OPERATING SYSTEM

Lecture 10

File Management

Dr Victoria Mukami

INTRODUCTION

During this lecture, we will focus on file management. More specifically, we will review the role of the file manager, the difference between files and file systems, and the different types of physical storage

Learning objectives

By the end of this topic, you should be able to:

1. Describe concepts around files and file systems
2. Understand the role of the file manager.
3. Describe the different types of physical storage

OVERVIEW

So far, we have reviewed the input and output management from an operating system's point of view. We now move to file management. Most of the work that users do with a computer or with an operating system is working on files and storing the same files. We start by reviewing the concepts around files and the file system.

FILES AND FILE SYSTEM

A file can be defined as a collection of related data, programs, and information that is stored within a computer. A file system on the other hand can be defined as one that allows users to create data collections [2]. File systems also have several desirable properties

Long-term existence [1]: Files are stored permanently until when the user deletes them. That is the system does not clear files when the user stops using the machine.

Sharable between processes [1]: Files are named appropriately and can include access permissions that allows for sharing.

Structure [1]: A file can have an internal structure that is convenient for a specific application. File organization is also key for any operating system.

Functions of a file system

A file system has several functions that can be performed on files. Some of these functions are discussed below:

Create: A new file is created and placed within the existing structure of files

Delete: A file is removed from the file system and destroyed

Open: An already created file is opened and edited or worked on.

Close: An open file is closed when an application or process no longer needs it.

Read: This is when a process reads a section or all the contents of a file

Write: When a process updates a file and makes changes to the file.

FILE MANAGER

The file manager is responsible for creating, editing, and controlling access to files [1]. It also manages the resources used by those files. The file manager needs to perform four tasks to carry out its responsibilities [1]

1. Keep track of where each file is stored [1]
2. Use policies that determine where and how each file will be stored. This is done by ensuring that the available space is used efficiently.
3. Allocate each file to a user who has rights to the file and track the user's use of the file
4. Deallocate a file when it is ready to be returned to storage. Additionally, indicate its availability if it is needed by others.

The file manager keeps track of "its file with directories that contain the filename, its physical location in secondary storage and information about it "[1].

Within the cloud, the file manager does not have a specific job as all the files are managed locally on devices. Remember on the cloud, the files are accessible remotely but stored within specific devices. Before we continue, let us review some definitions:

Field: This is the basic element of data [2]. A field represents a single piece of data. It is identified by a field header.

Record: This is a group of related fields. One record represents several related fields.

File: When several records are placed together, it becomes a file. In other words, a file is a group of related records.

Database: This is a group of related files. Within the database, several relationships exist between files making the structure complex.

Programs files contain instructions and data files contain data and the file manager treats them the same way [1]. Directories, on the other hand, are special files that contain lists of filenames and their attributes [1].

Files in a computer system reside within secondary storage. Each storage unit is considered a volume with each volume having several files [1]. Each volume gets a name and this name and other details concerning the volume are written by the file manager to a place that is accessible. Once the volume descriptor (the common area with the volume information) is created a master file directory (MDF) is stored and lists the names and characteristics of every file within the volume [1]. The names within the MDF refer to program files, data files, and system files [1].

A subdirectory is created when a user opens an account to access the computer system [1]. This entry within MDF appears as a subdirectory even though the user's directory is treated like a file. Files are named depending on the application. File names are either absolute or relative.

Relative filename

A relative file name includes only the name given by the user. If a user created a file called Document_One, that would be the file name.

Absolute filename

An absolute filename includes all information including where the path is found. With the example before where our relative file name was Document_One, the absolute file name on a Windows machine would be C:\Users\Admin\Document_One.

Different operating systems have different ways in which file names can be done. The following table is a summary of different operating systems and the requirements for files [1].

Table 1: File Names Summary (Adapted) [1]

Operating System Case Sensitive	Special Characters	Not Allowed	Extension Required	Maximum Character Length
Android	Yes	* ? $ & [ ] / \	No	126/255*
Linux	Yes	* ? $ & [ ] / \	No	256
Mac OS X	Yes	Colon	No	255

MS-DOS	No	Only hyphens and underlines allowed	Yes	8.3
UNIX	Yes	* ? $ & [ ] / \	No	256
Windows	No	Most special characters are not allowed	Yes	255/256

FILE ORGANIZATION

When we talk about file organization, we mainly look at the arrangement of records within a file. Files are composed of records [1]. The format for any record is either fixed length or variable length.

Fixed-length records: these are the easiest to access directly [1]. They are stored in a fixed size, therefore, any part of the records that is larger than the fixed size is truncated [1]. If the record is smaller than the fixed length, then the extra storage space will be wasted.

Variable-length records: these do not leave any empty storage and do not cause any truncation. They are not easily accessible directly because it is not easy to locate where all the files are stored [1]. When accessed sequentially they are easy to locate.

PHYSICAL FILE ORGANIZATION

This concerns the way records are arranged and the characteristics used to store them. There are several file organization methods.

Sequential record organization: this is the easiest to implement since variable-length records are retrieved one after the other. To access the files, the file manager just needs to search from the beginning until it finds it.

Direct file organization: this uses direct access files. This gives users the flexibility to access any record in any order. Unlike the sequential record organization that requires a sequence to retrieve files, this is known as a random organization [1].

Indexed sequential record organization: this takes the good of sequential and direct access. It uses the indexed sequential access method application that handles overflows while maintaining the record order [1].

PHYSICAL STORAGE ALLOCATION

Remember that records are subdivided into fields. For instance, a record may contain several fields. Most records are managed by application programs apart from those that are oriented toward database applications are managed by the file manager [1]. Physical storage is either contiguous, non-contiguous, or indexed.

Contiguous Storage

Contiguous means one after the other. In this regard, contiguous storage refers to records being stored one after the other [1]. This is like a sequence which is very easy to implement. Think of how music was stored on a CD or DVD. One after the other. The primary advantage of this is the ease of implementation. In addition, there is the ease of direct access due to each part of a file being stored in the same compact area [1].

The biggest drawback is the lack of flexibility. A file cannot expand as it is not possible unless there is a space next to the file. If the file needs to expand, then it needs to be moved to a larger section every time it is updated [1]. A second drawback is a fragmentation. Fragmentation occurs because space is reserved for files and this space may be unused especially if the file is smaller. This space needs to be compacted to minimize wasting space. During fragmentation, files are inaccessible.

Non-Contiguous Storage

This is like random file storage. Files are stored within any available space within the disk. Any additional sections to a file are stored within other free sections of the hard disk. These additional sections are known as extents [1]. Linking of the extents happens by having a directory entry that consists of the file name, the storage location of the first and last extent and the total number of extents [1]. The system can locate the middle extents by following the pointers of the first extent [1]. Each extent then holds a pointer to the next extent. A different method involves all the file extents at the directory level. The drawback to this method is that it does not allow direct access. This method eliminates the need for defragmentation.

Indexed Storage

This is a method that allows for direct record access by bringing together all the links of the extent into an index block [1]. Each file has its index block that consists of the addresses of each disk sector that make up the file. When a file has been created the pointers to each index block are set to null [1]. As each sector is filled the pointer is set to the appropriate address. Indexed storage supports both sequential and direct access. It does not, however, improve storage space use due to the index block.

SUMMARY

During this lecture, we focused on file management. More specifically, we have reviewed the role of the file manager, the difference between files and file systems, and the different types of physical storage.

DISCUSSION TOPIC

Windows 11 is one of the newer operating systems by Microsoft. Find out the type of physical storage that Windows 11 supports. What are the drawbacks and benefits of the same?

REFERENCES

[1] McHoes, A., & Flynn, I., Understanding Operating Systems. Boston: Cengage Learning, 2018

[2] Stallings, W., Operating Systems: Internals and Design Principles. Harlow: Pearson Education Limited, 2018.