Esame del Sistema Operativo Windows

Parte 1 - Indice

Esame del Sistema Operativo Windows
Genesi - 1

- **MS-DOS**
  - Mono-utente, modalità *command line*, inizialmente basato sul modello CP/M
    - 1981 : 1.0 (8 kB) → PC IBM 8088 (16 *bit*)
    - 1986 : 3.0 (36 kB) → PC IBM/AT (i286 @ 8 MHz, ≤ 16 MB)
- **Windows** 1ª generazione
  - Modalità GUI, ma solo come rivestimento di MS-DOS
  - Interfaccia copia del modello Lisa di Apple
    - 1990 - 1993 : 3.0, 3.1, 3.11 → i386 (32 *bit*)

Esame del Sistema Operativo Windows
Genesi - 2

- **Windows** 2ª generazione
  - Vero e proprio S/O multiprogrammato, ma sempre mono-utente, FS su modello FAT
    - 1995 : Windows 95 (MS-DOS 7.0)
    - 1998 : Windows 98 (MS-DOS 7.1)
      - Nucleo a procedure non rientranti (incapaci di consentire più esecuzioni simultanee) → accesso a nucleo protetto da semaforo a mutua esclusione → scarsissimi benefici di multiprogrammazione
      - ¼ dello spazio di indirizzamento di processo (4 GB totali) condiviso R/W con gli altri processi; ¼ condiviso R/W con il nucleo → scarsissima integrità dei dati critici

Modeste modifiche

    - 2000 : Windows Me (ancora MS-DOS)

Esame del Sistema Operativo Windows
Genesi - 3

- **Windows** 3ª generazione
  - Progetto NT: abbandono della base MS-DOS (con architettura a 16 *bit*); enfasi su sicurezza ed affidabilità; nuova generazione di FS (`ntfs`)
    - 1993 : Windows NT 3.1 → fiasco commerciale per la mancanza di programmi di utilità
    - 1996 : Windows NT 4.0 → reintroduzione di interfaccia e programmi Windows 95
      - Scritto in C e C++ per massima portabilità, ma di grande complessità (16 M linee di codice!)
      - Molto superiore a Windows 95, ma privo di supporto per "*plug-and-play*", gestione batterie e emulatore MS-DOS

Esame del Sistema Operativo Windows
Genesi - 4

- **Windows** 3ª generazione (segue)
  - Architettura di NT 3.1 a *microkernel* e modello *client-server*: la maggior parte dei servizi incapsulata in processi di sistema eseguiti in modo utente ed offerti ai processi applicativi in modalità a scambio messaggi
  - Elevata portabilità (dipendenze localizzate nel nucleo) ma scarsa velocità (poca esecuzione in modo privilegiato)
  - Architettura di NT 4.0 a nucleo monolitico: servizi di sistema riposizionati entro il nucleo

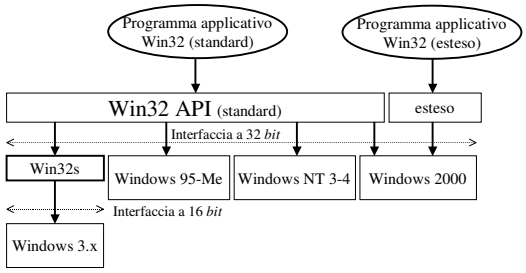
Esame del Sistema Operativo Windows
Genesi - 5

- **Windows** 3ª generazione (segue)
  - 1999 : Windows 2000 (alias di NT 5.0)
    - Il S/O esegue in modo nucleo, separato da quello dei processi utente, il cui spazio di indirizzamento è però interamente privato
    - Include supporto per periferiche rimuovibili ("*plug-and-play*"), per internazionalizzazione (unica versione configurabile per lingua nazionale) ed alcune migliore ad `ntfs`
    - MS-DOS completamente rimpiazzato da una *shell* di comandi che ne replica ed estende le funzionalità
    - Enorme complessità: oltre 29 M linee di codice C!

Esame del Sistema Operativo Windows
Interfaccia di programmazione - 1

- Basato su principio speculare a quello adottato da UNIX e Linux
  - Interfaccia di sistema non pubblica
  - Vasta libreria pubblica di procedure detta Win32 API (*Application Programming Interface*) ad uso del programmatore, ma controllata da Microsoft
  - Alcune procedure includono chiamate di sistema, altre svolgono servizi di utilità eseguiti interamente in modo utente
    - Nessun sforzo di evitare ridondanza o rigore gerarchico

Esame del Sistema Operativo Windows
Interfaccia di programmazione - 2



Esame del Sistema Operativo Windows
Informazioni di configurazione

- Tutte le informazioni vitali di configurazione del sistema sono raccolte in una specie di FS detto **`registry`**, salvato su disco in file speciali (**`hives`**)
  - *Directory* → *key*
  - *File* → *entry* = {nome, tipo, dati}
- 6 directory principali con prefisso **`HKEY_`**
  - Per esempio: **`HKEY_LOCAL_MACHINE`**, con *entry* descrittive dell'*hardware* e delle sue periferiche (**`HARDWARE`**), dei programmi installati (**`SOFTWARE`**) e con informazioni utili per l'inizializzazione (**`SYSTEM`**)

Esame del Sistema Operativo Windows
Architettura di sistema - 1

- Sistema strutturato su 2 livelli gerarchici
  - Nucleo a struttura monolitica, che esegue in modo privilegiato
    - Dipendenze dalla scheda madre dello specifico elaboratore (registri, indirizzi di periferiche, vettore delle interruzioni, orologi, accesso al BIOS) isolate in un livello detto **`HAL`** (*hardware abstraction layer*) → insieme standard di servizi
      - Recentemente affiancato da un interfaccia di maggior potenza e velocità detto **`DirectX`**
  - Sottosistemi d'ambiente, implementati come processi che eseguono in modo normale

Esame del Sistema Operativo Windows
Architettura di sistema - 2

- Su **`HAL`** poggia un livello detto **`kernel`**, che eleva il livello di astrazione dei servizi **`HAL`**
  - Gestione della concorrenza (ordinamento, prerilascio, salvataggio e ripristino dei contesti)
  - Gestione degli "oggetti di controllo", associati a tutte le entità attive del sistema (processi e servizi associati alle interruzioni)
    - Oggetto ***D**eferred **P**rocedure **C**all*: racchiude la parte meno urgente di un servizio di interruzione ed esegue in modo nucleo
    - Oggetto ***A**synchronous **P**rocedure **C**all*: come DPC, ma esegue in modo normale
  - Gestione degli "oggetti di ordinamento", associati a tutte le entità passive (semafori, eventi, orologi) usate per la sincronizzazione delle entità attive

Esame del Sistema Operativo Windows
Architettura di sistema - 3

- Il livello più alto del S/O vero e proprio è detto **`executive`**, il quale è suddiviso in 10 aggregati di procedure funzionalmente legate

`Object manager`: gestisce tutti gli oggetti creati dal S/O, allocando loro memoria virtuale (1)
`I/O manager`: gestisce tutti i dispositivi, incluse le partizioni di disco
`Process manager`: gestisce tutte le entità concorrenti del sistema
`Memory manager`: gestisce la memoria virtuale con modalità "*page-on-demand*"
`Cache manager`: gestisce in RAM una *cache* di blocchi di disco

Esame del Sistema Operativo Windows
Architettura di sistema - 4

- Solo ⬭ e ⬬ sono componenti attive, ma tutte eseguono in modo nucleo

`Plug-and-play manager`: viene informato di tutte le periferiche connesse al sistema per associarvi il relativo gestore (2)
`Power manager`: cerca di contenere il consumo energetico del sistema
`Configuration manager`: gestisce la **`registry`**
`Security manager`: si occupa dell'esecuzione delle politiche di sicurezza richiesti per applicazioni riservate
`Local procedure call manager`: fornisce meccanismi efficaci per la comunicazione tra le componenti attive del sistema

Esame del Sistema Operativo Windows
Architettura di sistema - 5

- Del livello **`executive`** fa parte anche il GDI, inizialmente in spazio di utente in NT 3.x
  - Di gran lunga la sua componente più grande
  - Ricollocata in modo nucleo per aumentarne le prestazioni
- **`kernel`** ed **`executive`** sono raccolti in un unico eseguibile (**`ntoskrnl.exe`**)
- **`HAL`** fornito come libreria condivisa raccolta in un unico *file* (**`hal.dll`**)
- Gestori delle periferiche caricati dinamicamente e registrati in **`registry`** via **`Configuration manager`**

Esame del Sistema Operativo Windows
Architettura di sistema - 6

- Durante l'esecuzione, il sistema crea, manipola e distrugge "oggetti" interni, nessuno dei quali permane tra due accensione successive
  - Un oggetto per ogni entità sia attiva che passiva
    - Tutti gli oggetti hanno alcuni metodi comuni
  - Gli oggetti sono descrittori (in RAM) delle relative entità
    - Alcuni possono essere temporaneamente posti su disco
- Il **`kernel`** mantiene una tabella degli oggetti
  - 29 *bit* per puntatore all'oggetto + 3 *bit* come *flag*
  - 32 *bit* per i diritti associati alle operazioni sull'oggetto
- L'**`object manager`** suddivide gli oggetti in categorie (*directory*) specifiche

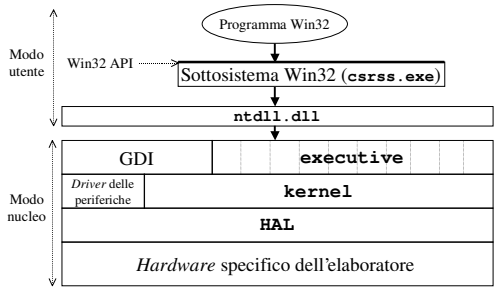
Esame del Sistema Operativo Windows
Architettura di sistema - 7

- In spazio di utente sono disponibili 3 ulteriori categorie di componenti di sistema
  - **`DLL`** (*Dynamic Link Libraries*), che raccolgono specifiche procedure di libreria in gruppi visibili ai e condivisi dai vari programmi
    - Ogni processo utente include chiamate parametriche a specifiche DLL, invece del codice delle procedure richieste
  - Sottosistemi d'ambiente (**`.exe`**), che forniscono ciascuno uno specifico interfaccia di programmazione, di cui il principale è il Win32 API (e gli altri 2 sono inutili!)
  - Processi di servizio

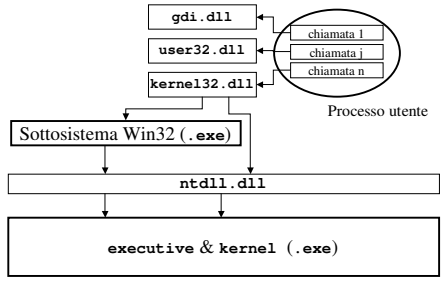
Esame del Sistema Operativo Windows
Architettura di sistema - 8

- In complesso, oltre 800 **`DLL`**, per più di 13000 procedure invocabili dai processi utente. Tra esse:
  - **`user32.dll`** : invocate in modo utente per i servizi GUI
  - **`gdi32.dll`** : invocate in modo utente per tutti i servizi grafici sottostanti al GUI
  - **`kernel32.dll`** : invocate in modo utente per tutti gli altri servizi
  - **`ntdll.dll`** : il vero interfaccia di sistema tra modo utente e modo nucleo (**`executive`** e **`kernel`**)
  - **`hal.dll`** : eseguite in modo privilegiato per accedere all'hardware specifico dell'elaboratore

Esame del Sistema Operativo Windows
Architettura di sistema - 9



Esame del Sistema Operativo Windows
Architettura di sistema - 10



Esame del Sistema Operativo Windows
Gestione dei processi - 1

- *Job* = {processi gestiti come singola unità}
- Processo = possessore di risorse, con una o più *thread*
  - ID unico, 4 GB di spazio di indirizzamento (2 in modo utente e 2 in modo nucleo), inizialmente con singola *thread*, molto simile al processo UNIX
- *Thread* = flusso di controllo ordinato dal nucleo
  - Esegue per conto e nell'ambiente del processo (che non ha stato di avanzamento), con ID localmente unico, 2 *stack* (1 per modo)
- *Fiber* = suddivisione di *thread*, ignota al nucleo
  - Esegue nell'ambiente della *thread*, gestita interamente a livello di sottosistema Win32

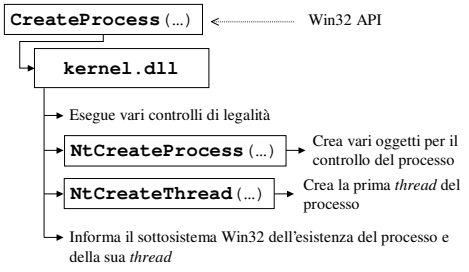
Esame del Sistema Operativo Windows
Gestione dei processi - 2

- Le *thread* hanno vari modi per comunicare
  - *Pipe* : canali bidirezionali come in UNIX/Linux, a sequenza di *byte* senza struttura, oppure per messaggi (sequenze con struttura)
  - *Mailslot* : canali unidirezionali, anche su rete
  - *Socket* : come *pipe* ma per comunicazioni remote
  - RPC (chiamata di procedura remota) : per invocare procedure nello spazio di altri processi e riceverne il risultato localmente
  - Condivisione di memoria : usando (porzioni di) *file* mappato in memoria

Esame del Sistema Operativo Windows
Gestione dei processi - 3

- Le *thread* hanno vari modi per sincronizzarsi
  - Semafori binari (**mutex**) o contatori
  - Sezioni critiche, limitate allo spazio di indirizzamento della *thread* che la crea
  - Eventi di 2 tipi
    - A *reset* manuale, che rilascia più *thread* sino ad un esplicito *reset* che cancella l'evento
    - A *reset* automatico, che rilascia solo una *thread* e poi cancella l'evento

Esame del Sistema Operativo Windows
Gestione dei processi - 4



Esame del Sistema Operativo Windows
Ordinamento dei processi - 1

- Ordinamento con prerilascio a priorità, non gestito da una entità attiva di sistema, ma effettuato da azioni esplicite della *thread* in esecuzione
  - Nel sospendersi in attesa di una risorsa occupata o nell'inviare un segnale di sincronizzazione
    - L'esecuzione è già in modo nucleo
  - Al completamento del proprio quanto di tempo
    - L'esecuzione passa in modo nucleo
- Implicitamente attribuito alla *thread* corrente
  - Al verificarsi di eventi asincroni (interruzione, allarme)

Esame del Sistema Operativo Windows
Ordinamento dei processi - 2

- 6 classi di priorità per processo
  - `Realtime`, `high`, `above-normal`, `normal`, `below-normal`, `idle`
- 7 classi di priorità per *thread*
  - `Time-critical`, `highest`, `above-normal`, `normal`, `below-normal`, `lowest`, `idle`
- 32 livelli di priorità (31..0), ciascuno associato ad una coda di *thread* pronte
  - *Thread* non distinte per processo di appartenenza
  - 31..16 priorità di sistema; 15..0 priorità ordinarie
- Ricerca per priorità decrescente, selezione dalla testa della coda ed attivazione del quanto

Esame del Sistema Operativo Windows
Ordinamento dei processi - 3

- Ciascuna *thread* ha una priorità base (iniziale) ed una corrente (che varia in esecuzione)
- La priorità corrente si eleva quando la *thread*
  - Completa un'operazione di I/O
    - Per maggior utilizzazione delle periferiche
  - Ottiene un semaforo o riceve un segnale d'evento
    - Per miglior risposta dei processi interattivi
- La priorità corrente decade ad ogni quanto consumato
- Usa una tecnica brutale per mitigare il problema di inversione di priorità
  - Una *thread* pronta non selezionata per una certa durata riceve incremento di priorità per 2 quanti

Esame del Sistema Operativo Windows
Inizializzazione - 1

- Attivazione del programma di *boot* come in Linux
- Lettura della struttura di FS, localizzazione ed esecuzione del *file* **`ntldr`** che carica Windows 2000
  - Il FS può avere struttura FAT-16, FAT-32, **`ntfs`**
- Lettura del *file* di configurazione **`Boot.ini`**, caricamento di **`hal.dll`**, **`ntoskrnl.exe`**, **`bootvid.dll`**, lettura di **`registry`** e configurazione delle periferiche
- Attivazione di **`ntoskrnl.exe`** e creazione del gestore di sessione (processo utente nativo **`smss.exe`**)
  - Creazione del *daemon* di *login* (**`winlogon.exe`**)
  - Attivazione del gestore di autenticazione (**`lsass.exe`**)
  - Attivazione del capostipite di tutti i servizi (**`services.exe`**)

Esame del Sistema Operativo Windows
Inizializzazione - 2

- Il *daemon* (**`winlogon.exe`**) usa il programma **`msgina.dll`** per eseguire la sequenza di *login* desiderata
  - Ciò consente di configurarne più le modalità di esecuzione
- Poi preleva da **`registry`** il profilo dell'utente, da cui determina il programma di *shell* da eseguire
  - Generalmente si tratta di **`explorer.exe`**, configurabile tramite **`registry`**