



THE OPEN UNIVERSITY OF KENYA

DESIGN PLAN

Programme title	Bachelor of Science in Cybersecurity and Digital Forensics
Course title	Operating Systems
Learning Module number	03
Learning module title	System Calls
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 the concept of System Calls in Operating Systems. We'll begin by watching lecture videos giving a detailed background on System calls. 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 System Calls in Operating Systems. We'll be looking at the basics of System Calls, their functions and application in specific areas of the Operating System.
Module objectives:	This module aims at facilitating learners to acquire knowledge about: <ol style="list-style-type: none">1. System calls and their functions;2. System calls for process management,3. System calls for file and directory management,4. Specific cases of System calls in UNIX and windows win32 API.
Module learning outcomes:	By the end of the module, you should be able to:

	<ol style="list-style-type: none"> 1. Define a system call and their functions; 2. Describe system calls for process management, 3. Categorize system calls for file and directory management, 4. Illustrate system calls used in UNIX and windows win32 API.
<p>Planned Learning Resources</p> <p>ACTIVITY 1: INTRODUCTION VIDEO 1: Pre-recorded lecture on topic emphasizing LEARNING OUTCOME 1: Factual knowledge.</p> 	<p>Video 1: Course Introduction (8 minutes)</p> <p>Welcome to this third module of the Operating Systems course. In this module, you will learn about concepts of System calls, that are at the core of operations of Operating Systems.</p> <p>In the second module of this course, we looked at among others, the components and structure of an Operating System. We saw that an Operating System abstracts the underlying details of the hardware by acting as an interface between the hardware and user applications. It also has a key function in management of computing resources. In managing the user programs, the Operating System manages the process of creating, writing, reading and deleting files and directories while management of computing resources happens in the background and is wholly transparent to the users of computers.</p> <p>The Operating System plays its role of interfacing between the underlying hardware and user applications through special routines that allow programs to request for services from the Operating System kernel. These special routines are referred to as System Calls. A system call therefore offers the services of an Operating System to user programs via Application Programming Interfaces (APIs).</p> <p><i>Insert a graphical illustration of a system call within the context of OS ecosystem:</i> <i>The image should be a layered diagram showing the following:</i> <i>Level 6: User programs</i> <i>Level 5: User Interface</i> <i>Level 4: System Calls</i> <i>Level 3: Program control, I/O, File System, Communications</i> <i>Level 2: Error management, Resources, Auditing, Security</i> <i>Level 1: Underlying Hardware</i></p> <p>A workflow of a system call be illustrated as follows: A user process runs in the user mode until it is interrupted by a system call. The system call is executed in the kernel mode and does the requested action. The system call finishes execution then returns control to the user program in the user mode.</p> <p><i>Insert a diagram that illustrates how system calls work.</i> <i>The figure should show both user and kernel modes. In user mode show:</i></p> <ol style="list-style-type: none"> <i>1. user process executing</i>

2. **User process receives a system call**
3. **System call executes {In the kernel mode}**
4. **Returns from System call**

Categories of System calls

System calls can be categorized into six different classes in tandem with their functions, namely:

1. Process Management

These include system calls such as:

- Those used in creation of processes, e.g. *fork* in UNIX and *NtCreateProcess* in Windows NT
- Process termination e.g. *exit*
- Others such as *load, execute, wait, allocate, free (e.g. for memory), signal etc.*

2. File Management. Examples include:

- create file, delete file
- open, close
- read, write, reposition
- get/set file attributes

3. Device Management

- request device, release device
- read, write, reposition
- get/set device attributes
- logically attach or detach devices

4. Information maintenance

- get/set total system information (including time, date, computer name, enterprise etc.)
- get/set process, file, or device metadata (including author, opener, creation time and date, etc.)

5. Communication management

These are used in transfer of information from one process or device to another. They include:

- create, delete (for communication connections)
- send, receive (for messages)
- transfer status information
- attach or detach (e.g. remote devices)

6. System protection

used for setting or getting protection for different aspects of the system. These include:




- get/set file permissions



ACTIVITY 2: READING
READING MATERIAL 1



1. System Calls

Andrew S Tanenbaum. (2016). Modern Operating Systems
Paperback. Pearson. pp 50 -61

	<p>https://www.amazon.com/Modern-Operating-Systems-Andrew-Tanenbaum/dp/9332575770#detailBullets_feature_div</p> <p>2. Other sources</p> <p>a) https://www.knowledgehut.com/blog/web-development/system-calls-in-os</p> <p>b) https://www.guru99.com/system-call-operating-system.html</p>
<p>ACTIVITY 3: Comprehension questions:</p> 	<ol style="list-style-type: none"> 1) Use a diagram to explain the series of events takes place when a process invokes the system call. 2) Describe the relationship between an API, the system-call interface, and the operating system. 3) Describe three general methods used to pass parameters to the operating system during system calls.
<p>LEARNING OUTCOME 2: Conceptual knowledge</p> <p>ACTIVITY 4: Video to be used.</p>	<p>In this section, you are provided with links to videos that you are expected to watch in order to enhance your understating of the concept of system calls. Watch them carefully then use them to undertake the activities that follow.</p>
<p>CASE 1:</p> 	<p>Describe case here.</p>
<p>ACTIVITY 5: READING MATERIAL</p> 	<p>In this section you are required to learn more about System calls used in UNIX and Win32 API.</p> <ol style="list-style-type: none"> 1) Windows System calls <ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/cpp/c-runtime-library/system-calls?view=msvc-170 2) Low-level I/O: <ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/cpp/c-runtime-library/low-level-i-o?view=msvc-170 3) File handling: <ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/cpp/c-runtime-library/file-handling?view=msvc-170 4) Directory control: <ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/cpp/c-runtime-library/directory-control?view=msvc-170

	<ul style="list-style-type: none"> Use the information learnt above to write a blog on how to program system calls in Windows. In your blog give an example using C programming language and Win32 APIs.
<p>ACTIVITY 6: ONLINE DISCUSSION</p> 	<p>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.</p> <p>You can use the LMS platform to send questions to your instructor on the discussion topics that he/she has posted on the LMS.</p> <p>The group discussion will be graded based on a weight that will be indicated on the LMS.</p>
<p>LEARNING OUTCOME 3: PRACTICAL SKILLS VIDEO 3:</p> 	<p>In this section we shall be watching videos that show how to use system calls in practice.</p> <ol style="list-style-type: none"> 1) First let's begin by watching this 12 minute video that explains more concepts about System calls. 2) Then Watch this 20-minute video, in which we take a practical view of how fork() and exec() system calls in UNIX can be applied using sample code in C programming language.
<p>ACTIVITY 7: Learner practice sessions</p>	<p>The simple C program below demonstrates the use of system calls. It is intended to be run in a UNIX environment.</p> <p>You are required to go through it line by line taking note of each system call that has been used then use it to complete the task that follows.</p> <pre> #include <unistd.h> #include <fcntl.h> #include <sys/stat.h> #include <sys/types.h> #include <stdio.h> int main() { int n, fd; char buff[100]; // declaring buffer printf("Enter text to write in the file:\n"); n = read(0, buff, 100); fd = open("file", O_CREAT O_RDWR, 0777); write(fd, buff, n); write(1, buff, n); int close(int fd); </pre>

	<pre>return 0; }</pre>
ASSESSMENT OF PRACTICAL SKILL:	<ol style="list-style-type: none"> 1) Re-write the simple C program above giving comments that describe what each line of code does. 2) Give sample Output from the above code. 3) Explain the parameters of the system calls that you have identified in the program above.
LEARNING OUTCOME 4: KEY/TRANSFERABLE SKILLS	<p>In this section, you are provided with online resources for further reading in order to know more about system calls:</p> <ol style="list-style-type: none"> 1) System Call in OS <ul style="list-style-type: none"> • https://www.guru99.com/system-call-operating-system.html 2) Introduction of System Call: <ul style="list-style-type: none"> • https://www.geeksforgeeks.org/introduction-of-system-call/ 3) System Calls in Operating System (OS) <ul style="list-style-type: none"> • https://www.javatpoint.com/system-calls-in-operating-system • System calls in unix and windows: https://www.tutorialspoint.com/system-calls-in-unix-and-windows#:~:text=System%20calls%20in%20Windows%20are,system%20using%20the%20system%20calls.
ACTIVITY 8	<p>In this module, you have learnt about several concepts in System calls. You are required to write a two page essay on how system calls work in Operating Systems.</p> <p>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.</p> <p>Some of the guidelines to success in this activity include:</p> <ol style="list-style-type: none"> 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:	<ol style="list-style-type: none"> 1. A system call is a routine built into the kernel and performs a basic function. <ol style="list-style-type: none"> a) True



b) False

Answer: a

2. When we execute a C program, CPU runs in ____ mode.

- a) user
- b) kernel
- c) supervisory
- d) system

Answer: a

3. In ____ mode, the kernel runs on behalf of the user.

- a) user
- b) kernel
- c) real
- d) all

Answer: b

4. All UNIX and LINUX systems have one thing in common which is ____

- a) set of system calls
- b) set of commands
- c) set of instructions
- d) set of text editors

Answer: a

5. The chmod command invokes the ____ system call.

- a) chmod
- b) ch
- c) read
- d) change

Answer: a

6. For reading input, which of the following system call is used?

- a) write
- b) rd
- c) read
- d) change

Answer: c

7. Which of the following system call is used for opening or creating a file?

- a) read
- b) write

	<p>c) open d) close</p> <p>Answer: c</p> <p>10. open system call returns the file descriptor as ____ a) int b) float c) char d) double</p> <p>Answer: c</p> <p>9. Which of the following mode is used for opening a file in both reading and writing? a) O_RDONLY b) O_WRONLY c) O_RDWR d) O_WDR</p> <p>Answer: c</p> <p>2. perror also uses error number to print the error messages. a) True b) False Answer: b</p>
<p>TAKE HOME MESSAGE</p>	<p>Your course instructor will create a feedback section in the LMS to facilitate provision of your take home message.</p> <p>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.</p>
<p>Reference list</p>	<ol style="list-style-type: none"> 1. Andrew S Tanenbaum. (2016). <i>Modern Operating Systems Paperback, 5th Edition</i>. Pearson. 2. Silberschatz A., Galvin P. B. and Gagne G. (2008). <i>Operating System Concepts, 8th Edition</i>. Wiley. ISBN: 9780470128725 3. Meyers, M. (2016). <i>CompTIA A+ Certification Guide</i>. McGraw- Hill Education