marco lapegna

## Allocazione memoria al kernel (Windows XP)

- Allocazione dello spazio di indirizzamento virtuale del sistema
  - Partizioni di dimensioni fisse
  - Allocated al momento del boot sulla base di informazioni del registro



### PROBLEMI

- Boot piu' lento
- Funzionalita' caricate in memoria anche se non usate
- Reboot in caso di modifica di configurazione

## Allocazione memoria al kernel (Windows Vista)

- Allocazione dello spazio di indirizzamento virtuale del sistema in Windows Vista
  - Partizioni di dimensioni variabili
  - Allocazione on demand della tabella delle pagine del kernel



### VANTAGGI

- Maggiore flessibilita'
- Maggiore efficienza

## Un problema con la memoria virtuale

- La memoria virtuale di Windows XP
  - conserva in memoria centrale le pagine delle applicazioni piu' usate nell'intervallo di tempo recente (politica LRU)
  - Porta sul disco nell'area di swap quelle meno usate quando la memoria scarseggia
  - Riporta in memoria l'applicazione quando viene riutilizzata



### PROBLEMI

- La memoria centrale e' volatile (si perde l'informazione delle pagine usate se si spegne il sistema)
- Il meccanismo funziona bene se si usa di continuo un numero limitato di applicazioni
- Spesso la memoria centrale non contiene le pagine "migliori"

## SuperFetch

- Tecnologia che precarica al momento del boot e tiene sempre in memoria centrale le pagine delle applicazioni piu' usate
- Scenari sofisticati
  - Uso nel week end / uso giorni lavorativi
  - Uso al mattino / uso al pomeriggio
- Differenti livelli di prioritita'
- Esempio:
  - Lun-Ven l'utente dopo il boot usa Outlook, Word ed Excel
  - Sab-Dom l'utente dopo il boot usa Flight Simulator



SuperFetch precarica automaticamente la memoria di conseguenza

## Implementazione di SuperFetch

- SuperFetch include sia il **modo kernel** sia il **modo utente**
- Viene creata un'area sul disco per conservare uno "warm state memory"
- analizza le azioni effettuate dall'utente e si **appunta le applicazioni avviate con più frequenza**

### Svantaggio

- Microsoft consiglia **almeno 1 GB di memoria centrale** per osservare una maggiore reattività dei programmi

14. Windows XP

77

marco lapegna

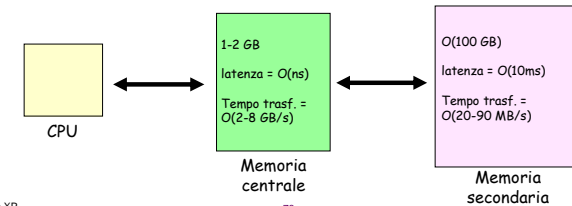
## Una alternativa: ReadyBoost

- La tecnologia SuperFetch si basa su una **considerazione**:

La **memoria centrale** non è solo il supporto da cui la CPU preleva i dati e le istruzioni, ma anche una **cache per il supporto di memorizzazione secondaria**



Esaltazione del ruolo di cache della memoria centrale



14. Windows XP

78

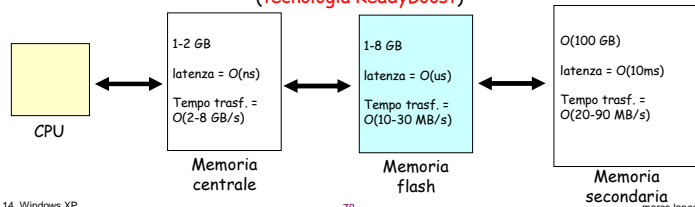
marco lapegna

## Una alternativa: ReadyBoost

- **Problema:** la memoria centrale è ideale per il caching dei dati ma è **costosa e non sempre è possibile aggiungerla** ad un sistema
- **Soluzione:** le **memorie esterne flash** sono più economiche della memoria centrale e al tempo stesso più veloci del disco nel caso di accesso casuale



usare le **memorie flash** come **cache** tra il disco e la memoria centrale  
(tecnologia ReadyBoost)



14. Windows XP

79

marco lapegna

## Una alternativa: ReadyBoost

- Supportati diversi tipi di dispositivi: chiavi USB, Secure Digital card, Compact Flash, PCI express
- Dati sono compressi (tipicamente 2x)
- Dati non permanenti cifrati con chiave casuale ad ogni boot

14. Windows XP

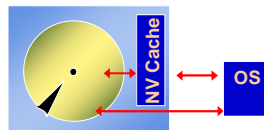
80

marco lapegna

## Hybrid Hard Drives e Ready Drive

- Recentemente sono apparsi sul mercato gli **Hybrid Hard Drive** (H-HD) che includono una **memoria cache non volatile** (NV cache) tipicamente di 256 MB - 1 GB

- Tale memoria puo' essere usata per conservare dati SuperFetch, dati relativi al boot, info sui file ibernati



Il s.o. riconosce automaticamente questo tipo di dischi e gestisce l'NV Cache in modo da rendere i PC con H-HD più efficienti e veloci (**tecnologia Ready Drive**)

- Vantaggi**
  - Minor usura del disco
  - Minor consumo di energia
  - Migliori prestazioni anche al momento del boot

14. Windows XP

81

marco lapegna

## Implementazione

- ReadyBoost e ReadyDrive enfatizzano il **ruolo delle cache** al fine di ottenere **migliori prestazioni**
- L'efficacia di tali tecnologie dipende fortemente da come si riesce a prevedere quali saranno i **dati piu' utili** in un prossimo futuro
- Analogamente al problema degli algoritmi di paginazione, tali tecnologie usano **strategie LRU-like**

14. Windows XP

82

marco lapegna

## File system

- Inizialmente per Windows Vista era previsto un nuovo file system **WinFS** costruito sulla base di NTFS
- La **complessita' del progetto** ha reso necessario pero' un rinvio del rilascio e a oggi ancora non si sa se verra' implementato nelle future versioni
- Windows Vista e' basato quindi su una **evoluzione di NTFS** di Windows XP
- La **principale differenza e'**:
  - NTFS 5 (Windows XP) transazionale solo per i metadati
  - NTFS 6 (Windows Vista) transazionale per tutte le operazioni

14. Windows XP

83

marco lapegna

## Miglioramenti all'I/O

- Possibilita' di **cancellazione di richieste di I/O** sincrone pendenti
  - Esempio: con XP non e' possibile cancellare la funzione open se non e' ancora completata → attesa in caso di richiesta di una pagina web e la rete cade
- Minore interferenza** dei processi in background (antivirus, deframmentazione, ...) con i processi "critici"
  - Possibilita' di riservare una percentuale di I/O processi critici (es. Per Windows Media Player)

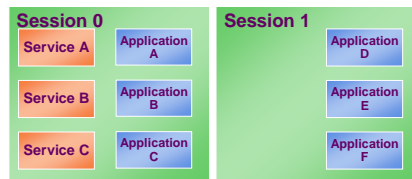
14. Windows XP

84

marco lapegna

## Miglioramenti alla sicurezza

- Con Windows XP i principali servizi del SO vengono eseguiti con i **permessi del primo utente** che si collega (sessione 0)
  - Nomi di programmi creati dalla console possono confliggere con i servizi del SO
  - Servizi che vengono eseguiti con permessi da amministratore possono danneggiare il sistema in caso di malfunzionamento



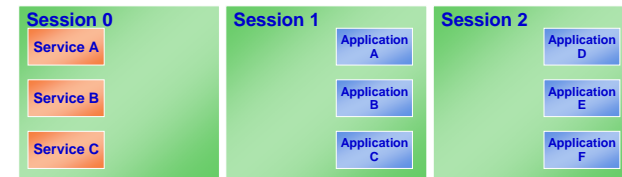
14. Windows XP

85

marco lapegna

## Isolamento della sessione 0

- In Windows Vista, il primo utente che si collega viene eseguito in una **sessione differente** e non può connettersi alla sessione 0
  - Elimina collisioni di nomi
  - Più difficile interferire con i servizi del sistema



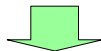
14. Windows XP

86

marco lapegna

## Altre caratteristiche di Windows Vista

- Nuove primitive di sincronizzazione
- Nuove strutture dati per la gestione di thread
- Nuova procedura di boot basata su database
- Servizi di identificazione dei thread
- Processi "protetti" per applicazioni multimediali
- Supporti per dischi di grandi dimensioni
- Chkdsk e dimensionamento on line dei volumi NTFS
- Nuovi meccanismi per comunicazione tra processi mediante meccanismi di Advanced Local Procedure Call
- .....



Non un sistema operativo rivoluzionario  
ma una **significativa evoluzione** di Windows XP

14. Windows XP

87

marco lapegna