OPERATING SYSTEM

ULEMU MPONELA
DMI ST JOHN THE BAPTIST UNIVERSITY : UNIT-IV

Table of Contents

1. **First In First Out (FIFO) algorithm** 4
2. Optimal Page algorithm 5
3. Least Recently Used (LRU) algorithm 5
4. Page Buffering algorithm 5
5. Least frequently Used(LFU) algorithm 5
6. Most frequently Used(MFU) algorithm 5

File Concept 6

a) **Sequential access** 7

b) **Direct/Random access** 7

c) **Indexed sequential access** 7

Types of Directory Structure 7

Unit IV Virtual Memory:

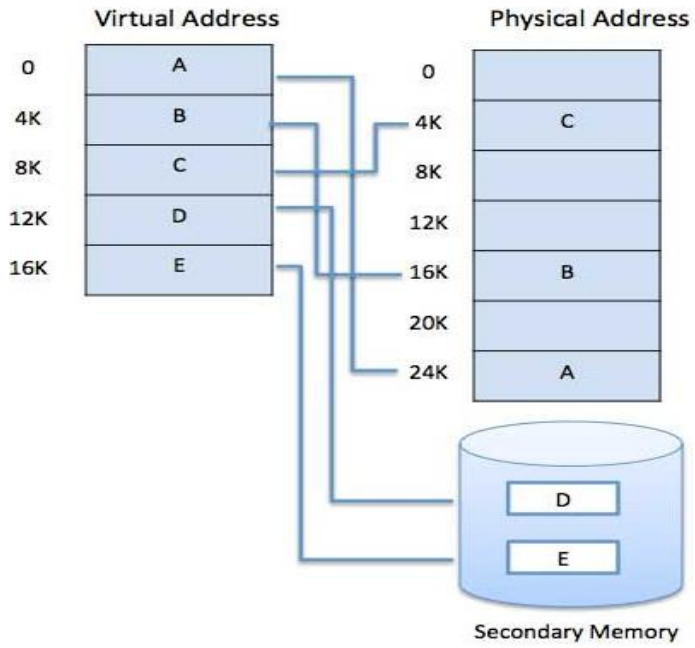
Demand Paging - Page Replacement - Page Replacement Algorithms - Thrashing. File System: Introduction - File Concepts - Access Methods - Directory Structures – Protection.

VIRTUAL MEMORY

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer's RAM.

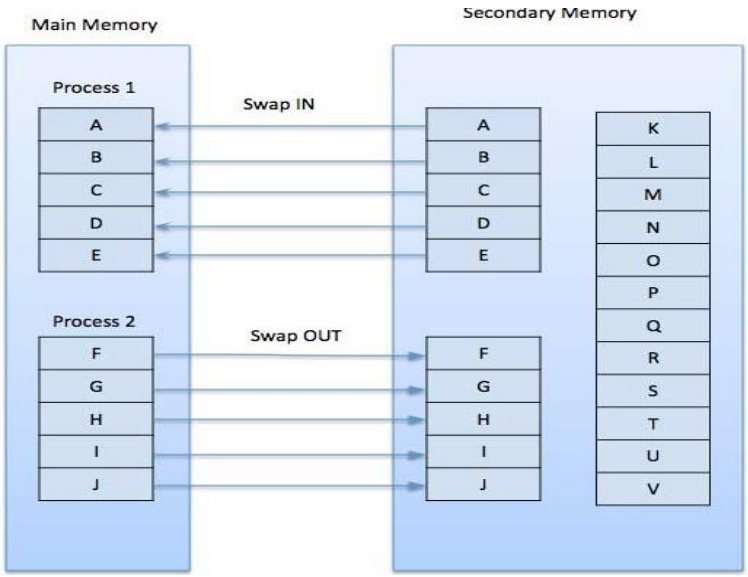
The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

Modern microprocessors intended for general-purpose use, a memory management unit, or MMU, is built into the hardware. The MMU's job is to translate virtual addresses into physical addresses. A basic example is given below



DEMAND PAGING

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program's pages out to the disk or any of the new program's pages into the main memory Instead, it just begins executing the new program after loading the first page and fetches that program's pages as they are referenced.



While executing a program, if the program references a page which is not available in the main memory because it was swapped out a little ago, the processor treats this invalid memory reference as a page fault and transfers control from the program to the operating system to demand the page back into the memory.

PAGE REPLACEMENT ALGORITHMS

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages.

When the page that was selected for replacement and was paged out, is referenced again, it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm.

A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page misses, while balancing it with the costs of primary storage and processor time of the algorithm itself. There are many different page replacement algorithms. We evaluate an algorithm by running it on a particular string of memory reference and computing the number of page faults

1. First In First Out (FIFO) algorithm

- Oldest page in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages from the tail and add new pages at the head.

2. Optimal Page algorithm

- An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. An optimal page-replacement algorithm exists, and has been called OPT or MIN.
- Replace the page that will not be used for the longest period of time. Use the time when a page is to be used.

3. Least Recently Used (LRU) algorithm

- Page which has not been used for the longest time in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages by looking back into time.

4. Page Buffering algorithm

- To get a process start quickly, keep a pool of free frames.
- On page fault, select a page to be replaced.
- Write the new page in the frame of free pool, mark the page table and restart the process.
- Now write the dirty page out of disk and place the frame holding replaced page in free pool.

5. Least frequently Used(LFU) algorithm

- The page with the smallest count is the one which will be selected for replacement.
- This algorithm suffers from the situation in which a page is used heavily during the initial phase of a process, but then is never used again.

6. Most frequently Used(MFU) algorithm

- This algorithm is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

THRASHING

In case, if the page fault and swapping happens very frequently at a higher rate, then the operating system has to spend more time swapping these pages. This state in the operating system is termed thrashing. Because of thrashing the CPU utilization is going to be reduced.

Let's understand by an example, if any process does not have the number of frames that it needs to support pages in active use then it will quickly page fault. And at this point,

the process must replace some pages. As all the pages of the process are actively in use, it must replace a page that will be needed again right away. Consequently, the process will quickly fault again, and again, and again, replacing pages that it must bring back in immediately. This high paging activity by a process is called thrashing.

During thrashing, the CPU spends less time on some actual productive work spend more time swapping.

FILE SYSTEMS

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user

To a user, the file is the smallest unit of storage on a computer system. The user performs file operations such as open, close, read, write and modify. They can create or delete files from the system

File Concept

Computer store information in storage media such as disk, tape drives, and optical disks. The operating system provides a logical view of the information stored in the disk. This logical storage unit is a file.

The information stored in files are non-volatile, means they are not lost during power failures. A file is named collection of related information that is stored on physical storage.

Data cannot be written to a computer unless it is written to a file. A file, in general, is a sequence of bits, bytes, lines, or records defined by its owner or creator. The file has a structure defined by its owner or creator and depends on the file type.

- Text file – It has a sequence of characters.
- Image file – It has visual information such as photographs, vectors art and so on.
- Source file – It has subroutines and function that are compiled later.
- Object file – It has a sequence of bytes, organized into bytes and used by the linker.
- Executable file – The binary code that the loader brings to memory for execution is stored in an exe file.

FILE ACCESS METHODS

File access mechanism refers to the manner in which the records of a file may be accessed.

When a file is used, information is read and accessed into computer memory and there are several ways to access this information of the file. Some systems provide only one access method for files.

There are several ways to access files:

- Sequential access
- Direct/Random access
- Indexed sequential access

a) Sequential access

A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other. This access method is the most primitive one. Example: Compilers usually access files in this fashion.

b) Direct/Random access

- Random access file organization provides, accessing the records directly.
- Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.
- The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

c) Indexed sequential access

- This mechanism is built up on base of sequential access.
- An index is created for each file which contains pointers to various blocks.
- Index is searched sequentially and its pointer is used to access the file directly.

DIRECTORY STRUCTURE

A Directory is the collection of the correlated files on the disk. In simple words, a directory is like a container which contains file and folder. In a directory, we can store the complete file attributes or some attributes of the file. A directory can be comprised of various files. With the help of the directory, we can maintain the information related to the files.

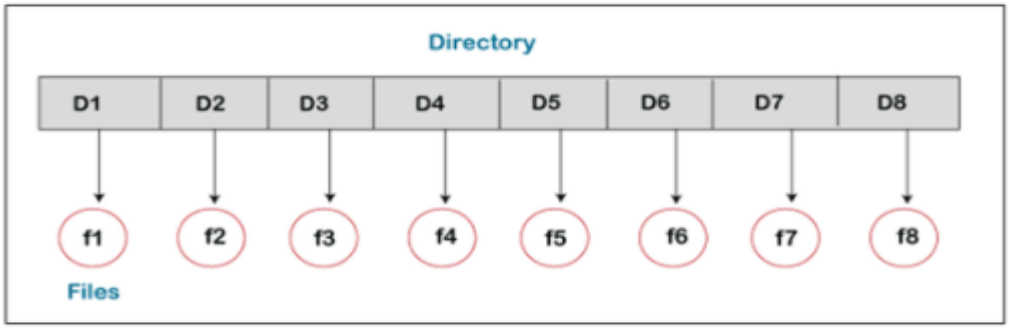
Types of Directory Structure

There are various types of directory structure:

- Single-Level Directory
- Two-Level Directory
- Tree-Structured Directory
- Acyclic Graph Directory
- General-Graph Directory

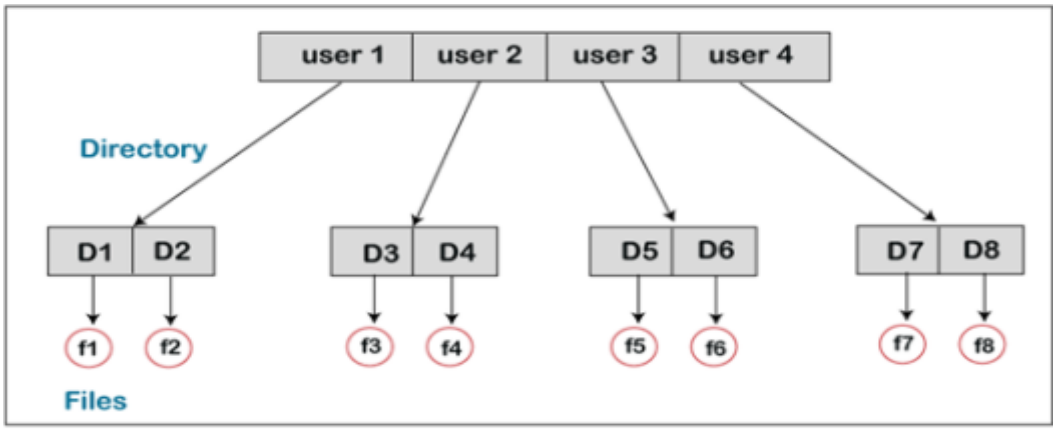
1) Single-Level Directory:

This is the easiest directory structure. There is only one directory in a single-level directory, and that directory is called a root directory. In a single-level directory, all the files are present in one directory that makes it easy to understand. In this, under the root directory, the user cannot create the subdirectories.



2) Two-Level Directory

Two-Level Directory is another type of directory structure. In this, it is possible to create an individual directory for each of the users. There is one master node in the two-level directory that include an individual directory for every user. At the second level of the directory, there is a different directory present for each of the users. Without permission, no user can enter into the other user's directory.



3) Tree-Structured Directory

A Tree-structured directory is another type of directory structure in which the directory entry may be a sub-directory or a file. The tree-structured directory reduces the limitations of the two-level directory. We can group the same type of files into one directory.

In a tree-structured directory, there is an own directory of each user, and any user is not allowed to enter into the directory of another user. Although the user can read the data of root, the user cannot modify or write it. The system administrator only has full access to the root directory. In this, searching is quite effective and we use the current working concept. We can access the file by using two kinds of paths, either absolute or relative.

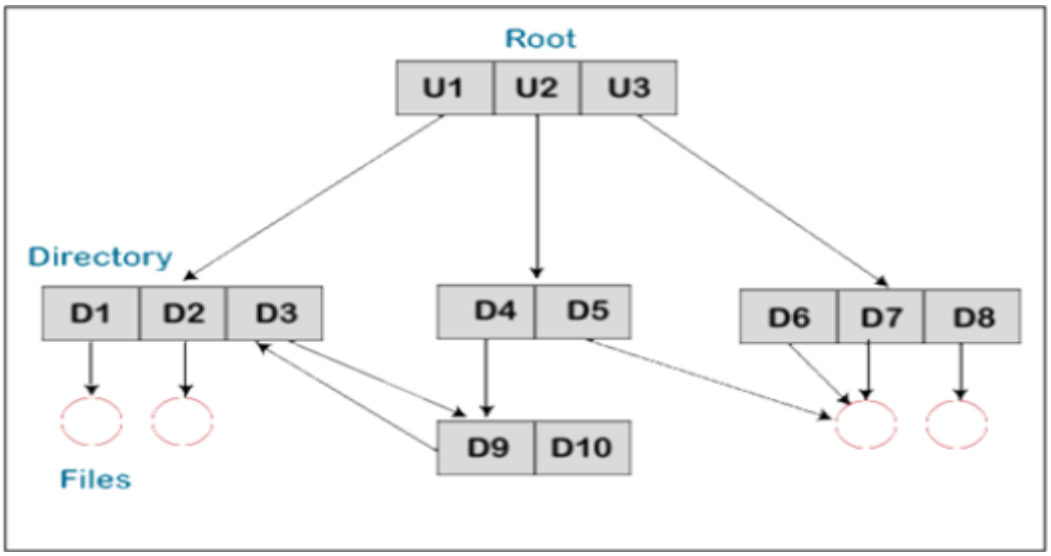
4) Acyclic-Graph Directory

In the tree-structure directory, the same files cannot exist in the multiple directories, so sharing the files is the main problem in the tree-structure directory. With the help of the acyclic-graph directory, we can provide the sharing of files. In the acyclic-graph directory, more than one directory can point to a similar file or subdirectory. We can share those files among the two directory entries.
With the help of aliases, and links, we can create this type of directory graph. We may also have a different path for the same file. Links may be of two kinds, which are hard link (physical) and symbolic (logical).
If we delete the files in acyclic graph structures, then

- In the hard link (physical) case, we can remove the actual files only if all the references to the file are deleted.
- In the symbolic link (logical) case, we just delete the file, and there is only a dangling point that is left.

5) General-Graph Directory

The General-Graph directory is another vital type of directory structure. In this type of directory, within a directory we can create cycle of the directory where we can derive the various directory with the help of more than one parent directory.
The main issue in the general-graph directory is to calculate the total space or size, taken by the directories and the files.



PROTECTING FILES

The files which have direct access of the any user have the need of protection. The files which are not accessible to other users doesn't require any kind of protection. The mechanism of the protection provides the facility of the controlled access by just limiting the types of access to the file. Access can be given or not given to any user depends on several factors, one of which is the type of access required. Several different types of operations can be controlled:

- **Read**
  Reading from a file.
- **Write**
  Writing or rewriting the file.
- **Execute**
  Loading the file and after loading the execution process starts.
- **Append**
  Writing the new information to the already existing file, editing must be end at the end of the existing file.
- **Delete**
  Deleting the file which is of no use and using its space for the another data.
- **List**
  List the name and attributes of the file.

Operations like renaming, editing the existing file, copying; these can also be controlled. There are many protection mechanism, each of them mechanism have different advantages and disadvantages and must be appropriate for the intended application.

Access Control:

There are different methods used by different users to access any file. The general way of protection is to associate identity-dependent access with all the files and directories and list called access-control list (ACL) which specify the names of the users and the types of access associate with each of the user. The main problem with the access list is their length. If we want to allow everyone to read a file, we must list all the users with the read access. This technique has two undesirable consequences:
Constructing such a list may be tedious and unrewarding task, especially if we do not know in advance the list of the users in the system.

Previously, the entry of the any directory is of the fixed size but now it changes to the variable size which results in the complicates space management. These problems can be resolved by use of a condensed version of the access list. To condense the length of the access-control list, many systems recognize three classification of users in connection with each file:

- **Owner**
  Owner is the user who has created the file.
- **Group**
  A group is a set of members who has similar needs and they are sharing the same file.
- **Universe**
  In the system, all other users are under the category called universe.

The most common recent approach is to combine access-control lists with the normal general owner, group, and universe access control scheme. For example: Solaris uses the three categories of access by default but allows access-control lists to be added to specific files and directories when more fine-grained access control is desired.

Other Protection Approaches:

The access to any system is also controlled by the password. If the use of password is random and it is changed often, this may be result in limit the effective access to a file.