Sistemi Operativi 2003/2004

WINDOWS 2000

Windows NT

Item	Windows 95/98	Windows NT
Full 32-bit system?	No	Yes
Security?	No	Yes
Protected file mappings?	No	Yes
Private addr space for each MS-DOS prog?	No	Yes
Unicode?	No	Yes
Runs on	Intel 80x86	80x86, Alpha, MIPS, ...
Multiprocessor support?	No	Yes
Re-entrant code inside OS?	No	Yes
Plug and play?	Yes	No
Power management?	Yes	No
FAT-32 file system?	Yes	Optional
NTFS file system	No	Yes
Win32 API?	Yes	Yes
Run all old MS-DOS programs?	Yes	No
Some critical OS data writable by user?	Yes	No

Diversità tra Windows 98 e Windows NT

Windows 2000 (1)

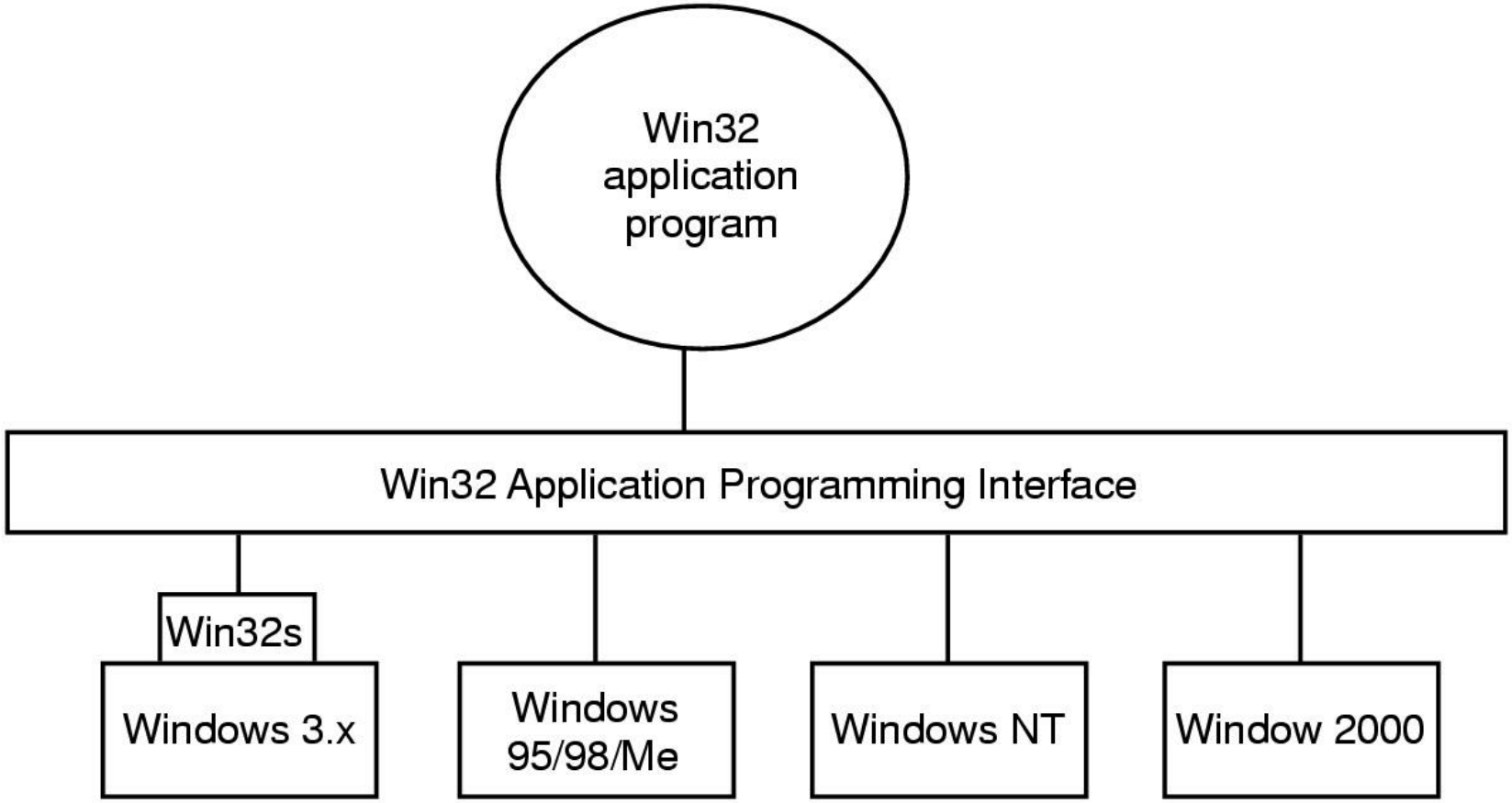
Version	Max RAM	CPUs	Max clients	Cluster size	Optimized for
Professional	4 GB	2	10	0	Response time
Server	4 GB	4	Unlimited	0	Throughput
Advanced server	8 GB	8	Unlimited	2	Throughput
Datacenter server	64 GB	32	Unlimited	4	Throughput

Versioni diverse di Windows 2000

Windows 2000 (2)

Year	AT&T	BSD	MINIX	Linux	Solaris	Win NT
1976	V6 9K					
1979	V7 21K					
1980		4.1 38K				
1982	Sys III 58K					
1984		4.2 98K				
1986		4.3 179K				
1987	SVR3 92K		1.0 13K			
1989	SVR4 280K					
1991				0.01 10K		
1993		Free 1.0 235K			5.3 850K	3.1 6M
1994		4.4 Lite 743K		1.0 165K		3.5 10M
1996				2.0 470K		4.0 16M
1997			2.0 62K		5.6 1.4M	
1999				2.2 1M		
2000		Free 4.0 1.4M			5.8 2.0M	2000 29M

The Win32 Application Programming Interface



Attraverso Win32 API lo stesso programma può essere eseguito sulla maggior parte di versioni Windows

The Registry (1)

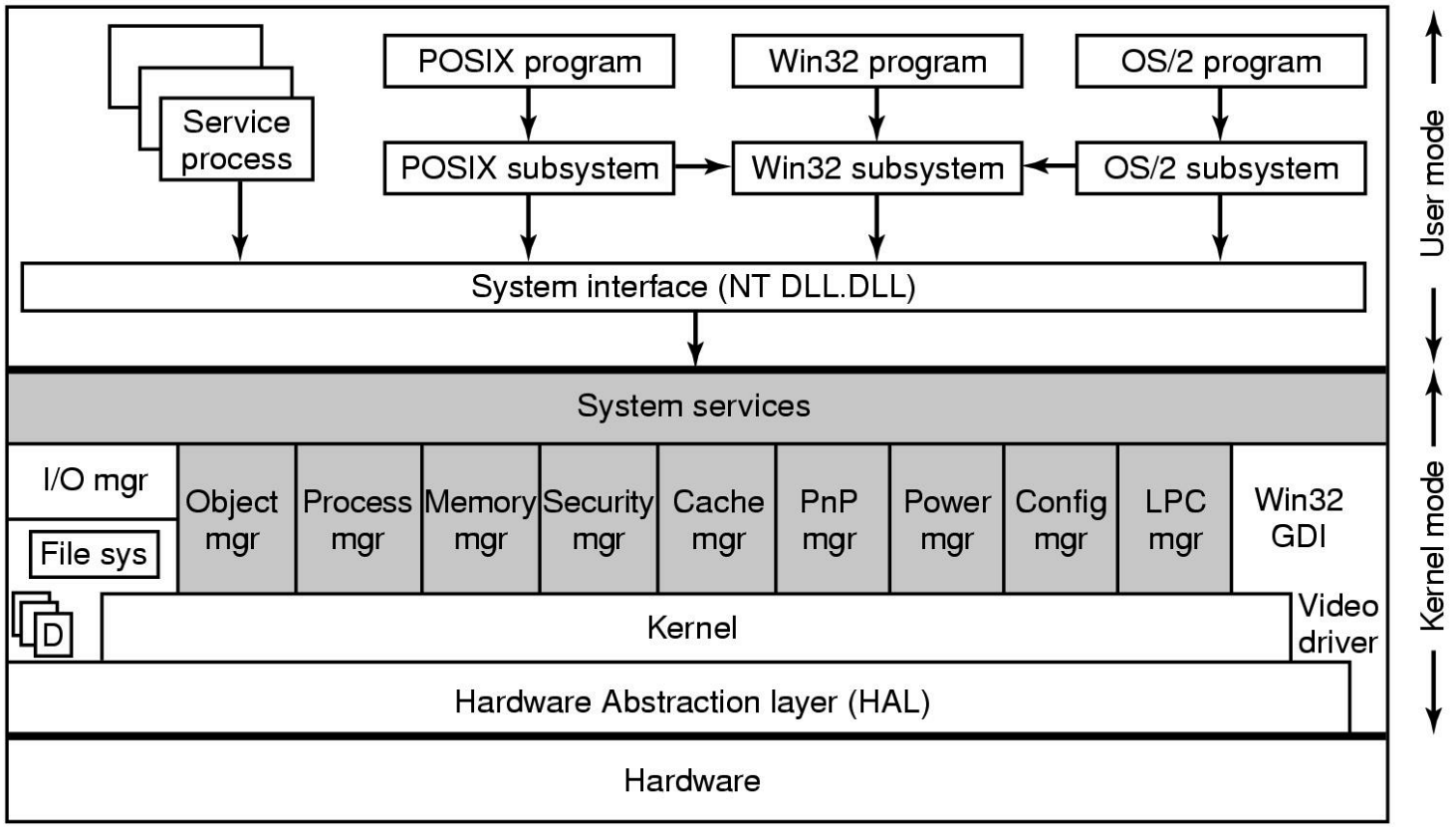
Key	Description
HKEY_LOCAL_MACHINE	Properties of the hardware and software
HARDWARE	Hardware description and mapping of hardware to drivers
SAM	Security and account information for users
SECURITY	System-wide security policies
SOFTWARE	Generic information about installed application programs
SYSTEM	Information for booting the system
HKEY_USERS	Information about the users; one subkey per user
USER-AST-ID	User AST's profile
AppEvents	Which sound to make when (incoming email/fax, error, etc.)
Console	Command prompt settings (colors, fonts, history, etc.)
Control Panel	Desktop appearance, screensaver, mouse sensitivity, etc.
Environment	Environment variables
Keyboard Layout	Which keyboard: 102-key US, AZERTY, Dvorak, etc.
Printers	Information about installed printers
Software	User preferences for Microsoft and third party software
HKEY_PERFORMANCE_DATA	Hundreds of counters monitoring system performance
HKEY_CLASSES_ROOT	Link to HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES
HKEY_CURRENT_CONFIG	Link to the current hardware profile
HKEY_CURRENT_USER	Link to the current user profile

- Chiavi di livello superiore

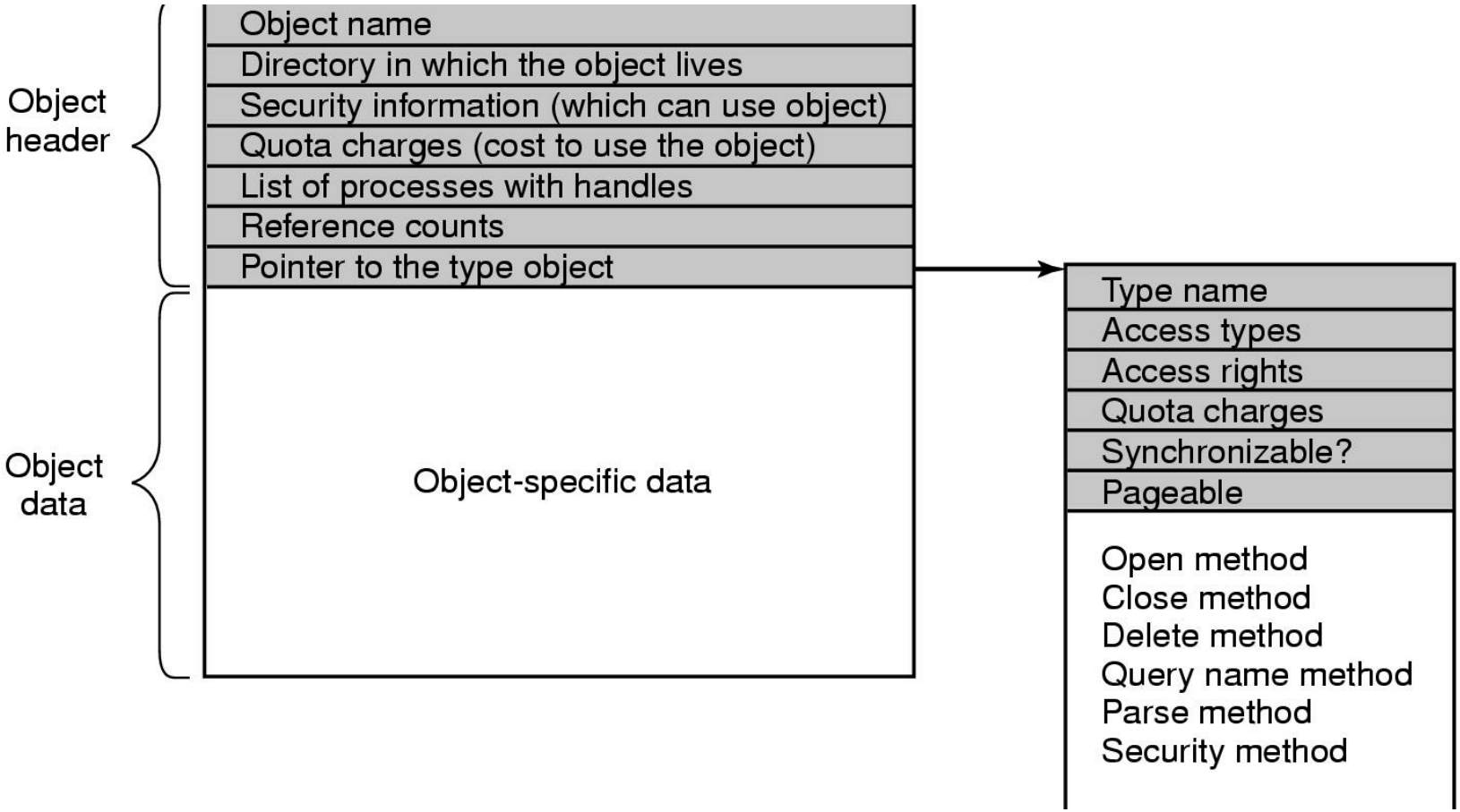
The Registry (2)

Win32 API function	Description
RegCreateKeyEx	Create a new registry key
RegDeleteKey	Delete a registry key
RegOpenKeyEx	Open a key to get a handle to it
RegEnumKeyEx	Enumerate the subkeys subordinate to the key of the handle
RegQueryValueEx	Look up the data for a value within a key

Il sistema operativo



Oggetti


Object header
Object name
Directory in which the object lives
Security information (which can use object)
Quota charges (cost to use the object)
List of processes with handles
Reference counts
Pointer to the type object
Object data
Object-specific data
Type name
Access types
Access rights
Quota charges
Synchronizable?
Pageable
Open method
Close method
Delete method
Query name method
Parse method
Security method

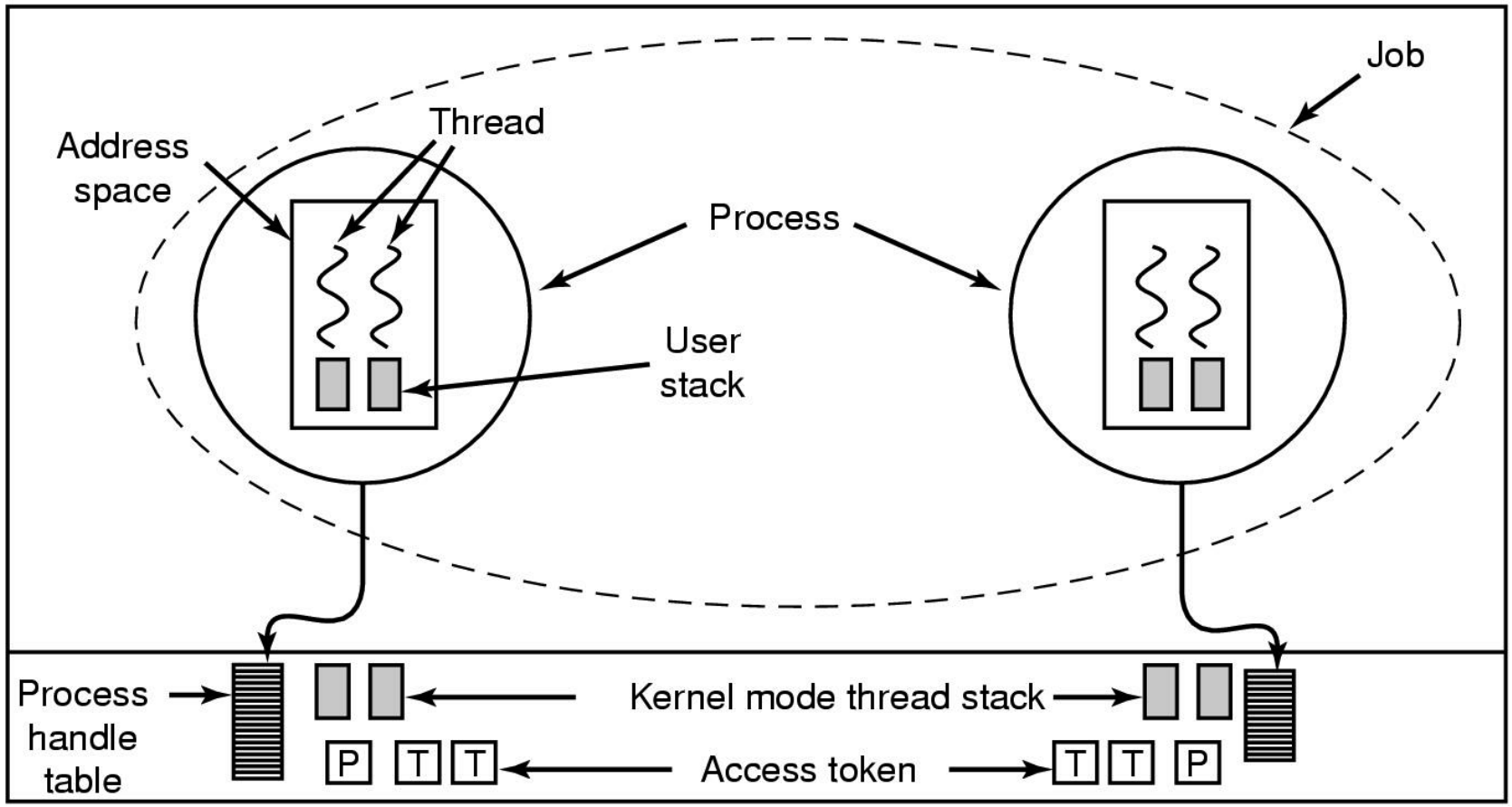
Tipi di oggetto

Type	Description
Process	User process
Thread	Thread within a process
Semaphore	Counting semaphore used for interprocess synchronization
Mutex	Binary semaphore used to enter a critical region
Event	Synchronization object with persistent state (signaled/not)
Port	Mechanism for interprocess message passing
Timer	Object allowing a thread to sleep for a fixed time interval
Queue	Object used for completion notification on asynchronous I/O
Open file	Object associated with an open file
Access token	Security descriptor for some object
Profile	Data structure used for profiling CPU usage
Section	Structure used for mapping files onto virtual address space
Key	Registry key
Object directory	Directory for grouping objects within the object manager
Symbolic link	Pointer to another object by name
Device	I/O device object
Device driver	Each loaded device driver has its own object

Processi e Thread

Name	Description
Job	Collection of processes that share quotas and limits
Process	Container for holding resources
Thread	Entity scheduled by the kernel
Fiber	Lightweight thread managed entirely in user space

Processi e Thread



Processi

- Un processo viene creato quando un altro processo chiama la funzione API Win 32 `CreateProcess`
- Tutti i processi sono creati uguali, anche se esiste una gerarchia implicita nel fatto che al processo chiamante viene restituito un gestore del nuovo processo

Thread

- Ogni processo viene creato con un solo thread
- I thread sono creati con la funzione API Win 32 `CreateThread`
- In W2000 i thread sono gestiti a livello di kernel

Implementazione dei processi

- La funzione `CreateProcess` chiama una procedura in `kernel32.dll` che svolge le seguenti azioni:
  - Esamina del file eseguibile associato (POSIX, OS/2, MS-DOS, W2000) per predisporre il corretto ambiente di esecuzione;
  - Chiama la Syscall `NtCreateProcess` per la creazione delle strutture dati, terminate le operazioni la syscall restituisce il controllo a `kernel32.dll`

Implementazione dei processi

- Chiama la syscall `NtCreateThread` per la creazione del thread iniziale
- Passa i parametri caratteristici del nuovo processo al relativo ambiente d'esecuzione (Win32, MS-Dos, OS/2, ecc.), che provvede ad aggiornare i propri valori
- Il processo (thread) può ora andare in esecuzione

Job, Process, Thread & Fiber Mgmt. API Calls

Win32 API Function	Description
CreateProcess	Create a new process
CreateThread	Create a new thread in an existing process
CreateFiber	Create a new fiber
ExitProcess	Terminate current process and all its threads
ExitThread	Terminate this thread
ExitFiber	Terminate this fiber
SetPriorityClass	Set the priority class for a process
SetThreadPriority	Set the priority for one thread
CreateSemaphore	Create a new semapahore
CreateMutex	Create a new mutex
OpenSemaphore	Open an existing semaphore
OpenMutex	Open an existing mutex
WaitForSingleObject	Block on a single semaphore, mutex, etc.
WaitForMultipleObjects	Block on a set of objects whose handles are given
PulseEvent	Set an event to signaled then to nonsignaled
ReleaseMutex	Release a mutex to allow another thread to acquire it
ReleaseSemaphore	Increase the semaphore count by 1
EnterCriticalSection	Acquire the lock on a critical section
LeaveCriticalSection	Release the lock on a critical section

La creazione dei processi

- Quando il computer viene acceso un programma residente in ROM, legge il contenuto del primo blocco del disco di boot e lo esegue
- Questo settore contiene un programma di bootstrap e una partition table (Master boot record)
- Il bootstrap individua la partizione attiva, legge il boot sector ed esegue il programma ivi presente: il boot

La creazione dei processi

- Il boot individua il file `ntldr,` lo carica in memoria e lo esegue
- `Ntldr` legge il file `boot.ini` che elenca tutte le versioni di `hal.dll` e `ntoskrnl.exe` disponibili e altri parametri relativi al sistema hw.
- Ntldr quindi seleziona i file `hal.dll` e `ntoskrnl.exe` più appropriati e i driver necessari per avviare il sistema e li carica in memoria
- Il controllo viene passato a `ntoskrnl.exe`
- Vengono chiamate le varie componenti dell'esecutivo per provvedere all'inizializzazione delle aree di memoria di loro competenza

La creazione dei processi

- Viene creato il primo processo: session manager (smss.exe) che provvede ad attivare:
  - Il sottosistema Win32 (csrss.exe)
  - Dll
  - Winlogon.exe

La creazione dei processi

- Winlogon.exe provvede a:
  - Identificare l’utente
  - Attivare il processo di autenticazione (lsass.exe)
  - Attivare il processo services.exe che attiva tutti i processi demoni per lo spazio utente, e carica tutti i driver non ancora caricati

Booting Windows 2000

Process	Description
idle	Not really a process, but home to the idle thread
system	Creates smss.exe & paging files; reads registry; opens DLLs
smss.exe	First real proc; much initialization; creates csrss & winlogon
csrss.exe	Win32 subsystem process
winlogon.exe	Login daemon
lsass.exe	Authentication manager
services.exe	Looks in registry and starts services
Printer server	Allows remote jobs to use the printer
File server	Serves requests for local files
Telnet daemon	Allows remote logins
Incoming email handler	Accepts and stores inbound email
Incoming fax handler	Accepts and prints inbound faxes
DNS resolver	Internet domain name system server
Event logger	Logs various system events
Plug-and-play manager	Monitors hardware to see what is out there

File importanti

File	Mode	Fcns	Contents
hal.dll	Kernel	95	Low-level hardware management, e.g., port I/O
ntoskrnl.exe	Kernel	1209	Windows 2000 operating system (kernel + executive)
win32k.sys	Kernel	-	Many system calls including most of the graphics
ntdll.dll	User	1179	Dispatcher from user mode to kernel mode
csrss.exe	User	0	Win32 environment subsystem process
kernel32.dll	User	823	Most of the core (nongraphics) system calls
gdi32.dll	User	543	Font, text, color, brush, pen, bitmap, palette, drawing, etc. calls
user32.dll	User	695	Window, icon, menu, cursor, dialog, clipboard, etc. calls
advapi32.dll	User	557	Security, cryptography, registry, management calls

Gestione della memoria

- Memoria Paginata
- Nessuna segmentazione
- La dimensione delle pagine deve essere multiplo di due e comunque minore di 64K
- Pagine di 4K sul Pentium, 8K e 16K su Itanium
- Ogni processo ha uno spazio virtuale di 4GB

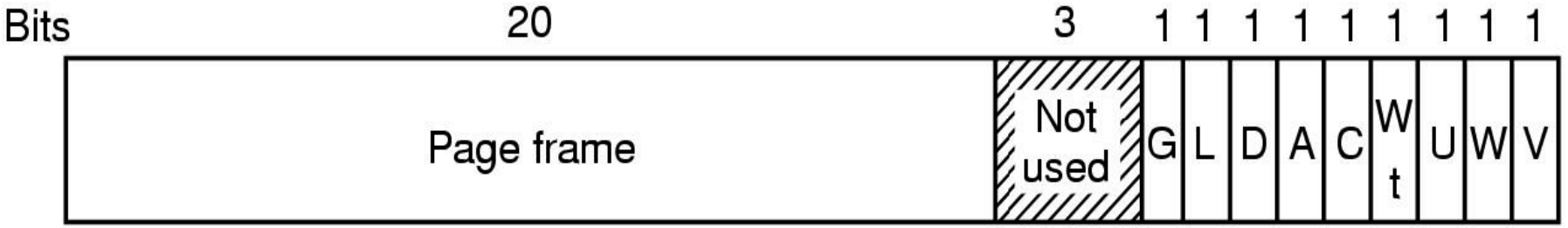
Memoria virtuale

- In W2000 ogni pagina virtuale può trovarsi in tre stati:
  - Free: non è attualmente in uso ed se referenziata provoca un page fault
  - Commited: la pagine è in uso, se è in memoria vi si accede altrimenti page fault
  - Reserved: non è disponibile per essere utilizzata fino a che la “prenotazione” non è esplicitamente rimossa

Strategia di allocazione pagine

- Demand pagin with clustering
  - A fronte di un page fault sono caricate in memoria oltre alla pagina che l'ha creato, 1-8 pagine consecutive
- La procedura di gestione del page fault è standard

Tabella delle pagine

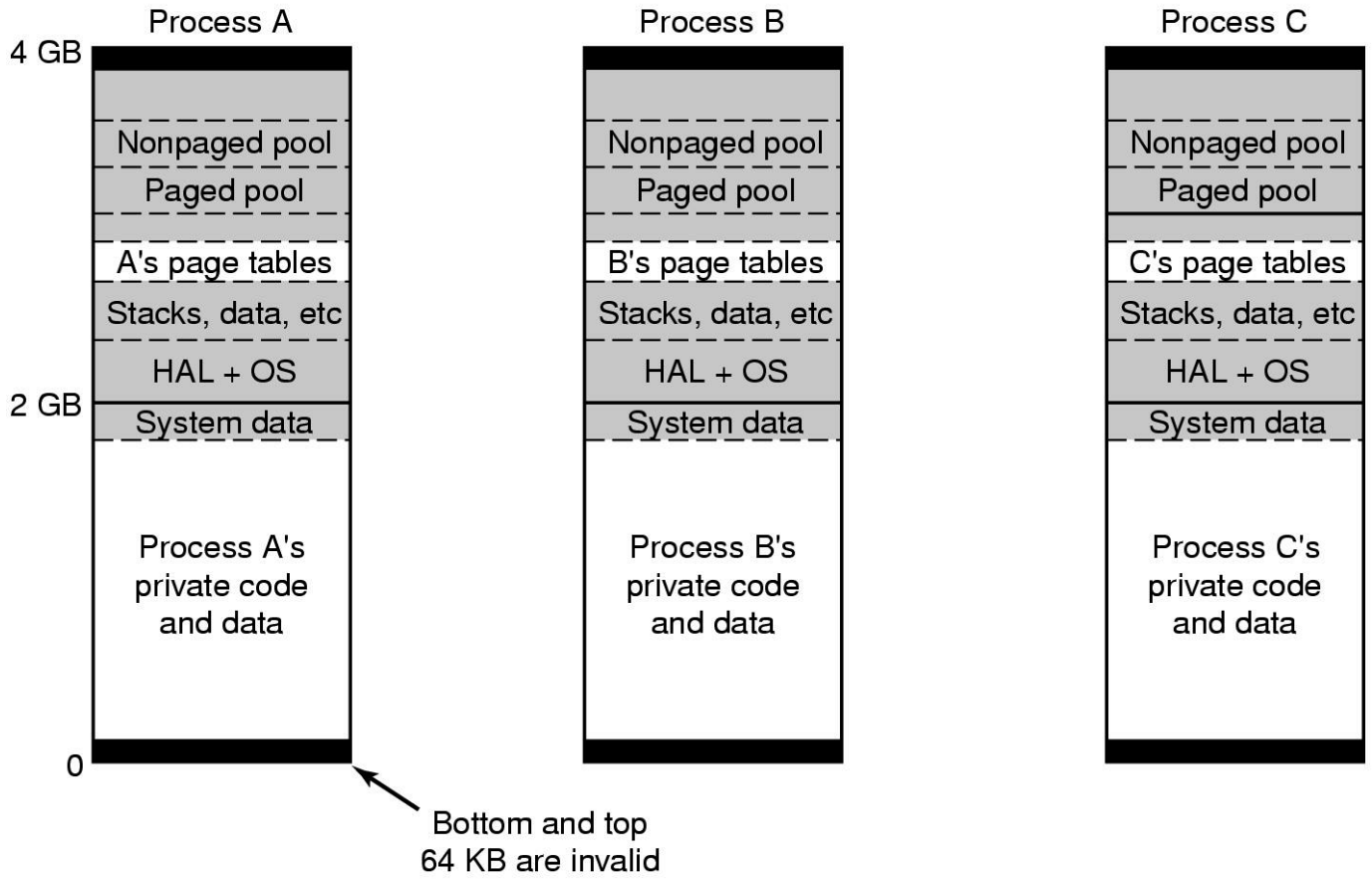


G: Page is global to all processes
L: Large (4-MB) page
D: Page is dirty
A: Page has been accessed

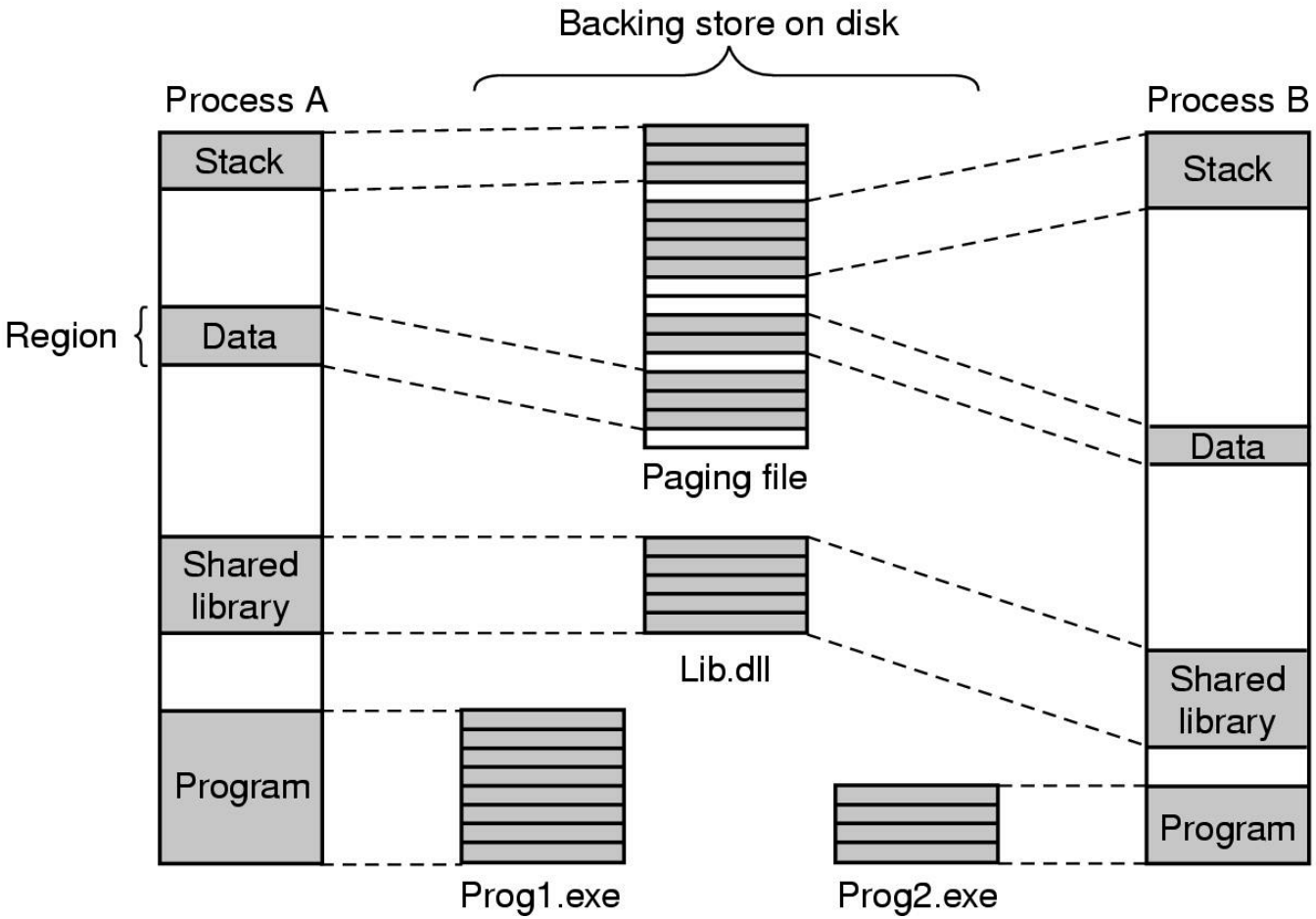
Wt: Write through (no caching)
U: Page is accessible in user mode
W: Writing to the page permitted
V: Valid page table entry

Un elemento della tabella delle pagine sul Pentium

Struttura di un processo (1)



Struttura di un processo (2)



Memoria Virtuale

- Il memory manager opera esclusivamente a livello di processo, il concetto di thread non è noto in questo contesto
- Quando un processo viene avviato il memory manager crea un Virtual address descriptor che contiene tra le altre informazioni:
  - Indicazioni su dove sia contenuto il processo in termini di backing store
  - L’indirizzo della tabella delle pagine

Rimpiazzamento

- Il sistema cerca di mantenere un elevato numero di pagine di memoria libere al fine di soddisfare immediatamente le richieste di nuovo spazio
- Uso del concetto di working set legato ad ogni processo: le pagine di un processo che sono mappate in memoria

Rimpiazzamento

- Il working set è descritto attraverso due parametri: minimum size e maximum size
- Il valore di questi due parametri è fissato dal sysadmin e dipende anche dalla quantità di memoria disponibile
- Maximum size può modificarsi durante l’esecuzione del processo

Rimpiazzamento

- Quando un processo genera un page fault il sistema controlla se il numero di pagine allocate al processo è minore di minum size alloca una nuova pagine
- Se è maggiore della massima attua una politica di rimpiazzamento locale
- Se però un processo pagina troppo frequentemente il suo maximum size può essere modificato dal sistema
- Ogni processo può disporre al massimo di 512 pagine di memoria

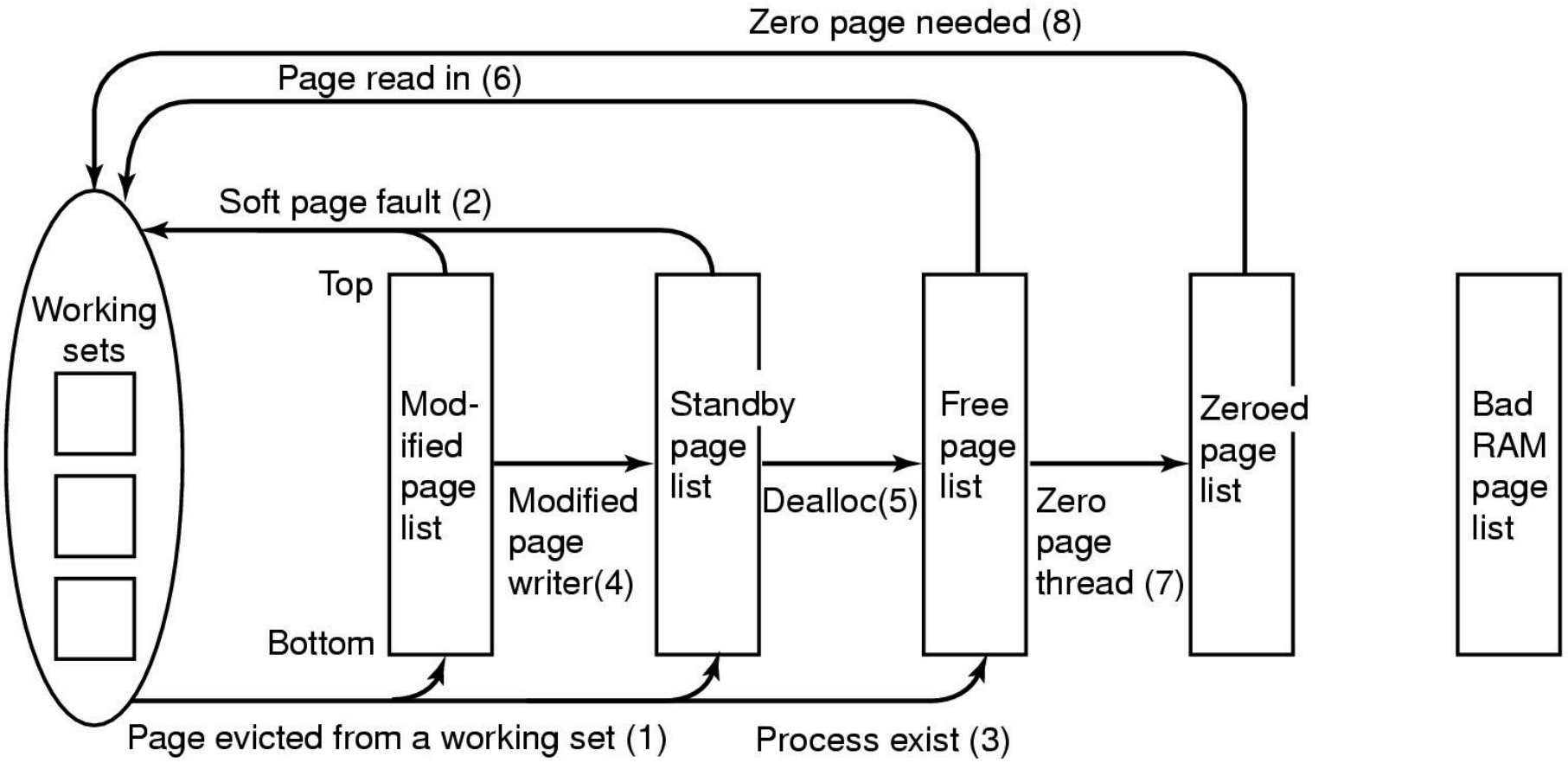
Rimpiazzamento

- Ogni secondo viene eseguito un demone (balance set manager) il cui obiettivo è verificare se vi sono abbastanza pagine libere ed in caso contrario liberarle lanciando in esecuzione un ulteriore demone (WS manager)
- WS manager rimuove la pagine da processi vittima, cioè processi con un numero di pagine superiore a minimum size o con recente scarsa attività

Gestione della free list

- Ogni pagina di memoria centrale si trova in uno o più dei seguenti insiemi:
  - Working set di un processo
  - Standby list
  - Modified list
  - Free list
  - Zeroed list
  - Bad page list

Physical Memory Management (1)

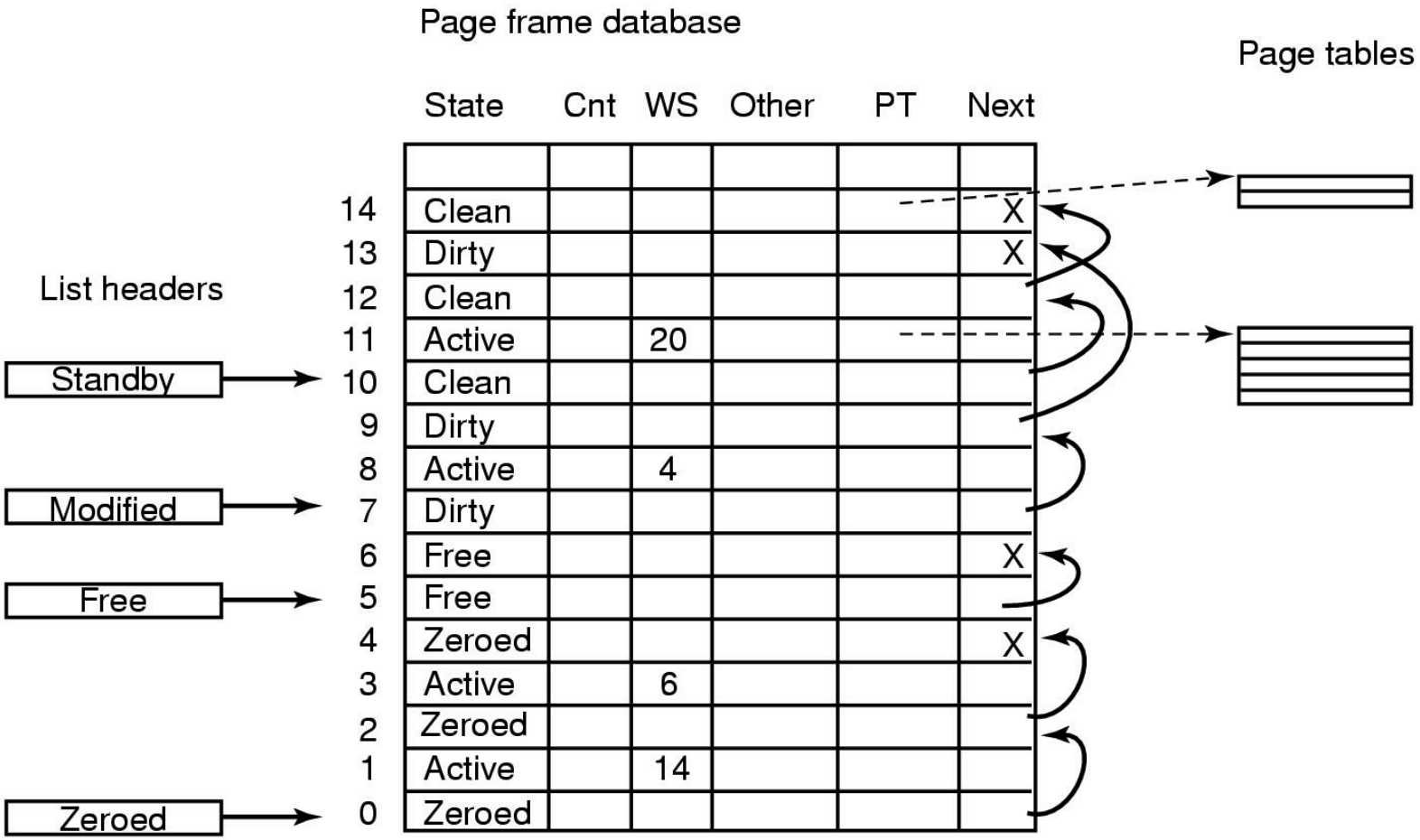


Le diverse page lists e il diagramma delle transizioni

Gestione pagine

- Quando un processo necessità di una nuova pagina la stessa viene recuperata dalla free list e se questa è vuota dalla standby
- Un demone (swapper thread) provvede a spostare pagine da processi poco attivi nelle modified e standby list
- Due demoni provvedono periodicamente a spostare pagine dalla modified list alla standby list

Physical Memory Management (2)



La tabella dei frame

Memory Management System Calls

Win32 API function	Description
VirtualAlloc	Reserve or commit a region
VirtualFree	Release or decommit a region
VirtualProtect	Change the read/write/execute protection on a region
VirtualQuery	Inquire about the status of a region
VirtualLock	Make a region memory resident (i.e., disable paging for it)
VirtualUnlock	Make a region pageable in the usual way
CreateFileMapping	Create a file mapping object and (optionally) assign it a name
MapViewOfFile	Map (part of) a file into the address space
UnmapViewOfFile	Remove a mapped file from the address space
OpenFileMapping	Open a previously created file mapping object

Win32 API per la gestione della memoria virtuale in Windows 2000

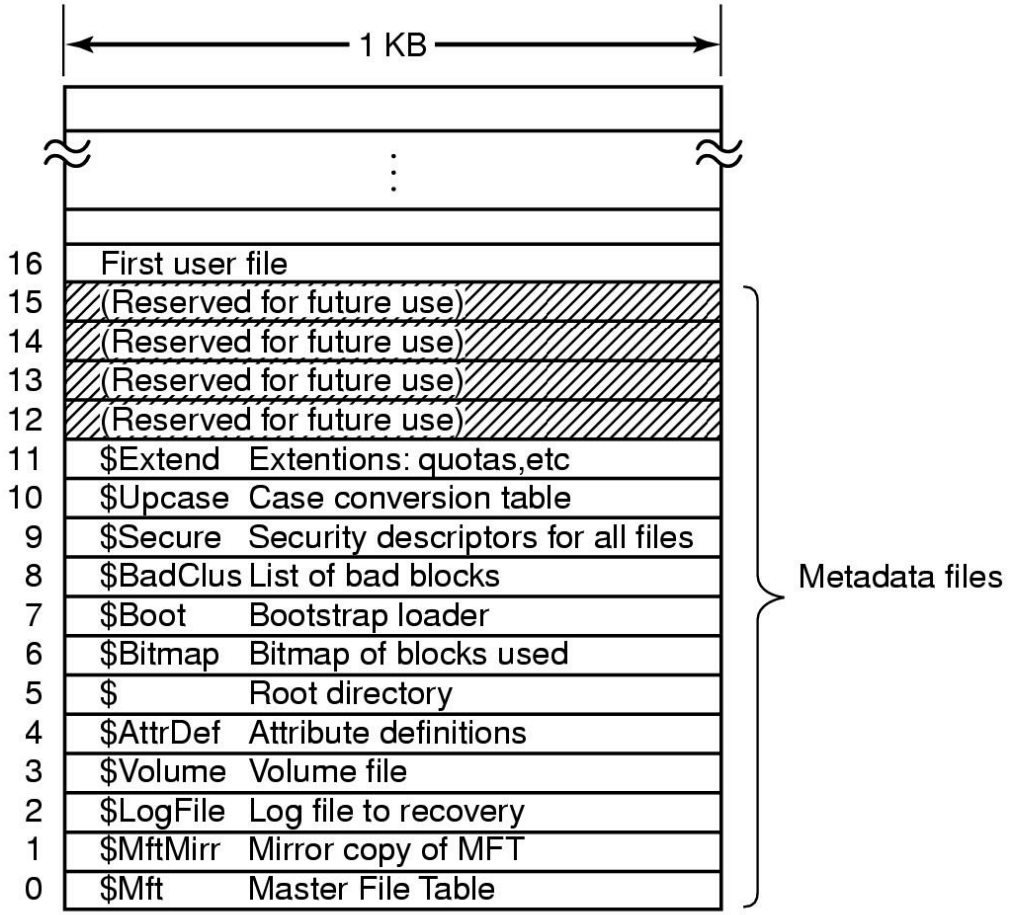
Formati Supportati

- W2000 supporta diversi formati di file system che differiscono tra loro per le modalità con cui gestiscono le informazioni in essi memorizzate:
  - CDFS: per i CD-ROM
  - UDF: per i DVD
  - FAT12, FAT16, FAT32
  - NTFS: file system nativo

NTFS

- I dischi sono divisi in volumi
- Il blocco logico in NTFS viene chiamato cluster
- NTFS indirizza solo i cluster
- La principale struttura dati per ogni volume è la Master File Table (MFT)
- MFT è un array di record la cui dimensione è fissa e pari a 1Kbyte
- La gestione dei blocchi liberi è fatta attraverso bitmap

MFT


1 KB
16 First user file
15 (Reserved for future use)
14 (Reserved for future use)
13 (Reserved for future use)
12 (Reserved for future use)
11 $Extend Extentions: quotas,etc
10 $Upcase Case conversion table
9 $Secure Security descriptors for all files
8 $BadClus List of bad blocks
7 $Boot Bootstrap loader
6 $Bitmap Bitmap of blocks used
5 $ Root directory
4 $AttrDef Attribute definitions
3 $Volume Volume file
2 $LogFile Log file to recovery
1 $MftMirr Mirror copy of MFT
0 $Mft Master File Table
Metadata files

MFT

- Ogni elemento di MFT è costituito da una sequenza di attributi
- Ogni attributo è definito da una coppia: header e valore
- Il numero di attributi che definiscono un record MFT e quindi un file non è fisso

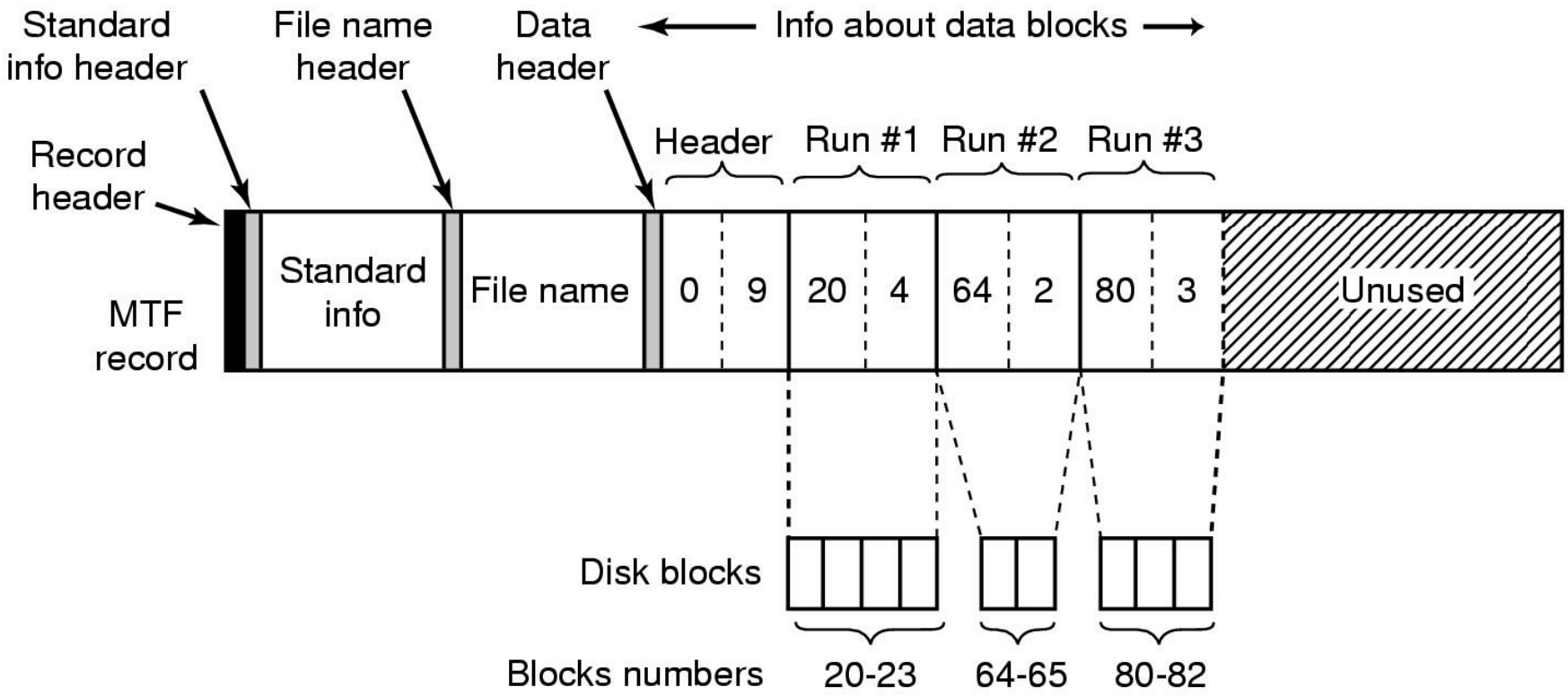
Attributi usati in un record MFT

Attribute	Description
Standard information	Flag bits, timestamps, etc.
File name	File name in Unicode; may be repeated for MS-DOS name
Security descriptor	Obsolete. Security information is now in $Extend$Secure
Attribute list	Location of additional MFT records, if needed
Object ID	64-bit file identifier unique to this volume
Reparse point	Used for mounting and symbolic links
Volume name	Name of this volume (used only in $Volume)
Volume information	Volume version (used only in $Volume)
Index root	Used for directories
Index allocation	Used for very large directories
Bitmap	Used for very large directories
Logged utility stream	Controls logging to $LogFile
Data	Stream data; may be repeated

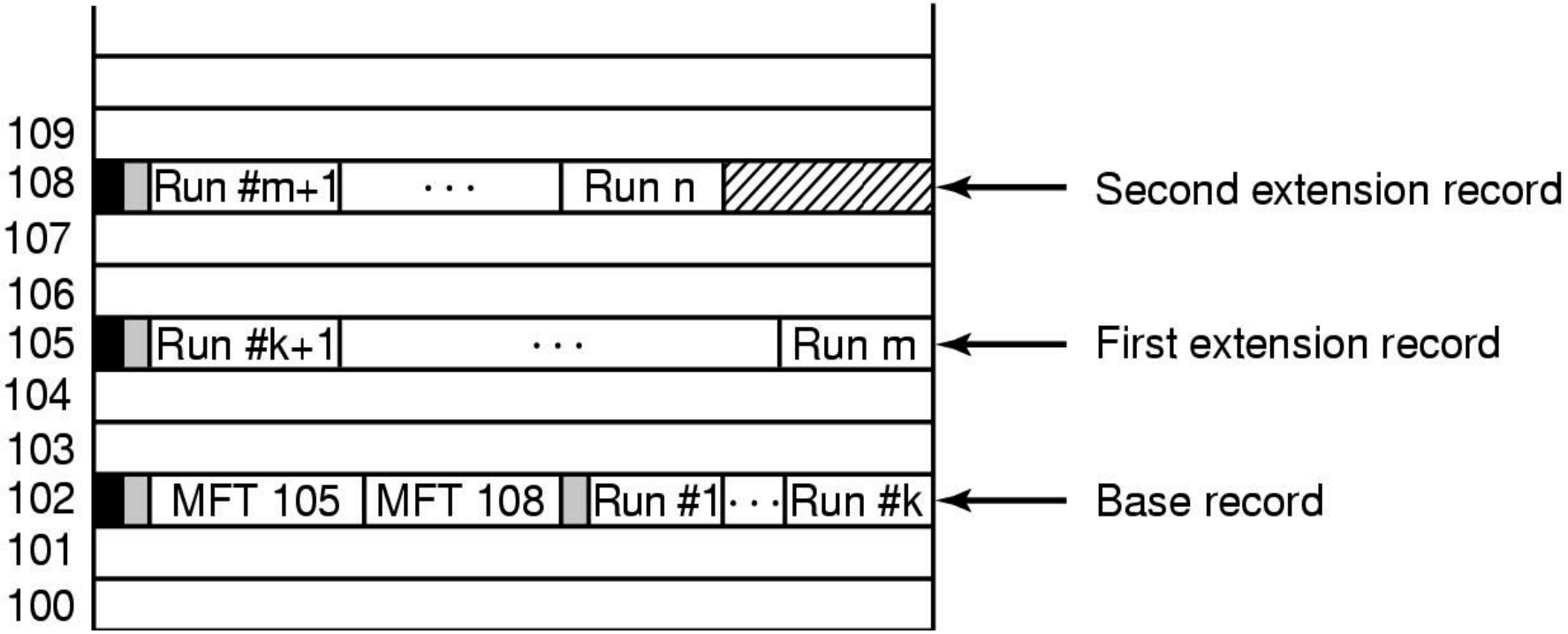
Attributi

- Un file può quindi avere più attributi di tipo data
- Se un file è particolarmente corto il suo contenuto è memorizzato nel MFT corrispondente
- Altrimenti nel MFT sono memorizzati gli indirizzi di sequenze di cluster (run) in cui sono presenti i dati

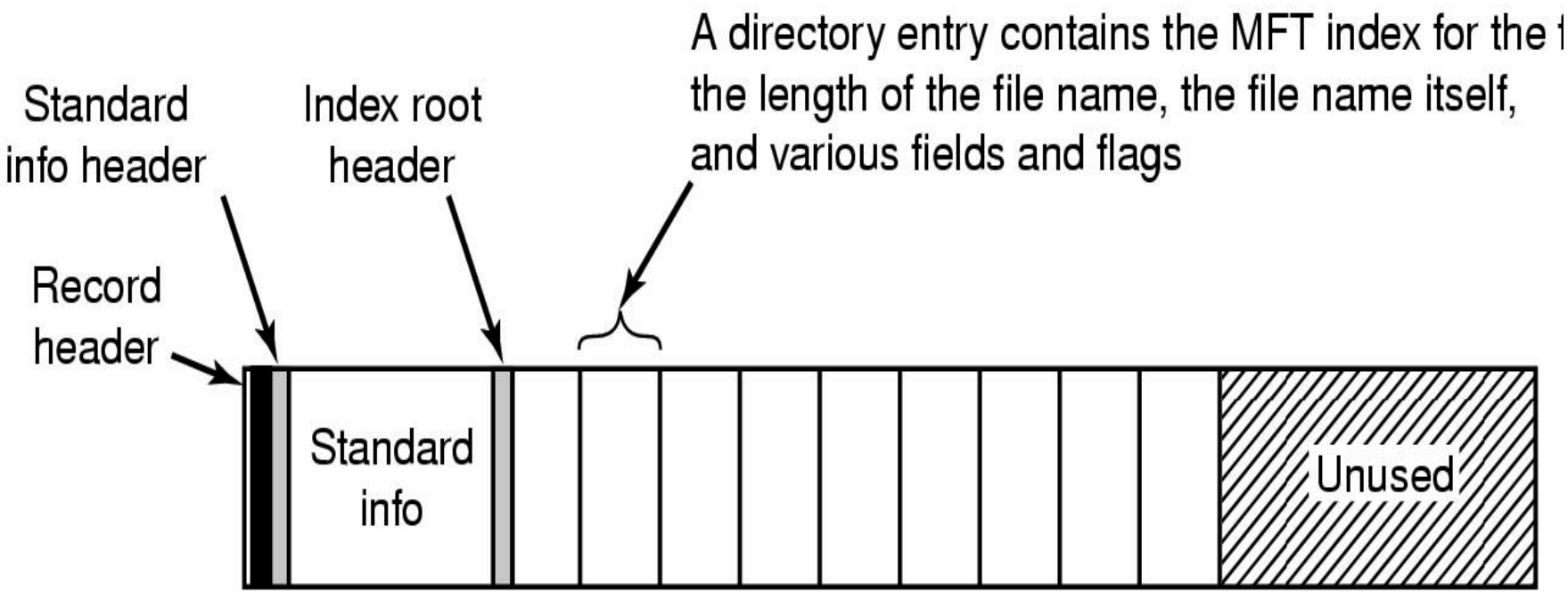
Esempio di Record MFT



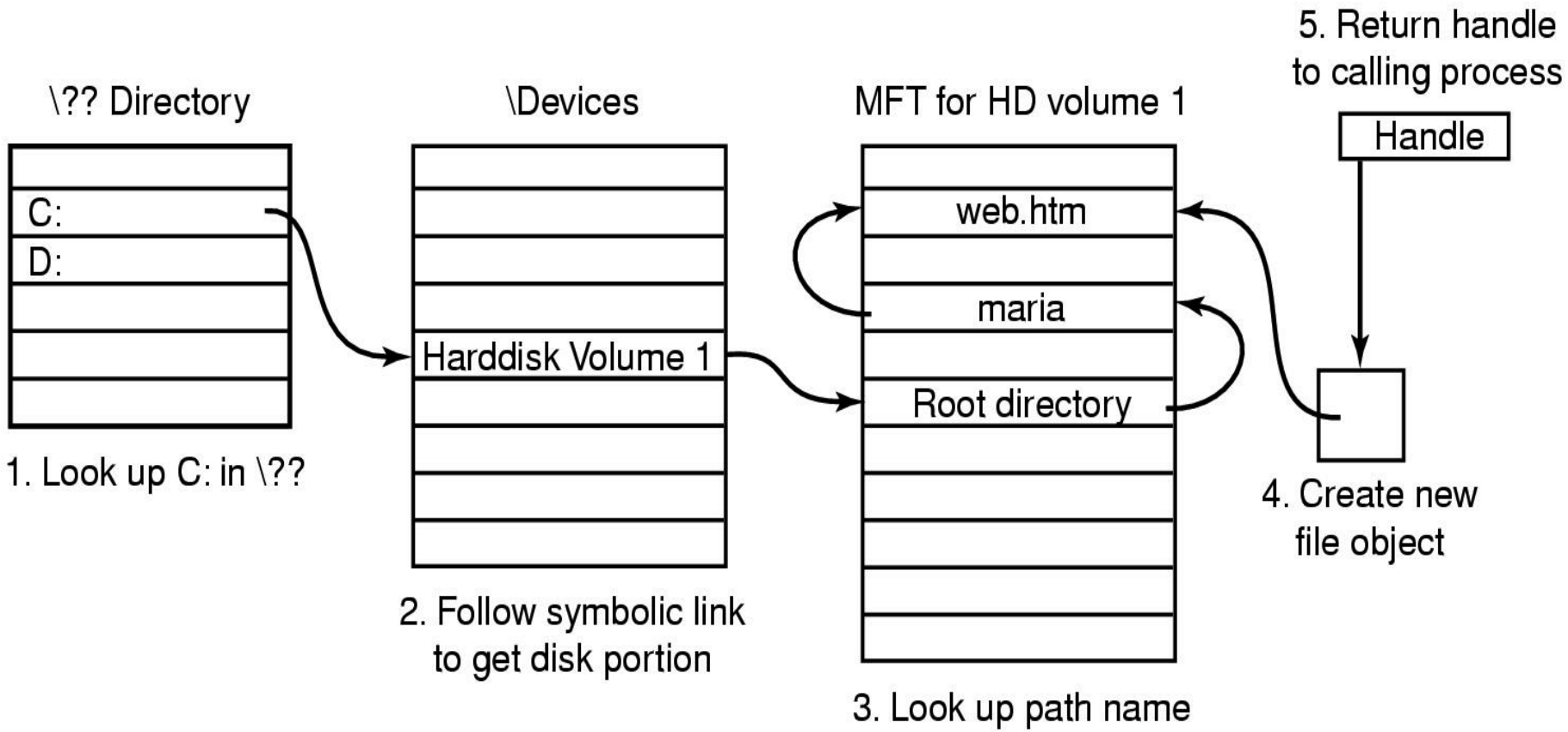
MFT complesso



Directory


A directory entry contains the MFT index for the f
the length of the file name, the file name itself,
and various fields and flags
Standard
info header
Index root
header
Record
header
Standard
info
Unused

File Name Lookup

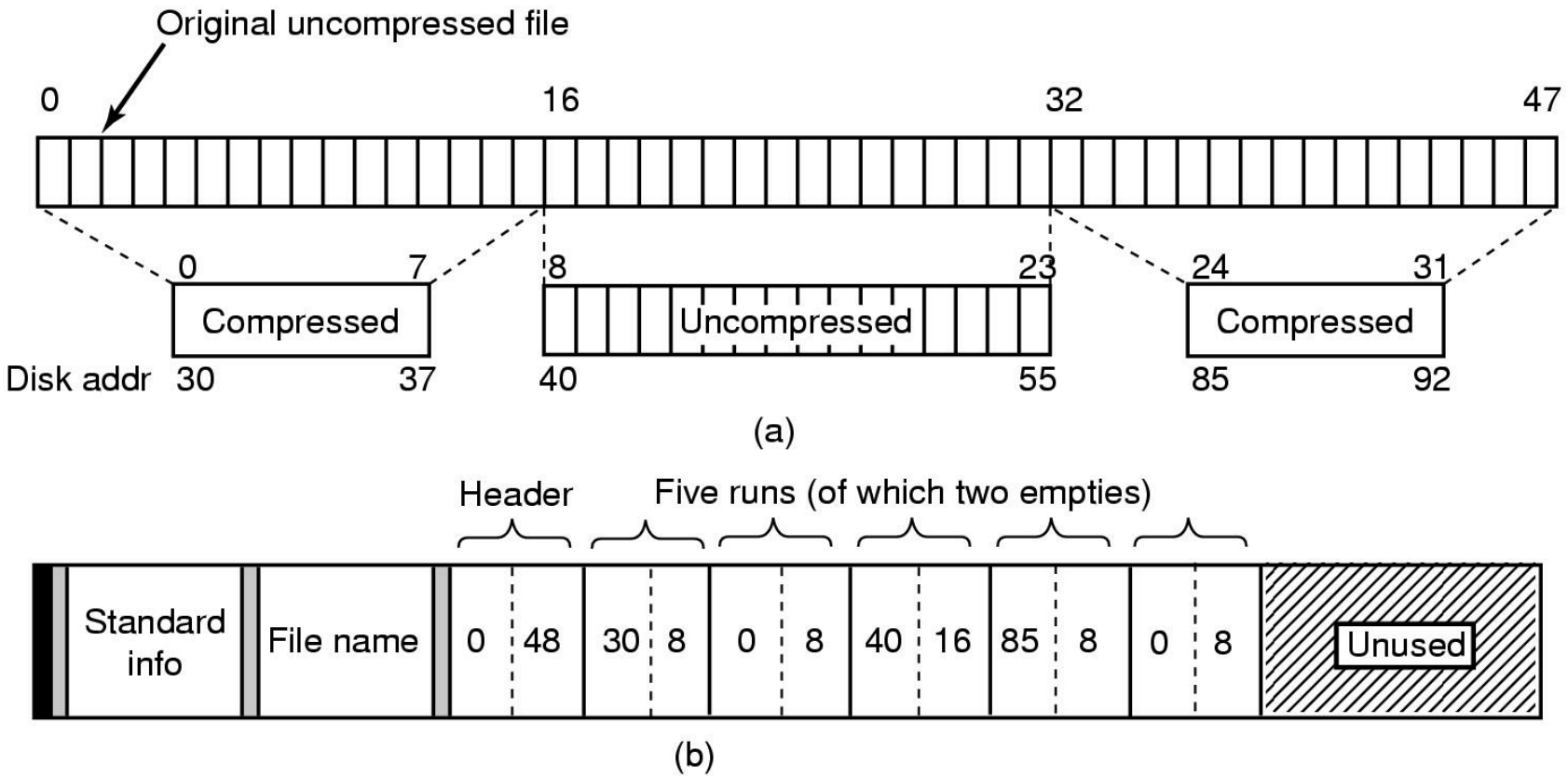


Passi per la ricerca del file *C:maria\web.htm*

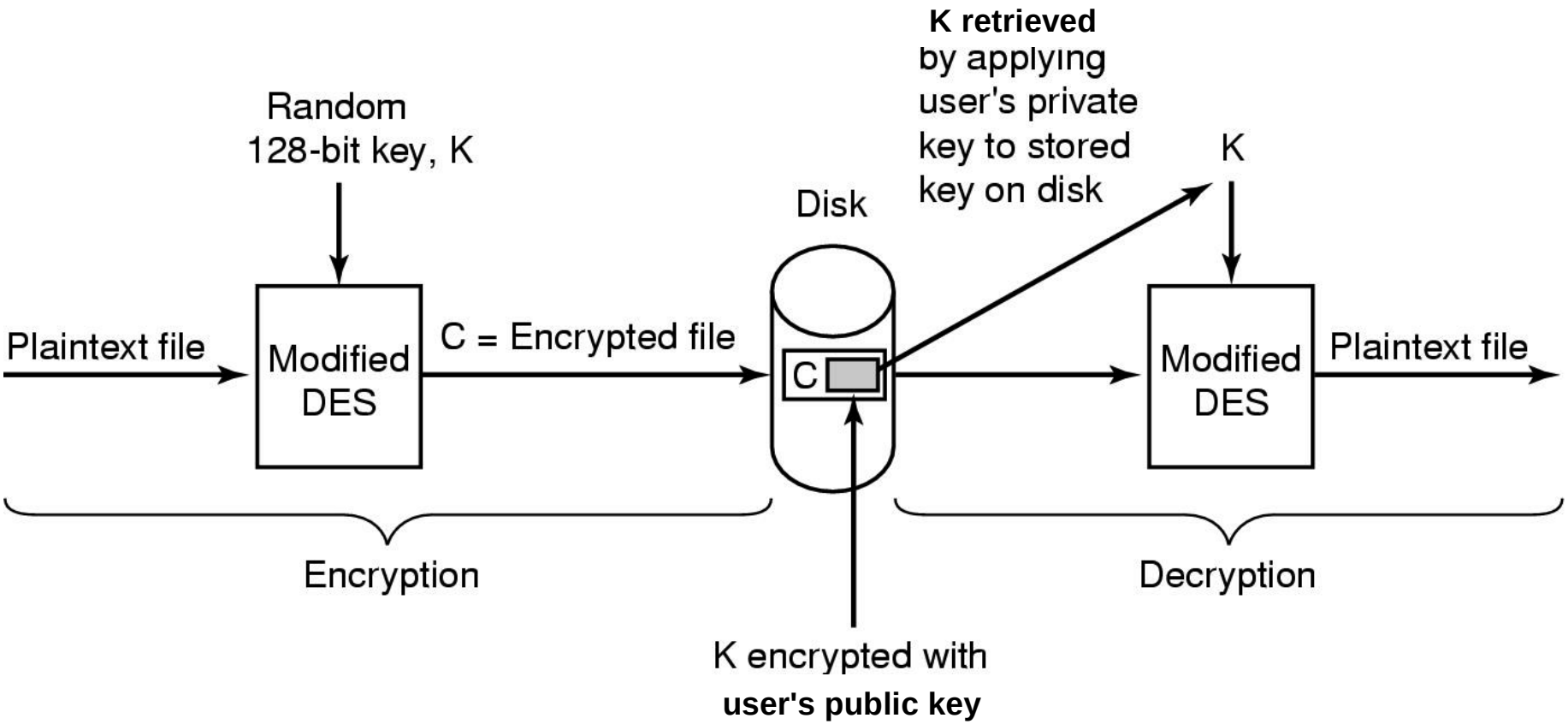
Compressione

- In W2000 un file può essere creato in modo compresso
- Ogni volta che un file viene riscritto su disco il sistema tenta di comprimerlo utilizzando come unità di riferimento 16 cluster
  - Se la compressione dà esito positivo si provvede a riscrivere i dati compressi

File Compression



File Encryption



Encrypted file system