THE OPEN UNIVERSITY OF KENYA

DESIGN PLAN

Programme title	Bachelor of Science in Cybersecurity and Digital Forensics
Course title	Operating Systems
Learning Module number	08
Learning module title	File Systems
Module Developer	Elisha Abade
Module duration in hours	8
Instructional Hour Equivalent (Divide duration by 2)	4
Reviewed by	
Vision	The innovative university for inclusive prosperity
Audience description	Learners of Cyber Security in first semester of second year
Instructions to learners ⓘ	In this course we shall be learning about Filesystems in Operating Systems. We'll begin by watching videos on Operating systems. You are encouraged to ensure that you have access to a reliable Internet and that your devices (computer, tablet or phone) have properly working multimedia systems. This module also presents a number of interactive and non-interactive activities. You will be required to complete all the activities.
Learning module description	This module aims to facilitate learners to have an understanding of the fundamental concepts of filesystems. We'll look at the functions, structure and organization as well as performance and optimization of filesystems.
Module objectives:	This module aims at facilitating learners to acquire knowledge about: 1. Filesystems and their functions; 2. Filesystem structure and organization; 3. Filesystem Management and Optimization 4. Filesystem performance
Module learning outcomes:	By the end of the module, you should be able to: 1. Explain a filesystems and its functions 2. Describe the structure and organization of filesystems 3. Select filesystem management and optimization approaches 4. Apply various metrics to show filesystem performance
Planned Learning Resources	

ACTIVITY 1: INTRODUCTION VIDEO 1: Pre-recorded lecture on topic emphasizing LEARNING OUTCOME 1: Factual knowledge. 	**Video 1: File systems and their functions (8 minutes)** Welcome to the eighth module of this course in Operating Systems. In this module you will be learning about definition and function of file systems, their structure and the security mechanisms that they provide so that they can protect their files. Computer systems need to store several pieces of information which are at times more than the virtual address space. The information must survive the termination of the process using it and at times, multiple processes will need to access a single information concurrently. In modern computers, this information is stored in disks which are organized in blocks that can be read and written by computer systems. In order to achieve storage of information, computer systems use the concept of file systems. A file system is an abstraction provided by the Operating System to deal with accessing the information kept in blocks on a disk. It is important to note that files are created by a process and there are thousands of them, all managed by the Operating System. The Operating system achieves its management function through implementation of a file system that it uses to: structure, name and protect the files and directories. A file system can be viewed from two perspectives: 1. User view: Deals with how files are named, protected and organized. 2. Implementation view: Deals with how files are organized on a disk. The user point of view 1. Naming 2. Structure 3. Directories **Naming** Operating Systems establish rules for naming of files and these vary from one Operating System to the other. The rules specify the types and number of characters that can be used. **File Structure** This defines the byte sequences, and the maximum length of records that can be stored. It also specifies the type of files that can be supported by an Operating System. **Types of files** There are different types of files that can be supported by an Operating System. These include: ● Regular files: They contain user information

- Directories: Usually contain other files within them
- Character special files: These serve specific tasks such as modeling serial (e.g. printers) I/O devices
- Block special files: These are also special files that model disks

Regular Files:
These include ASCII or binary files. They can use pipes to connect programs if they produce/consume ASCII.

System Calls for files
These are utility functions that are provided by the Operating System for management and manipulation of files. Some of them include:

- create: Creates file, typically with no data and sets some attributes
- delete: Free disk space
- open: Typically called after *create*, gets attributes and disk addresses into main memory.
- close: Frees table space that are used by attributes and addresses
- read: reads data from the current pointer position. You need to specify buffer into which data is placed
- Write: Usually begins writing into a file from the current position
- append: Puts a character or string at the end of the file
- seek: Puts the file pointer at a specific place in the file. The read or write activities begin from that position

Directories
These are files which are used to organize a collection of other files. In MS windows, they are commonly known as folders. They can also be organized into two structures, namely, single-level and hierarchical directory systems.

1. Single Level Directory Systems

Insert figure of a single level directory containing four files here

2. Hierarchical Directory Systems

Insert a figure showing hierarchical directories here

Directory Operations
Some of the operations for manipulation of directories are as indicated below:

- create: creates directory
- delete: deletes a directory but the directory has to be empty in order for you to delete it
- opendir: Opens a directory. It must be done before any operations on directory
- closedir: closes a directory

- readdir: returns the next entry in open directory
- rename: changes the name of a directory
- link: links file to another directory
- unlink: Gets rid of directory entry

Video 2: File system structure and organization

Files are typically stored in disks (stable storage) of computer systems. The disks where files are stored are typically broken up into one or more partitions. The first sector in a disk, (sector 0) is used to boot the computer and commonly referred to as the Master Boot Record (MBR). At the end of MBR is the partition table that has starting and ending addresses of each partition. One of the partitions is marked active in the master boot table

Typically when a computer boots, the BIOS reads and executes the MBR. The MBR then finds the active partition and reads in the first block (boot block). Program in the boot block locates the Operating System for that partition and reads it in.

A Possible File System Layout

Insert a diagram that illustrates a possible layout of the file system here.

File System Layout

In the layout indicated, a superblock contains information about the filesystem such as its type and number of blocks while the i-node contains information about files.

Allocating Blocks to files

There are at least four approaches that can be adopted by Operating Systems in allocation blocks to files. These include:

1. Contiguous allocation
2. Linked list allocation
3. Linked list using table
4. I-nodes

Contiguous allocation

In this approach, consecutive disk blocks are allocated to a file. This can be as illustrated in the figure below.

Designer: Insert a diagram that shows consecutive blocks of disk for a number of files e..g upto 5 files.

Advantage of contiguous allocation

1. Easy to implement
2. Read performance is great since only one seek is needed to locate the first block in the file.

Disadvantages of contiguous allocation

1. The bad-disk becomes fragmented over time

Linked list allocation

This approach involves storing a file as a linked list of disk blocks as illustrated in the diagram below.

Designer: insert diagram of a linked list with nodes organized such that the data area stores a block of a file and the "pointer" links to the next node until all blocks of a file are stored.

Advantages

1. Gets rid of fragmentation

Disadvantages

1. Random access is slow. Need to chase pointers to get to a block

Linked list using table

This approach involves using linked lists but puts pointers in a table in memory. The table is referred to as the "File Allocation Table (FAT)" and this is the approach commonly used by earlier versions of Windows Operating System.

Designer: insert diagram of a linked list with File Allocation Table in main memory.

I-nodes

The i-node keeps data structure in memory only for active files. The data structure lists disk addresses of the blocks and attributes of the files.

Log Structured File System

It is important to note that with advances in computing, we continue to get faster CPUs, bigger disks and memories bigger but disk seek time has not decreased. A number of technologies such as caching have been proposed to reduce the seek time.

Caching aims at optimizing writes because the disk needs to be updated frequently. This has led to structuring the disk as a log. The CPU "log-collects" writes and periodically sends them to a segment in the disk . This way, the writes tend to be very small.

A segment has a summary of contents (i-nodes, directories….). The Operating System keeps the i-node map on disk and caches it in memory to locate i-nodes.

A cleaner thread compacts the logs then scans segments for current i-nodes as it discards the ones that are not in use and sends the

current ones to memory. A writer thread then writes the current i-nodes out into new segments.

Journaling File Systems

It is necessary to guard against loss of files in the event of a disk crash. Typically when a file is removed from the disk, the following series of activities are undertaken:

- The file is removed from its directory.
- The i-node is then released to the pool of free i-nodes.
- All disk blocks are returned to the pool of free disk blocks

Based on the above series of activities, if a crash occurs then the file will potentially be messed up. In order to avoid this and ensure that the file system remains consistent, it is important to keep a journal (a list) of actions before you take them. That is, you write a journal to the disk then perform the required actions. This can then help to recover from a crash.

In order for this to succeed, there is a need to make the operations idempotent and also put in place data structures to help in achieving the idempotency. The Windows NTFS and Linux filesystems use journaling.

Video 3: File System Management and Optimization
Welcome to the third session of this module. In this session, you will learn about the different activities that the Operating System does as part of management and optimization of the filesystem.

To ensure proper operation and better performance of the filesystem, the Operating System undertakes a number of activities including:

1. Disk space management
2. File System Backups
3. File System Consistency
4. File System performance tracking

Disk Space Management
One key role in file systems management and optimization is the management of the disk space. There are different strategies that can be applied by the filesystem to ensure that the disk space is used well.

	One strategy is to use fixed size blocks which don't have to be adjacent because if files are stored as consecutive bytes and they grow then the file will have to be moved. While this is a good strategy, the challenge comes in how to determine an optimum block size. Bigger block sizes result in better space utilization, but worse transfer utilization. This brings a situation where the filesystem has to make a trade-off between space utilization and the data rate. Another strategy is to enforce disk quotas. This way, a file cannot have an entire disk for itself but has to have a maximum limit on the disk space it can occupy. In this approach, the filesystem keeps an entry in the open files table that points to the quota table and places limits (soft, hard) on users disk quota. There is one entry for each open file. **File System Backups** Another key task in filesystem management is backing up the filesystem so as to ensure recovery from disasters such as disk crashes. **File System Consistency** A disk crash before the file blocks are written out will potentially result in an inconsistent state. An Operating System therefore needs to provide utility programs that could help in checking consistency in blocks and files. Examples of such facilities include ***fsck*** in Unix and **scandisk** in Windows. Such utilities typically use two tables such that in one table they keep track of how many times a block is present in a file and the other is used to track how many times a block is present in the free list. Another approach is to look at files instead of blocks. In this approach a table of counters, one per file, is maintained. Starting from the root directory, the counter is incremented each time a file shows up in a directory. The OS then compares the counts with link counts from the i-nodes. These counts must be the same in order for the disk to be consistent. **File System Performance**

	Access to a disk is much slower than access to main memory. Because of this difference, many file systems have been designed with various optimizations to improve performance. In this session, you will be introduced to three of them, namely: caching, block read ahead. **Caching** Caching is the practice of implementing or utilizing a "cache". A cache (read as "cash") is a collection of blocks that logically belong on a disk but are being kept in memory for performance reasons. A cache is used to reduce disk access. It is at times referred to as "block cache" or "buffer cache". The most common operation of caching is to check all read requests to see if the needed block is in the cache. If it is, the read request can be satisfied without a disk access. If the block is not in the cache, it is first read into the cache and then copied to wherever it is needed. Subsequent requests for the same block can be satisfied from the cache. **Block Read Ahead** This is the second technique for performance management in filesystems. In this approach, attempts are made by the filesystem to fetch blocks into the cache before they are needed in order to increase the hit ratio. When the file system is asked to produce block k in a file, it does that, but when it is finished, it makes a sneaky check in the cache to see if block k + 1 is already there. If it is not, it schedules a read for block k + 1 in the hope that when it is needed, it will have already arrived in the cache. The worst case scenario is that if block k+1 will not have arrived in the cache, it will be on the way. **Reducing Disk-Arm Motion** This is the third approach to improving filesystem performance. This technique aims to reduce the amount of disk-arm motion by putting blocks that are likely to be accessed in sequence close to each other, preferably in the same cylinder. When an output file is written, the file system has to allocate the blocks one at a time, on demand.

ACTIVITY 2: READING READING MATERIAL 1 	In this section you are required to read the provided chapter so that you can further your understanding of the second objective of this course. a. File Management Andrew S Tanenbaum. (2016). Modern Operating Systems Paperback. Pearson. pp 263 - 325 https://www.amazon.com/Modern-Operating-Systems-Andrew-Tanenbaum/dp/9332575770#detailBullets_feature_div
ACTIVITY 3: Comprehension questions: 	1) Systems that support sequential files always have an operation to rewind files. Do systems that support random-access files need this, too? 2) Some operating systems provide a system call rename to give a file a new name. Is there any difference at all between using this call to rename a file and just copying the file to a new file with the new name, followed by deleting the old one? 3) Contiguous allocation of files leads to disk fragmentation, as mentioned in the text, because some space in the last disk block will be wasted in files whose length is not an integral number of blocks. Is this internal fragmentation or external fragmentation? 4) Describe the effects of a corrupted data block for a given file for: (a) contiguous, (b) linked, and (c) indexed (or table based). 5) Consider a file whose size varies between 4 KB and 4 MB during its lifetime. Which of the three allocation schemes (contiguous, linked and table/indexed) will be most appropriate? 6) For a given class, the student records are stored in a file. The records are randomly accessed and updated. Assume that each student's record is of fixed size. Which of the three allocation schemes (contiguous, linked and table/indexed) will be most appropriate?
LEARNING OUTCOME 2: Conceptual knowledge	

ACTIVITY 4: Video to be used.	
CASE 1:  	Two computer science students, John and Jane, are having a discussion about inodes. Jane asserts that memories have gotten so large and so cheap that when a file is opened, it is simpler and faster just to fetch a new copy of the i-node into the inode table, rather than search the entire table to see if it is already there. John thinks otherwise. In your opinion, who between these two students is right and why?
ACTIVITY 5: READING MATERIAL 	In this section you have been provided with links to several resources in filesystems. You are required to read all these online resources and use them to develop your blog as will be directed in the next section. 1) Filesystem in Operating Systems a) https://www.geeksforgeeks.org/file-system-implementation-in-operating-system/ b) https://www.geeksforgeeks.org/file-systems-in-operating-system/ c) https://www.tutorialspoint.com/operating_system/os_file_system.htm d) https://www.guru99.com/file-systems-operating-system.html#:~:text=Three%20types%20of%20files%20structure,series%20of%20functions%20and%20processes. 2) Filesystem Structure a) https://www.tutorialspoint.com/file-system-structure b) https://www.codecademy.com/resources/docs/general/file-system-structure c) https://www.ibm.com/docs/en/aix/7.2?topic=tree-file-system-structure d) https://www.javatpoint.com/os-file-system-structure e) https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/storage_administration_guide/ch-filesystem • Having read the above articles, you are required to write a blog in the LMS with focus on filesystem structure. In your blog also discuss the structure of filesystems used in Windows, Linux, UNIX, iOS and Android.
ACTIVITY 6: ONLINE DISCUSSION 	Your course instructor will create a discussion forum in the LMS to facilitate online group discussions. You are required to read the discussion topic and give comments. You are also encouraged to comment on contributions from at least three members of your group.

	You can use the LMS platform to send questions to your instructor on the discussion topics that he/she has posted on the LMS. The group discussion will be graded based on a weight that will be indicated on the LMS.
LEARNING OUTCOME 3: PRACTICAL SKILLS VIDEO 3: 	In this section you are provided with links to videos that present more information about file system structure. Watch them and then and use the information to work on the "lightning talk" activity below. 1) File System Operations (8:42 minutes) 2) The File Concept (17:12 minutes): 3) Directory Implementation (OS): 4) File Access Methods (8:05 minutes):
ACTIVITY 7: Learner practice sessions	In this session, you are required to do a "lightning talk" focusing on "Filesystem structure". In the "lightning talk", use your smartphone or any other video camera to record yourself in not more than "60 seconds" while explaining "***Filesystem structure and different file access methods***". Note: 1. Upload your video with the captions <*fname_lname_talk8*>. where "*fname*" is your first name and "*lname*" is your last name (or surname). The video must not be more than 60 seconds long.
ASSESSMENT OF PRACTICAL SKILL:	In the above activity, you uploaded your video, <***fname_lname_talk8***>. It will be assessed by the instructor by looking at among others: a) Accuracy of the assertions you have made (5 Marks) b) Degree of completeness of your response to the task (3 Marks) c) Adherence to the requirements with regards to topic and length of the video. (2 Marks)
LEARNING OUTCOME 4: KEY/TRANSFERABLE SKILLS	In this section you have been provided with links to online resources that discuss the filesystems of specific Operating Systems. You are required to read all of them and use the information to undertake the practical acitiviy provided in "activity 8" of this module: 1) UNIX/LINUX filesystem

	a) https://homepages.uc.edu/~thomam/Intro_Unix_Text/File_System.html#:~:text=The%20Unix%20file%20system%20is,of%20bytes%20(i.e.%20characters) b) https://www.geeksforgeeks.org/unix-file-system/ c) https://www.tutorialspoint.com/unix/unix-file-system.htm d) https://www.javatpoint.com/internal-structure-of-unix-file-system e) https://www.cis.rit.edu/class/simg211/unixintro/Filesystem.html 2) Windows Filesystem a) https://learn.microsoft.com/en-us/troubleshoot/windows-client/backup-and-storage/fat-hpfs-and-ntfs-file-systems b) https://www.makeuseof.com/ntfs-fat-exfat-windows-10-file-systems-explained/ 3) Apple iOS Filesystem a) https://docs.jamf.com/customer-education/jamf-100-course/5.0/Lesson_4_iOS_File_Storage_Structure.html#:~:text=The%20iOS%20file%20system%20contains,user%20volume%20contains%20user%20data. b) https://medium.com/@lucideus/understanding-the-ios-file-system-eee3dc87e455 c) https://developer.apple.com/library/archive/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html 4) Android Filesystem a) https://thesecmaster.com/explore-the-android-file-system-hierarchy-in-depth/ b) https://www.howtogeek.com/202644/how-to-manage-files-and-use-the-file-system-on-android/#:~:text=Android%20devices%20include%20a%20built,system%20and%20file%20manager%20apps. c) https://www.computerworld.com/article/3221287/android-file-management-an-easy-to-follow-guide.html d) https://medium.com/@aditi.kale20/file-system-of-android-a89dcbb693f1
ACTIVITY 8	In this module, you have learnt about several concepts in File Systems. You are required to write a two page essay on Filesystems used in Windows, UNIX, Android and Apple iOS. Your instructor will create an activity in the LMS that will allow you to submit this essay for assessment.

	The essay will be marked out of 15 Marks. Some of the guidelines to success in this activity include: a) Originality (avoid copying from the Internet and other sources) (5 Marks) b) Level of accuracy of the essay content (5 Marks) c) Completeness of content (3 Marks) d) Sticking to length (number of pages) requirements (1 Mark) e) Keeping to the theme (1 Mark)
QUIZZ: 	1. An “Inode” represents a) Buffer b) Data c) Files & Directories d) None of the mentioned 2. Journaling is preferred for a) Faster file system recovery b) Faster write operation c) Storing logs d) Storing metadata 3. Examples of Journaling filesystem a) Ext2 b) Ext3 c) UFS d) JFS 4. Hard links & soft links are same a) TRUE b) FALSE 5. ACL stands for a) ACCESS control list b) ACCESS check list c) Audit control list d) Audit check list 6. VFS a) Standalone filesystem b) Support multiple filesystem type c) Network filesystem d) None of the mentioned 7. Buffer cache helps to a) Store data b) Improved read/write performance c) Allocate memory d) None of the mentioned

	8. Wear leveling affects a) Hard disk b) Flash c) Optical storage d) RAM 9. Deframentation is the process of a) physically reorganizing the contents of the disk to store the pieces of each file close together and contiguously b) Create extra space in filesystem c) Resizing the filesystem d) None of the mentioned 10. The Superblock is required for a) Description of the basic size and shape of this file system b) This is the inode number of the first inode in the file system c) The number of free blocks in the file system d) All of the mentioned Answers 1. c) Files & Directories 2. a) Faster file system recovery 3. b) Ext3 4. b) FALSE 5. a) ACCESS control list 6. b) Support multiple filesystem type 7. b) Improved read/write performance 8. a) Hard disk 9. a) physically reorganizing the contents of the disk to store the pieces of each file close together and contiguously 10. d) All of the mentioned
TAKE HOME MESSAGE	Your course instructor will create a feedback section in the LMS to facilitate provision of your take home message. You are required to give a brief description of what you have learnt in this module in not more than half a page (typed) in the feedback section provided.
Reference list	1. Andrew S Tanenbaum. (2016). *Modern Operating Systems Paperback, 5th Edition*. Pearson. 2. Silberschatz A., Galvin P. B. and Gagne G. (2008). *Operating System Concepts*, 8th Edition. Wiley. ISBN: 9780470128725 3. Meyers, M. (2016). *CompTIA A+ Certification Guide*. McGraw-Hill Education