Module Code & Module Title

CS5001NA Networks and Operating System

Assessment Weightage & Type

20% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Karmaraj Giri

London Met ID: 20048909

College ID: np01cp4s210233

Assignment Due Date:

Assignment Submission Date: 7 Aug 2022

Word Count (TASK B):

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

This course teaches students how to program for the LINUX operating system. This coursework is divided into two parts. Making script files from Linux is required for the first assignment in this course. The program's code is contained in the script. In-depth examination of the Process management system is covered in the second section of the curriculum.

Table of Contents

1. Task A 5
1.1 Introduction 5
1.2 Script 6
1.2.1 Sub files 9
File 1 (AD) 9
File 2 (AY) 9
File 3 (FM) 10
File 4 (JL) 10
1.3 Testing 10
1.3.1 Test 1 10
1.3.2 Test 2 10
1.3.3 Test 3 11
1.3.4 Test 4 12
1.3.5 Test 5 12
1.3.6 Test 6 13
1.3.7 Test 7 13
1.3.8 Test 8 14
1.3.9 Test 9 15
1.3.10 Test 10 16
1.4 Conclusion 17
2. Task B 17
2.1 Introduction 17
2.2 Background 18
2.3 Process architecture 19
2.4 Process Hierarchies 20
2.5 Deadlock 20
2.6 Process scheduling 21

2.6.1 Process States 21
2.6.2 Process Control Block (PCB) 22
2.6.3 Policies and Algorithm 23
2.7 Inter - Process Communication 24
2.8 Implementation of Process 24
2.9 Conclusion 25
3. Bibliography 26

Table of Tables

Table 1 Test 1....10
Table 2 Test 2....11
Table 3 Test 3....11
Table 4 Test 4....12
Table 5 Test 5....13
Table 6 Test 6....13
Table 7 Test 7....14
Table 8 Test 8....14
Table 9 Test 9....15
Table 10 Test 10....16

Table of Figures

Figure 1 Script 1 6
Figure 2 Script 2 7
Figure 3 Script 3 8
Figure 4 Script 4 9
Figure 5 Test 1 10
Figure 6 Test 2 11
Figure 7 Test 3 12
Figure 8 Test 4 12
Figure 9 Test 5 13
Figure 10 Test 6 13
Figure 11 Test 7 14
Figure 12 Test 8 15
Figure 13 Test 9 16
Figure 14 Test 10 16
Figure 15 Process Architecture 19
Figure 16 Process States Model.... 21
Figure 17 Process Control Block (PCB).... 22

1. Task A

1.1 Introduction

An operating system called LINUX was created in the 1990s and has continued to be improved ever since. A robust, multi-user, multi-tasking operating system for servers, desktop computers, and laptops is LINUX. It also features a graphical user interface (GUI) that is akin to Microsoft Windows, which makes it easier to use. However, knowledge of LINUX is necessary for operations that aren't covered by a graphical program or for situations in which there isn't a windows interface accessible. Although there are various LINUX versions, they all have some things in common. On our server and PCs, the most widely used LINUX distributions are Ubuntu, Debian, Red Hat, and Arch.
The four fundamentals are the primary idea behind LINUX. Kernel, Shell, Command and Utilities, Files and Directories, and Fourth (Saini, 2021)

1. **The Kernal**

   The operating system's foundation is known as the kernel. It always operates in main memory and carries out crucial duties such handling disk input and output and memory management. In response to system calls, the kernel's primary function is to manage file storage and communication while allocating time and memory to programs. (Adams, 2019)

2. **The Shell**

   An interface between the user and the kernel is what is referred to as the Shell. The Shell is a command line interpreter (CLI), which organizes for the execution of the commands entered by the user. For every command, it uses the standard syntax. With the majority of LINUX variations, there are three regularly used Shells. The C shell, Bourne Shell, and Korn Shell are these. (Adams, 2019)

3. **Commands and Utilities**

   There are several utilities and commands. There are more than 250 standard commands in addition to a large number of extras offered by third

party software. Cp, mv, cat, grep, and other commands and utilities are a few examples. (Adams, 2019)

4. **Files and Directories**

   Data is gathered into files, which are then further arranged into directories, which are then grouped together into a structure known as the filesystem. (Adams, 2019)

1.2 Script

```
  GNU nano 2.0.6                                                File: 20048909CW2P2

#! /bin/bash

exitConfirmation(){
    echo -e "\nWould you like to close the program or not? (y|n): \c"
    read ans
    if [ ${ans,,} == y ]
    then
        echo Exitted at $(date).
        cat thankyou
        exit
    fi
    startapp
}

getBestMember(){
    PS3='Select the number from the list to display the information of
the member: '
    select member in ${choose[@]}
    do
        if [ -z $member ]
        then
            echo Please Try Again! Choose from the above list only
            getBestMember
        fi
        echo You chose $member as your favourite member.
        if [ -e $member ]
        then
                cat FM
        fi
        cat $member
        exitConfirmation
    done
}

memberErr(){
    echo -e "Members are repeating or there is no such member in the
list\nYou can only choose 3 members\nPlease try Again!"
    validateMember
}

validateMember(){
    echo -e "Enter any 3 members' code from the above list:- \c"
    read m1 m2 m3
    declare -a choose=($m1 $m2 $m3)
    for i in ${choose[@]}
    do
        if [[ $i = JL || $i = AY || $i = FM || $i = DH || $i = KC ]]
        then
                continue
        fi
        memberErr
    done
    if [[ ${#choose[@]} -eq 3 && $m1 != $m2 && $m1 != $m3 && $m2 != $m3
]]
    then
        getBestMember
    fi
    memberErr
 }

getOptionsMenu(){
    echo Welcome
echo Name:            $uname
echo ID:              $uid
echo Login time:      $(date)

echo '
Top 5 bands
-------------------------------------------------

^G Get Help    ^O WriteOut    ^R Read File    ^Y Prev Page
^X Exit        ^J Justify     ^W Where Is     ^V Next Page
```

Figure 1 Script 1

```
echo '
Top 5 bands
-------------------------------------------------
|    Band name       |      Band code           |
-------------------------------------------------
|    Beatles         |      BEA                 |
|    AC/DC           |      AD                  |
|    Queen           |      QUE                 |
|    Blondie         |      BLO                 |
|    Nirvana         |      NIR                 |
-------------------------------------------------

Choose one band code from the above list
    '
    until [ "$band" == AD ]
    do
        echo -e "your guess for best band: \c"
        read band
        if [ $band != AD ]
        then
            echo This is not the right band. Please Try Again.
        fi
    done
    echo -e "Congratulations! you chose AC/DC\n"
    cat AD
    band=''
    echo '
Top 5 members
-------------------------------------------------
|      Member Name     |     Member Code         |
-------------------------------------------------
|      John Lennon     |     JL                  |
|      Angus Young     |     AY                  |
|      Freddie MERCURY |     FM                  |
|      Debbie Harry    |     DH                  |
|      Kurt Cobian     |     KC                  |
-------------------------------------------------
```

Figure 2 Script 2

```
startapp(){
    getOptionsMenu

    validateMember
}

counter=0
while [ $counter -lt 3 ]
do
    echo -e "Enter secret key: \c"
    read -s key
    if [ $key == Karmaraj ]
    then
        if [ $# -ne 2 ]
        then
            echo -e 'Exactly 2 arguments required to run the program.
\nFirst: your name. \nSecond: your ID.'
                break
        fi
        if [[ $1 =~ [0-9] || $2 =~ [aA-zZ] ]]
        then
            echo Invalid input. The first input must be your name and
The second input must be your ID.
        break
        fi
        uname=$1
        uid=$2
        startapp
        break
    fi
    echo You have $(( 2-$counter )) tries remaining
```

Figure 3 Script 3

```
startapp(){
    getOptionsMenu

    validateMember
}

counter=0
while [ $counter -lt 3 ]
do
    echo -e "Enter secret key: \c"
    read -s key
    if [ $key == Karmaraj ]
    then
        if [ $# -ne 2 ]
        then
            echo -e 'Exactly 2 arguments required to run the program.
\nFirst: your name. \nSecond: your ID.'
                break
        fi
        if [[ $1 =~ [0-9] || $2 =~ [aA-zZ] ]]
        then
            echo Invalid input. The first input must be your name and
The second input must be your ID.
        break
        fi
        uname=$1
        uid=$2
        startapp
        break
    fi
    echo You have $(( 2-$counter )) tries remaining
    (( counter++ ))
done

echo Exitted at $(date).
```

Figure 4 Script 4

1.2.1 Sub files

File 1 (AD)

Australian rock group AC/DC was founded in 1973 by Scottish-born brothers Malcolm and Angus Young in Sydney. The band simply refers to their music as "rock & roll," yet it has been categorized as hard rock, blues rock, and heavy metal.

File 2 (AY)

Australian musician Angus McKinnon Young, who was born on March 31, 1955, is best known as the co-founder, main guitarist, songwriter, and lone

surviving original member of the hard rock group AC/DC. He is well-known for his upbeat performances, stage attire that resembles school uniforms, and his rendition of Chuck Berry's duckwalk.

File 3 (FM)

Farrokh Bulsara, better known by his stage name Freddie Mercury, was a British singer-songwriter best known for serving as the band Queen's lead vocalist from 1956 until 1991.

File 4 (JL)

John Winston Ono Lennon, also known as John Winston Lennon (born John Winston Lennon; 9 October 1940 – 8 December 1980), was an English singer, songwriter, musician, and peace activist. He is best known for being the founder, co-songwriter, co-lead vocalist, and rhythm guitarist of the Beatles, which brought him international fame.

1.3 Testing

1.3.1 Test 1

Test No	1
Task	Program must execute with correct parameters
Code Entered	Bash 20048909CW2P2 Karmaraj 94029
Expected Result	The program should ask for secret key.
Actual Result	The program asked for secret key.

Table 1 Test 1

```
[Karmarajs-MBP:~ karmarajgiri$ bash 20048909CW2P2 Karmaraj 94029
[Enter secret key: Welcome
```

Figure 5 Test 1

1.3.2 Test 2

Test No	2
Task	Correct key is entered

Key Entered	Karmaraj
Expected Result	List of all band must be displayed
Actual Result	List of the band was displayed

Table 2 Test 2

```
[Enter secret key: Welcome
Name:
ID: 94029
Login time: Sun Aug 7 13:12:37 +0545 2022

Top 5 bands
------------------------------
| Band name | Band code        |
------------------------------
| Beatles   |     BEA          |
| AC/DC     |     AD           |
| Queen     |     QUE          |
| Blondie   |     BLO          |
| Nirvana   |     NIR          |
 ------------------------------

Choose one band code from the above list
```

Figure 6 Test 2

1.3.3 Test 3

Test No	3
Task	Give correct
Code Entered	AD
Expected Result	List of member must appear with band description
Actual Result	List of member appeared with band description

Table 3 Test 3

```
your guess for best band: AD
Congratulations! you chose AC/DC

Australian rock group AC/DC was founded in 1973 by Scottish-born brothers Malcolm and Angus Young in Sydney. The band simply refers to their music as "rock & roll," yet it has been categorized as hard rock, blues rock, and heavy metal.

Top 5 members
---------------------------------------
|    Member Name   |  Member Code  |
 ---------------------------------------
| John Lennon      |  JL           |
| Angus Young      |  AY           |
| Freddie Mercury  |  FM           |
| Debbie Harry     |  DH           |
| Kurt Cobian      |  KC           |
---------------------------------------
```

Figure 7 Test 3

1.3.4 Test 4

Test No	4
Task	Choose band member code
Code Entered	AY FM DH
Expected Result	List of three member should appear asking to choose for information
Actual Result	List of three member appeard asking to choose for information

Table 4 Test 4

```
Top 5 members
---------------------------------------
|     Member Name    |   Member Code    |
 ---------------------------------------
| John Lennon        |   JL             |
| Angus Young        |   AY             |
| Freddie Mercury    |   FM             |
| Debbie Harry       |   DH             |
| Kurt Cobian        |   KC             |
---------------------------------------

Enter any 3 members' code from the above list:-  AY FM DH
1) AY
2) FM
3) DH
```

Figure 8 Test 4

1.3.5 Test 5

Test No	5
Task	Choose band members for information
Code Entered	1
Expected Result	Information of chose band members should be displayed
Actual Result	Information of chose band members was displayed

Table 5 Test 5

```
Enter any 3 members' code from the above list:-  AY FM DH
1) AY
2) FM
3) DH
Select the number from the list to display the information of
the member: 1
You chose AY as your favourite member.
Farrokh Bulsara, better known by his stage name Freddie Mercury, was a British singer-songwriter best known for serving as the
Australian musician Angus McKinnon Young, who was born on March 31, 1955, is best known as the co-founder, main guitarist, song
ormances, stage attire that resembles school uniforms, and his rendition of Chuck Berry's duckwalk.
```

Figure 9 Test 5

1.3.6 Test 6

Test No	6
Task	Enter y for closing the program
Code Entered	y
Expected Result	Program must be terminated showing exited date time and saying thank you
Actual Result	Program was terminated showing exited date time and saying thank you

Table 6 Test 6

```
Would you like to close the program or not? (y|n): y
20048909CW2P2: line 6: ${ans,,}: bad substitution
Exitted at Sun Aug 7 13:26:56 +0545 2022.
```

Figure 10 Test 6

1.3.7 Test 7

Test No	7
Task	Enter wrong secret key
Code Entered	Random key
Expected Result	Program must terminate after 3 chances
Actual Result	Program was terminated after 3 incorrect key

Table 7 Test 7

```
Karmarajs-MBP:~ karmarajgiri$ bash 20048909CW2P2 Karmaraj 94029
Enter secret key: You have 2 tries remaining
Enter secret key: You have 1 tries remaining
Enter secret key: You have 0 tries remaining
Exitted at Sun Aug 7 13:32:05 +0545 2022.
Karmarajs-MBP:~ karmarajgiri$
```

Figure 11 Test 7

1.3.8 Test 8

Test No	8
Task	Enter wrong band code
Code Entered	AZ, KX
Expected Result	Message must appear saying try again
Actual Result	Message saying try again appeared.

Table 8 Test 8

```
Top 5 bands
[------------------------------
[| Band name | Band code        |
 ------------------------------
 | Beatles   |     BEA          |
 | AC/DC     |     AD           |
 | Queen     |     QUE          |
 | Blondie   |     BLO          |
 | Nirvana   |     NIR          |
  ------------------------------

Choose one band code from the above list

your guess for best band: AZ
This is not the right band. Please Try Again.
your guess for best band: KX
This is not the right band. Please Try Again.
```

Figure 12 Test 8

1.3.9 Test 9

Test No	9
Task	Choosing member code whose information is not available
Code Entered	2
Expected Result	Message must appear saying no file or directory
Actual Result	Message saying no file or directory appeared.

Table 9 Test 9

```
Top 5 members
--------------------------------------
|     Member Name    |   Member Code   |
 -------------------------------------
| John Lennon        |   JL            |
| Angus Young        |   AY            |
| Freddie Mercury    |   FM            |
| Debbie Harry       |   DH            |
| Kurt Cobian        |   KC            |
--------------------------------------

Enter any 3 members' code from the above list:- JL DH KC
1) JL
2) DH
3) KC
Select the number from the list to display the information of
the member: 2
You chose DH as your favourite member.
cat: DH: No such file or directory
```

Figure 13 Test 9

1.3.10 Test 10

Test No	10
Task	Choose ‘n’ to continue the program
Code Entered	‘n’
Expected Result	Program must continue
Actual Result	Program continued

Table 10 Test 10

```
Would you like to close the program or not? (y|n): n
20048909CW2P2: line 6: ${ans,,}: bad substitution
Exitted at Sun Aug 7 13:31:20 +0545 2022.
```

Figure 14 Test 10

1.4 Conclusion

Because it includes all the code and testing, the initial portion of the coursework was fun to complete. If you are familiar with the task at hand, this portion is simple. We study how to create programs under the LINUX operating system in this section of the coursework. We discovered how to create script files in many shells. Making and editing the script was challenging at first, but with the assistance of our module leader and our module teacher, I was able to finish this assignment and learn about programming in LINUX.

2. Task B

2.1 Introduction

A computer's operating system is a piece of software that controls every component. connects to your computer and lets you control its resources. The operating system controls all of the following: devices, software, data, and networks. A process resembles an active piece of software. A process needs specific resources, such as CPU time, memory, files, and I/O devices, in order to finish its work. These resources are allocated to processes as they are created or operated. The process is the unit of work for the majority of systems. The system consists of various processes. Operating system processes run system code, whereas user processes run user code. These processes can all be running simultaneously. (Hardy, 2020)

Although most modern operating systems now support multithreaded processes, traditionally a process ran with just one control thread. The following thread and process management tasks are handled by the operating system: Processes produced by users and systems can be created and deleted. Process planning offers processes for synchronizing processes, communicating amongst processes, and resolving deadlocks. The process is what happens when a program is executed and the actions specified in that program are completed. This can be a specification of the execution unit in which the program runs. CPU processes can be started, scheduled, and ended with the help of the operating system. Child processes are those that the main process creates. Process control blocks (PCBs) make it easy to regulate process flows. (Hardy, 2020)

Process development, planning, termination, and dead ends are all included in the broad category of operations known as "process management." Currently used operating systems must have processes, which are running programs. Resources must be allocated by the operating system to allow processes to share and exchange data. Additionally, it allows for process synchronization and protects each process' resources from outside approaches. An operating system task is the management of all the processes that are active on the system. Tasks that can be used to manage operations include process planning and resource allocation. (Hardy, 2020)

UNIX's parent-child process management model is followed by Linux, but it also introduces the concept of "personality," which permits processes from other operating systems to function. Each process has a unique execution domain associated with it. How system calls and messages to the process are handled is defined by the execution domain. Based on predetermined criteria, the Linux scheduler analyzes the list of READY processes and selects one to execute. You have a choice of three scheduling options in the scheduler. Real-time processes are covered by the first two, whereas routine activities are covered by the third. Based on the nature and priority of the processes, the scheduler chooses the scheduling strategy for those in the READY queue. (Hardy, 2020)

2.1.1 Aims and Objectives

- Learning about an operating system's process management mechanism.
- Understanding the creation of a process
- Studying the hierarchy and process architecture.
- Finding out about IPC (Inter Process Communication).
- Finding out about Process Control Blocks (PCB).
- Understanding the true meaning of an inter-process communication.
- Recognizing how a method is put into practice.
- Learning the basics of multi-threading and threads.

2.2 Background

Early computers could only execute one program at a time. The system was entirely under the authority of this application, and all of its resources were at its disposal. On the other hand, modern computer systems enable the simultaneous

operation of numerous applications that are loaded into memory. The concept of processes, or running programs, arises as a result of the evolution's requirement for tighter management and separation of distinct applications. The unit of work in a contemporary timeshare system is called a process. The system consists of various processes. a system process that runs system code as well as a user process that runs user code. The possibility exists that all of these programs could run concurrently while sharing the CPU.

By switching CPUs between processes, the operating system can increase computer productivity. Early computers could only execute one program at a time. It's not exactly how it is right now. Finally, it's time to divide your time. The increased compartmentalization necessitated better control. Despite these developments, the computer still has a CPU, a programme counter, and a virtual CPU that the actual CPU uses to move between tasks.

2.3 Process architecture

When a program is loaded into memory and converted into a process, it can be broken down into the following four components: stack, heap, text, and data. A process in main memory is represented simply in Figure.

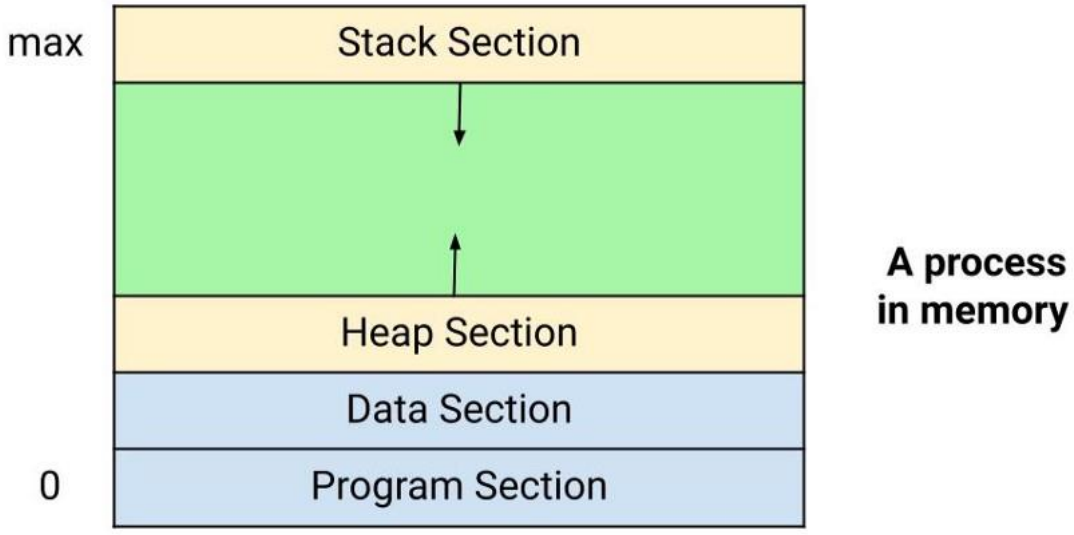

Figure 15 Process Architecture

Despite the fact that two processes can be connected to the same program, they are viewed as two separate execution paths. For instance, different versions of the email

client may be used by many users, and different web browsers may be used by one user at a time. Each of these processes is unique; they all have comparable body portions but different data, heap, and stack sections. Below is a brief summary of each section:-

- **Stack** - The method stack stores transient data such as method/function parameters, return addresses, and local variables.
- **Heap** - This is RAM that is dynamically distributed to a process while it is running.
- **Data** - The global and static variables are contained in this section.
- **Text** - The contents of the processor's registers and the program counter value's current value's activity are included in this.

2.4 Process Hierarchies

In computer systems, several processes must run concurrently, and certain processes must generate new processes as they operate. The parent and child processes often connect in a precise way when one process spawns another. Child processes can also generate other processes as necessary. This parent-child-like process structure creates the process hierarchy. On UNIX, a process and all of its children and grandchildren make up a process group. Any process group members to whom the keyboard is now connected will receive any signals given by the user through the keyboard. The signal can be intercepted by each process, which can then do its own tasks.

2.5 Deadlock

A system-wide resource request triangle is produced when two or more jobs are suspended while waiting for the availability of a critical resource. This issue occurs when other jobs that are prepared to run but are unable to do so because of a shortage of other resources occupy the resources needed for these jobs. increase. The deadlock is broken when the rest of the system fails. If the operating system cannot resolve the issue, you must stop the procedure.

Deadlock is more serious than protracted postponement or hunger since it affects multiple jobs. Resource binding makes the entire system "not simply a few apps." I

was affected. Deadlocks have increased in frequency with the emergence of interactive systems that optimize resource utilization through dynamic resource sharing, however this feature also increases the likelihood of deadlocks.
In certain systems, deadlocks are seen as an inconvenience that adds to the wait time. On the other hand, deadlocks in real-time systems might be harmful. The operating system must stop or end a stalemate when it happens. Deadlock is more serious than protracted postponement or hunger since it affects multiple jobs. Resource binding makes the entire system "not simply a few apps." I was affected. Deadlocks have increased in frequency with the emergence of interactive systems that optimize resource utilization through dynamic resource sharing, however this feature also increases the likelihood of deadlocks. In certain systems, deadlocks are seen as an inconvenience that adds to the wait time. On the other hand, deadlocks in real-time systems might be harmful. The operating system must stop or end a stalemate when it happens.

2.6 Process scheduling

Process scheduling, which entails removing the active process from the CPU and choosing a different process based on a certain strategy, is the responsibility of the process manager. Process scheduling in a multiprogramming operating system is crucial. In these operating systems, numerous processes can be put into executable memory at once, and they can share the CPU by using temporal multiplexing.

2.6.1 Process States

During operation, a process's state can change. The status of a process can be determined in part by its current activity. Every process is in a different state—new, operating, waiting, ready, or terminated.

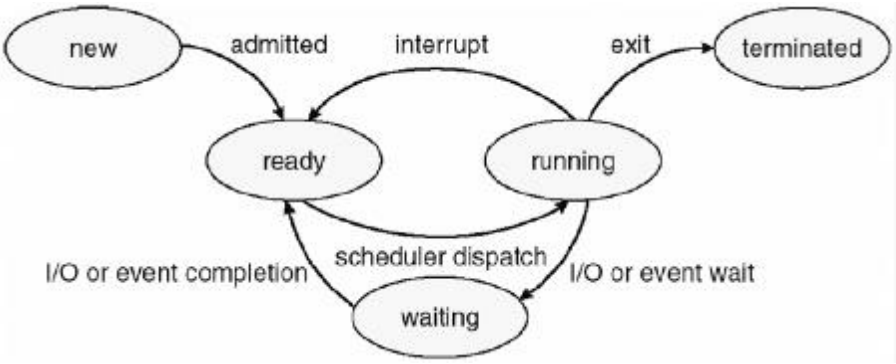

Figure 16 Process States Model

The following describes the process states: -

- **New** - A process is seen as being in a new state when it is just being created. During operation, a process's state can change. The status of a process can be determined in part by its current activity.
- **Ready** - All processes that have been loaded into primary memory and are awaiting the CPU are said to be in the ready state.
- **Running** - The state in which all active processes are is referred to as running.
- **Waiting** - The waiting state refers to any processes that leave the CPU for any reason (I/O or a high-priority task), and then wait to be executed.
- **Terminated** - A process that has left the CPU and main memory is said to be in the terminated state.

2.6.2 Process Control Block (PCB)

A process control block (PCB), often referred to as a task control block, serves as the operating system's representation of each process. In Figure 13, a printed circuit board is shown. It includes various information regarding a certain process, such as the following:

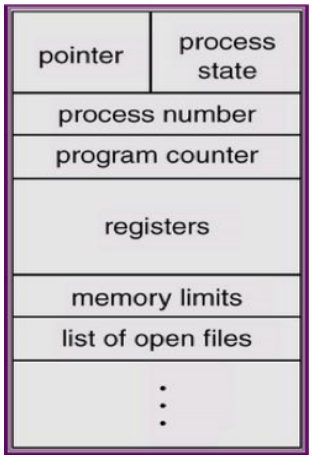

Figure 17 Process Control Block (PCB)

A Process Control Block is a data structure that the Operating System provides for each process. Integer process ID is used to identify the PCB (PID). Below is a description of how each PCB component functions.

Pointer - A stack pointer that needs to be stored whenever a process changes from one state to another in order to preserve its current location.

Process state - It keeps track of each process's current status.

Process number - Process ID or PID, which is used to store the process identifier, is given to each process as a special identification.
Process counter - This keeps track of the counter, which contains the address of the subsequent instruction that needs to be executed by the process.
Register - Among the CPU registers are the accumulator, base, registers, and general-purpose registers.
Memory limits - Details about the operating system's memory management system are contained in this area. Page tables, segment tables, and other types of tables may be included.
Open files list - This data contains a list of all the files that a process has opened.

2.6.3 Policies and Algorithm

There are two categories of scheduling procedures:

1) **Preemptive Scheduling** - Preemptive scheduling is a scheduling technique in which a running process may be terminated if a higher priority process joins the queue and is given CPU time. In this case, the high priority process uses the CPU cycle and the current process switches from the running queue to the ready queue.
2) **Non – Preemptive Scheduling** - A running process cannot be interrupted by any other process while scheduling is non-preemptive. Once the current process has finished its CPU cycle, any new processes that join the queue must wait until that point before starting. Let's utilize the same illustration but with non-preemptive scheduling this time.

   There are also six different categories of process scheduling algorithms:

   1) **First Come First Serve (FCFS)** - The implementation of this algorithm is the easiest. Priority will be given to the process that arrives first. The arrival time decreases as the CPU is delivered to the process more quickly. It is not preemptive in its scheduling approach.
   2) **Round Robin Scheduling** - The OS's Round Robin scheduling algorithm specifies a time quantum (slice). Cycles will be used to carry out every process. For a brief length of time (referred to as the time quantum), each process will have access to the CPU, after which it will return to the ready queue to wait for its turn. It is a form of planning that is done in advance.

3) S**hortest** – Job First (SJF) Scheduling - The CPU will initially be assigned to the task with the shortest burst time. The task will obtain CPU time more quickly the shorter the burst period is. This type of scheduling is non-preemptive.
4) **Short Remaining Time (SRT) Scheduling** - This particular form of SJF is employed as a prophylactic approach. Using this method, the OS schedules the Job based on the remaining execution time.
5) **Priority Based Scheduling** - With this approach, each procedure will be given priority. The rate at which the job receives CPU time depends on its priority. The two processes will be ordered in accordance with their arrival time if their priorities are equal.
6) **Multiple Level Queue Scheduling** - In multiple level queue scheduling, the ready queue is separated into numerous queues, and processes are assigned to queues based on their unique characteristics, such as memory size, process priority, and so forth. For each queue, a different scheduling algorithm might be utilized.

2.7 Inter - Process Communication

Inter-process communication in an operating system refers to the capacity for numerous processes to communicate with one another. Shared memory, message queues, FIFO, and other techniques are some of the ways IPC can be implemented in an OS.

A system has two different types of processes: independent and collaborative. Processes that work together have an effect on one another and may share information and data. A mechanism for transmitting data and information between several processes running on one or more computers connected by a network is called inter - process communication, or IPC.

2.8 Implementation of Process

The process model is implemented by the operating system using a table (an array of structures) called the process table, with one entry per process. (Some authors refer to these components as process control blocks.) The programme counter, stack pointer, memory allocation, open file status, accounting and scheduling data,

and any other data that needs to be saved when the process is changed from running to ready or blocked so that it can be restarted as if it had never been stopped are all contained in this entry.

2.9 Conclusion

A program that is currently running is called a process. The state evolves as the process goes on. The state of a process is determined by its current activity. One can start, halt, restart, or terminate any process. Each process in the operating system is represented by a process control block (PCB). Processes are queued while they are not in use. The two types of queues in the operating system are I/O request queues and ready queues. The Ready Queue contains all of the prepared processes that are awaiting the CPU's execution. Each process is represented by a PCB, which may be combined to form an entire queue. Long-term (job) scheduling is the process of choosing processes that can compete for CPU time. In long-term planning, memory management, in particular, is frequently a major consideration. Short-term (CPU) scheduling is the process of choosing processes from the ready queue. The operating system must offer a way for the parent process to spawn a new child process. Parents have two options: they can run both themselves and their child at once, or they can wait for their child to finish. Concurrency is advantageous for a number of reasons. Advantages include information sharing, quick calculations, modularity, and convenience. Operating system processes can be either independent or collaborative. Collaborative processes need an inter - process communication mechanism in order to communicate with one another. The two ways that communication is theoretically performed are message passing and shared memory. The shared memory architecture requires interaction between processes in order to share some variables. Processes are anticipated to exchange data using these shared variables. In a shared memory system, communication is established by the application programmer. Shared memory is the sole thing that the operating system must offer. The message forwarding technique can be used by processes to communicate. Communication is facilitated by the operating system. These two systems can be used simultaneously in the same operating system and are not mutually exclusive.

3. Bibliography

Adams, D., 2019. *linuxhint.* [Online]
Available at: https://linuxhint.com/introduction_linux_kernel/
[Accessed 21 Jan 2019].
Hardy, J., 2020. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/operating_system/index.htm#
[Accessed 12 Mar 2020].
Saini, A., 2021. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-to-linux-operating-system/
[Accessed 18 Feb 2021].