Outline for Today's Lecture

Administrative:

– Happy Thanksgiving

Objective:

– NT File System

File System API Calls in Windows 2000 (1)

Win32 API function	UNIX	Description
CreateFile	open	Create a file or open an existing file; return a handle
DeleteFile	unlink	Destroy an existing file
CloseHandle	close	Close a file
ReadFile	read	Read data from a file
WriteFile	write	Write data to a file
SetFilePointer	lseek	Set the file pointer to a specific place in the file
GetFileAttributes	stat	Return the file properties
LockFile	fcntl	Lock a region of the file to provide mutual exclusion
UnlockFile	fcntl	Unlock a previously locked region of the file

- Principle Win32 API functions for file I/O
- Second column gives nearest UNIX equivalent

33

File System API Calls in Windows 2000 (2)

```
/* Open files for input and output. */
inhandle = CreateFile("data", GENERIC_READ, 0, NULL, OPEN_EXISTING, 0, NULL);
outhandle = CreateFile("newf", GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
    FILE_ATTRIBUTE_NORMAL, NULL);

/* Copy the file. */
do {
    s = ReadFile(inhandle, buffer, BUF_SIZE, &count, NULL);
    if (s && count > 0) WriteFile(outhandle, buffer, count, &ocnt, NULL);
} while (s > 0 && count > 0);

/* Close the files. */
CloseHandle(inhandle);
CloseHandle(outhandle);
```

A program fragment for copying a file using the Windows 2000 API functions

34

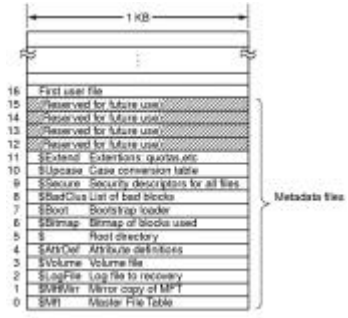
File System API Calls in Windows 2000 (3)

Win32 API function	UNIX	Description
CreateDirectory	mkdir	Create a new directory
RemoveDirectory	rmdir	Remove an empty directory
FindFirstFile	opendir	Initialize to start reading the entries in a directory
FindNextFile	readdir	Read the next directory entry
MoveFile	rename	Move a file from one directory to another
SetCurrentDirectory	chdir	Change the current working directory

- Principle Win32 API functions for directory management
- Second column gives nearest UNIX equivalent, when one exists

35

File System Structure (1)



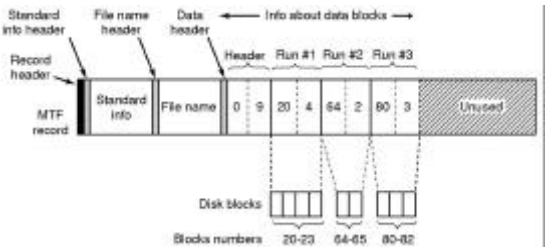
The NTFS master file table

File System Structure (2)

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

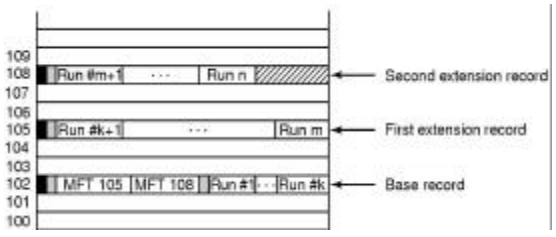
The attributes used in MFT records

File System Structure (3)



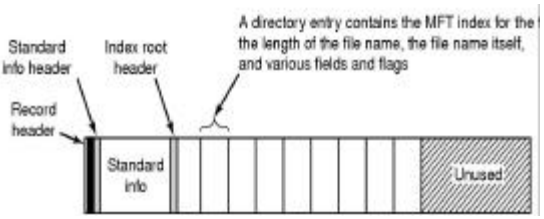
An MFT record for a three-run, nine-block file

File System Structure (4)



A file that requires three MFT records to store its runs

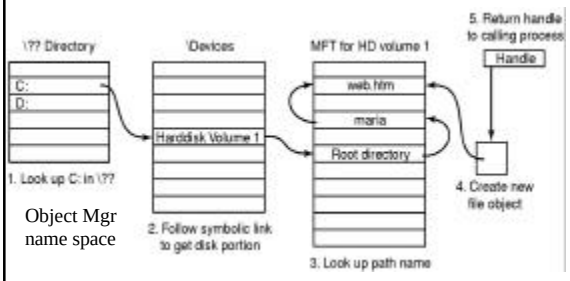
File System Structure (5)



The MFT record for a small directory.

40

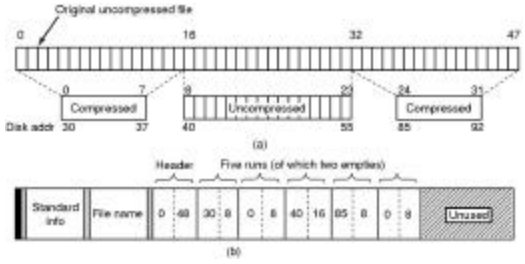
File Name Lookup



Steps in looking up the file *C:\maria\web.htm*

41

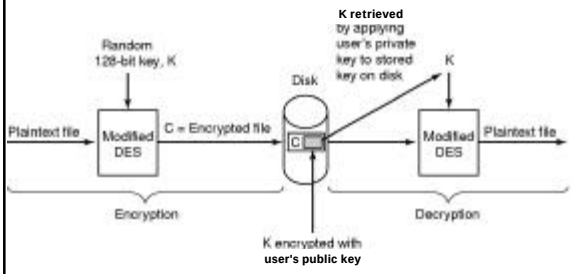
File Compression



(a) An example of a 48-block file being compressed to 32 blocks
(b) The MTF record for the file after compression

42

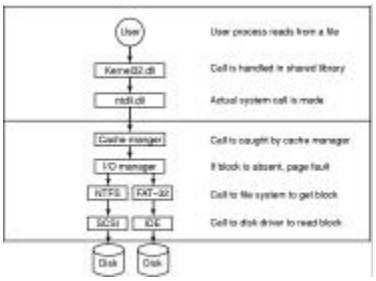
File Encryption



Operation of the encrypting file system

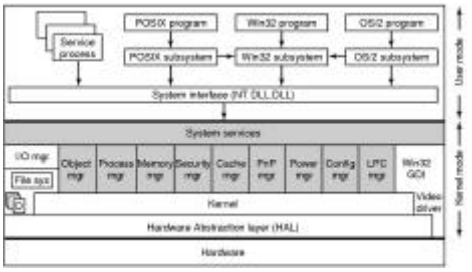
43

Caching in Windows 2000



The path through the cache to the hardware

The Operating System Structure



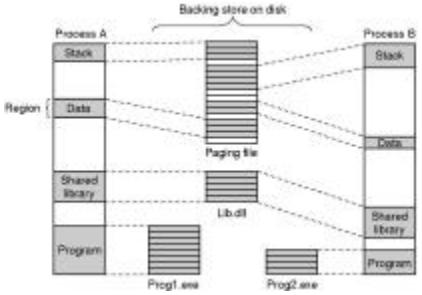
- Structure of Windows 2000 (slightly simplified).
- Boxes, D, are device drivers
- Service processes are system daemons

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

The principal Win32 API functions for mapping virtual memory in Windows 2000

Fundamental Concepts (2)



- Mapped regions with their shadow pages on disk
- The *lib.dll* file is mapped into two address spaces at same time

Comparisons

	FFS	LFS	NTFS
Data blocks	Clustering, Cylinder grouping	Log: contiguous by temporal ordering	Runs of contiguous blocks possible, immed.
Directories	Directory nodes, cylinder grouping	Directory nodes, in log	They *are* MFT entries
Block indices	Inodes, specified loc in cylinder group	Inodes in log	In MFT entries for files

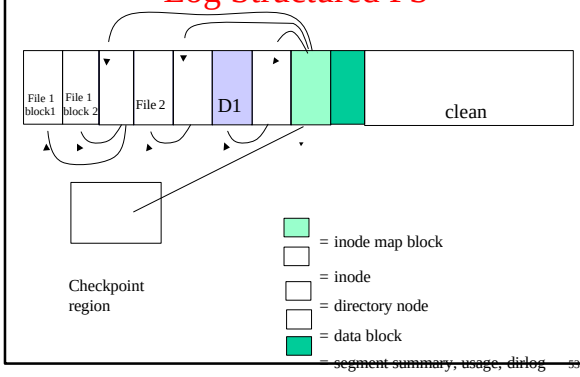
51

FFS Cylinder Groups

- FFS defines *cylinder groups* as the unit of disk locality, and it factors locality into allocation choices.
  - typical: thousands of cylinders, dozens of groups
  - Strategy: place "related" data blocks in the same cylinder group whenever possible.
    - seek latency is proportional to seek distance
  - Smear large files across groups:
    - Place a run of contiguous blocks in each group.
  - Reserve inode blocks in each cylinder group.
    - This allows inodes to be allocated close to their directory entries and close to their data blocks (for small files).

52

Log Structured FS



53