

OS: Windows XP

Name: Mahmoud Mohamed Kamal Ismail
Track: Open-Source Application Development
ID: 19

Contents	Page
1. File Systems (Windows XP)	2
2. Virtual Memory Management (Windows XP)	4
2.1. Introduction	4
2.2. Algorithms	5

1. File Systems (Windows XP)

In a computer, a file system (sometimes written filesystem) FS is the way in which files are named and where they are placed logically for storage and retrieval, FS allows users and programs to organize and sorts files on a computer, often through the use of directories (or "folders"). Access to data stored on disks is a central feature of all operating systems. Computers store data on disks using files, which are structured in specific ways in order to allow for faster access, higher reliability, and to make better use out of the drive's available space. The specific way in which files are stored on a disk is called a file system, it enables files to have names and attributes. It also allows them to be stored in a hierarchy of directories or folders arranged in a directory tree.

We can summarize the basic features of File System (FS) as:

- A file system is an abstraction to store, retrieve and update a set of files. The term also identifies the data structures specified by some of those abstractions, which are designed to organize multiple files as a single stream of bytes. responsible for organizing files and directories, and keeping track of which areas of the media belong to which file and which are not being used.
- The file system manages access to the data of the files, and manages the available space of the device(s) which contain it.
- File system is a system for organizing data in an efficient manner, directories and files, generally in terms of how it is implemented in the disk operating system, collection of files and directories stored on a given drive (floppy drive, hard drive, RAM drive, etc.).
- File systems allocate space, multiple physical units on the device.
- A file system can be thought of as an index or database containing the physical location of every piece of data on a hard drive. A file system is setup on a drive during a format.
- FS is the method for storing and retrieving files on a disk. It is system software that takes commands from the operating system to read and write the disk clusters (Groups of sectors).
- FS is a data processing application that manages individual files. It opens, closes, reads and writes the file as a single entity.
- As such, the file system is highly operating-system-dependent. In most cases you don't generally have a "choice" between different file system types. However, in some operating systems you do, and there can be a performance impact from the choice. Windows NT and 2000 typically give you your choice of file system; Windows 2000 supports FAT16, FAT32 and NTFS.
- Some file systems store files in packages as small as 512 bytes, while others store files in larger chunks called allocation units or clusters. Some are very simple file systems with few features and little overhead (such as the FAT file system used in DOS and Windows 9x), and others have many features but comparatively higher overhead (NTFS used in NT).

There are numerous file systems in use; for example, FAT32 and NTFS are Windows XP file systems.

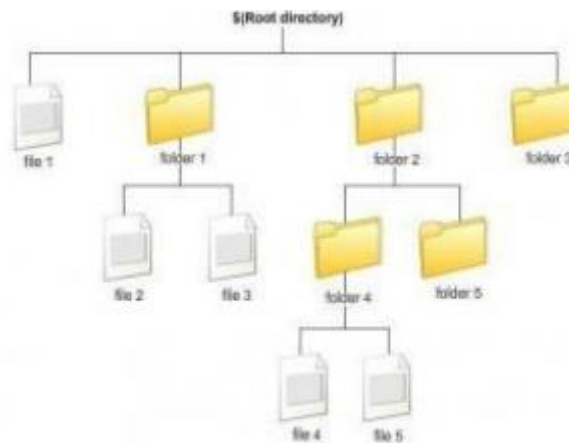


Figure 1.1: Examples of a file tree diagram.

File System Formulas (Types):

- **File Allocation Table 32 (FAT32):** FAT32 is the latest version of the FAT file system. It was introduced in 1996 for Windows 95. FAT32 addresses the limitations in FAT12 and FAT16, except for the file size limit of close to 4GB. It depends smaller clusters and larger storage units than those of FAT16, thus this produces more effective allocation of areas in storage units.
 - FAT32 has better, more and interactive recovery utilities (scandisk).
 - Scandisk is very quick.
 - Faster on drives less than 10GB.
 - Cluster size is 32KB.
- **New Technology File System (NTFS):** NTFS is a file system type that is commonly used for Microsoft Windows. It is the standard file system for Windows XP. It provides numerous improvements over the FAT file system, including better security and better disk utilization. NTFS is a proprietary file system developed by Microsoft Corporation for its Windows line of operating systems, beginning with Windows NT 3.1 and Windows 2000, including Windows XP, Windows Server 2003, and all their successors to date. NTFS offers several features that are not available with FAT 32: file compression, encryption, permissions, and mirroring drives. NTFS supersedes the FAT file system as the preferred file system for Microsoft's Windows operating systems. NTFS has several technical improvements over FAT and HPFS (High Performance File System), such as the use of advanced data structures to improve performance, reliability, and disk space utilization, plus additional extensions, such as security access control lists (ACL).

Notes:

- Use NTFS if You want to encrypt files, assign permissions to files.
- You will be formatting partitions larger than 32GB.
- You need to store individual files that are larger than 4GB.
- Provides better security and increased it on personal files with file encryption.

- Allows extended file names, foreign characters.
- Smaller file clusters, 4KB.
- The minimum supported volume size for NTFS is 10MB with a maximum of 2TB with no limit to file size.
- If you don't have a need for FAT32 and find a drive that is not NTFS, you should perform the following steps to convert your drive(s) to NTFS:
Click Start ---> Click All Programs ---> Click Accessories --->Click Command Prompt

2. Virtual Memory Management (Windows XP)

2.1. Introduction

- A virtual memory (VM) system can execute a process that is not completely resident in main memory.
- Processes can be larger than physical memory.
- Systems can use VM techniques to make process creation efficient, and to allow processes to share files, libraries, and memory.
- VM can very significantly degrade performance if not implemented with care, or not used with care.

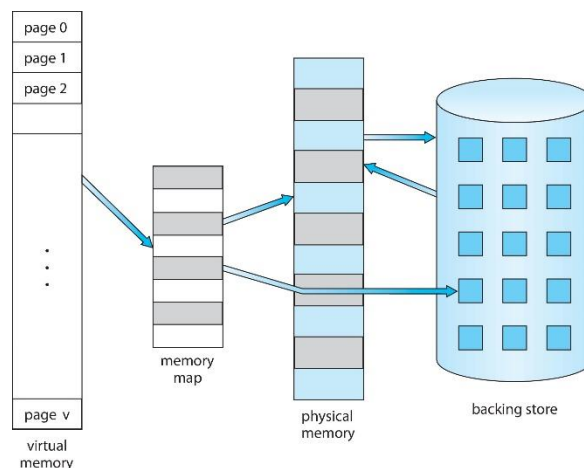


Figure 2.1: Diagram showing virtual memory that is larger than physical memory.

2.2. Algorithms

○ **Copy-on-Write:**

Windows XP uses copy-on-write with `fork()`. Instead of copying the entire parent process, the system copies only the pages on which the parent or child actually writes.

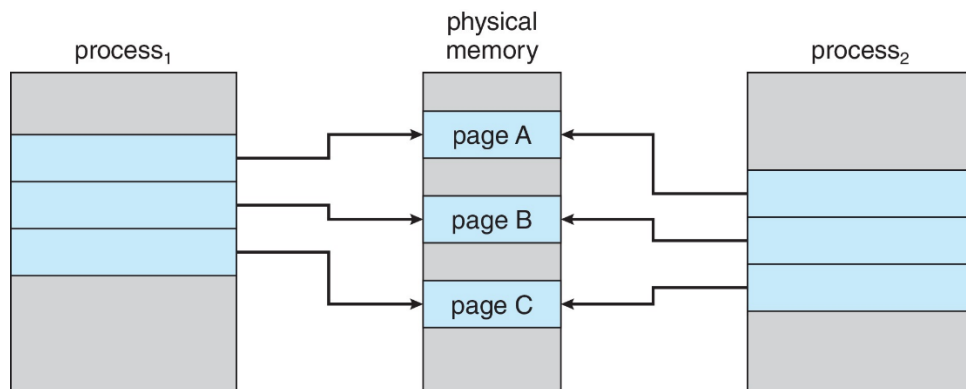


Figure 2.2: Before process 1 modifies page C.

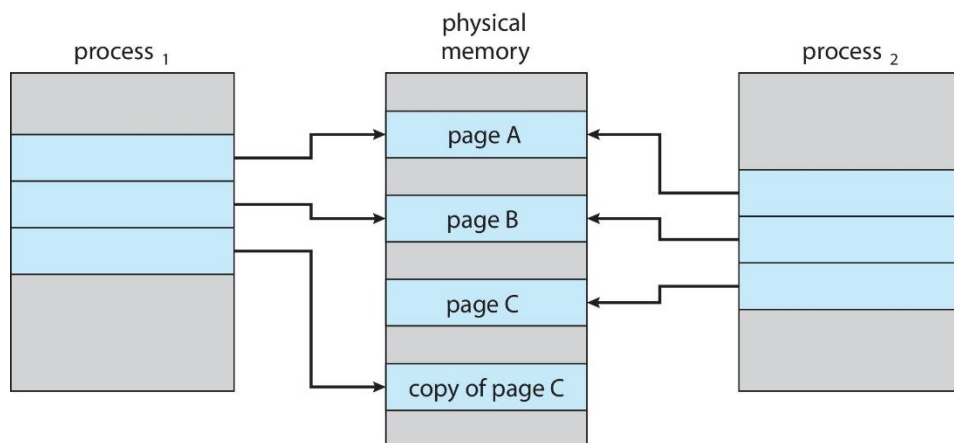


Figure 2.3: After process 1 modifies page C.

- Copy-on-write saves time if the child process does an `exec()` right after it is created.
- If the child process does not `exec()`, copy-on-write assures that only as much new memory as necessary will be allocated
- Some versions of unix utilize another variant of `fork()` called `vfork()`. With `vfork()`, the parent sleeps while the child uses the address space of the parent. The child is not supposed to write to the parent's space. The child is expected to soon use a form of `exec`, which gives the child a new address space of its own, separate and distinct from that of the parent. When the parent wakes it will see any changes the child made to the parent's address space, so caution is critical. This is considered dangerous and inelegant but efficient.

○ Page Replacement

Physical memory may become over-allocated -- there may be no free frames when a page-fault occurs! The standard solution is called page replacement.

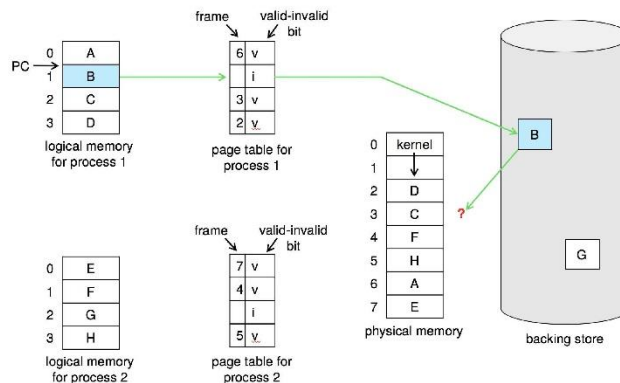


Figure 2.4: Need for page replacement.

Basic Page Replacement

If there is no free frame to service a page-fault then one way to handle the problem is to use a frame that is NOT free, but which does not appear to be needed too badly by the process that is currently using it.

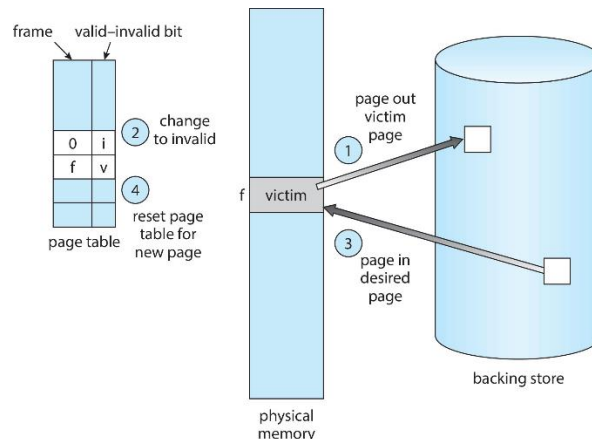


Figure 2.5: Page replacement.

- The OS writes the contents of the "victim" frame to swap space first, if it has been modified (as recorded in the "modify bit" (a.k.a. "dirty bit") associated with the frame). Otherwise, to save time, the OS **does not** write the frame. (If for some reason the page is missing from swap space and needs to be there, then write it anyway.)
- When performing page replacement, the OS must update data structures to reflect the change to the use of the frame. For example, it has to make changes to the frame table, and to the page tables of the processes that gain and lose the frame. (It's possible for the process that gains to be the same as the loser.)
- Each OS needs a solution to its frame-allocation problem: How many frames shall be allocated to each process?

- The OS also must have a page-replacement algorithm: When a page is to be replaced, which frame shall be the "victim"? We want a replacement algorithm that is best for keeping the page-fault rate low.
- We evaluate a page replacement algorithm by trying it out on some reference strings.
- Suppose you make a list L of all the logical memory addresses referenced by a process P during its lifetime, in the order the references are made. Suppose you make a list L' from L by just using the page numbers of each address from L. Suppose you then "consolidate" L' into a third list L" by "collapsing" all runs of the same page number into just one "copy" of that page number. The result would be the reference string of the process P.
- The reference string of a process is a sequence of transitions that the process makes. Each item in the reference string represents the process beginning to access a page that it was not accessing immediately beforehand. A page fault can happen only when a process makes such a transition.
- The length of a reference string is just the number of items in the sequence.
- We can evaluate a page-replacement algorithm by counting page faults on reference strings. We count the number of page faults a process gets when it uses an algorithm on a reference string.