Module Code & Module Title

CS5001NI Networks and Operating System

Assessment Weightage & Type

20% Individual Coursework

Year and Semester

2021-22 Autumn

Student Name: Punam Thapa Magar

London Met ID: 20048968

College ID: np01cp4s210273

Assignment Due Date: 25th April 2022

Assignment Submission Date:25th April 2022

Title: Unix script programming

Word Count (Task B):2002 words

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.

Table of Contents

List of Tables

List of Figures

Task A

Introduction

The Unix operating system is a set of programs that act as a link between the computer and the user. The computer programs that allocate the system resources and coordinate all the details of the computer internally are called the operating system or the kernel. Users communicate with the kernel through a program known as the shell. The shell is a command-line interpreter which translates commands entered by the user and converts them into a language that is understood by the kernel. There are various commands and utilities which we can make use of in our day-to-day activities. cp, mv, cat, grep, ls, mkdir, touch, chmod, cp, etc (tutorialspoint, 2022).

For this coursework, we have utilized Debian as the terminal. All the task was done in the bash file and the file was tested in the terminal. Moreover, Debian is a conspicuous and uninhibitedly available PC working system that uses the Linus kernel, and other program portions from the GNU venture and it is perhaps the most popular desktop Linux distribution and is utilized by a huge number of the people all over the world. It is alike liberated from cost and open source. In this coursework, a large number of the Unix utility command is utilized to finish it. Along with that scripting language is utilized to foster the user interactive program. Here the program takes the input from the client and validates and then shows a message to the client. Many commands are used to develop the program. In this program, for loop, do-while loop and if…then…else commands are utilized too. In addition, to display the content of the file cat command is utilized and to check if the input is empty or not -z is used, and the read command is used to take the input from the user.

Script

```
#!/bin/bash
Username=$1
ID=$2

entry()
{ if [[ $# -eq 0 ]]
  then
        echo "No arguements supplied."
 elif [[ $# == 2 ]]
 then
        if [[ $Username =~  [a-z|A-Z]+$ ]] #validation of the data entered as input by user
        then
                echo "Username is accepted."
                echo "----------------------------"
                if ! [[ $ID =~ [0-9]+$ || $ID =~ [0.9].[0.9]+$ ]]
                then
                        echo "The ID should contain numbers only."
                        exit
                else
                        echo "ID is accepted"
                        echo "-----------------------------"
                        input #calling input function
                 fi
        else
                echo "The username should contain alphabets only."
                exit
        fi
 else
                echo "<< Please enter two parameters only >>"
                exit
 fi
}

input()
{
        attempt=0
        secretkey=kenzie
        while [ "$attempt" -lt 3 ]
        do
                echo -e "Enter the secret key:-\c"
                read entered_key
                ((attempt++))

                if [ "$entered_key" = "$secretkey" ]
                then
                        echo ""
                        echo "-------------------!!Congratulation!Access granted!!----------------------------"
                        echo ""
                        echo "Welcome, $Username!"
                        echo ""
                        echo "Logged in successful."
                        echo ""
                        echo -e "ID: $ID"
                        echo -e "Name: $Username"
                        echo "Date: `date +%Y-%m-%d`" #displays the date,year,month and day at the process of the login
                        echo "Executed time: `date +%H:%M:%S`" #displays the date,hour,min and second at the process of login
                        guessteam #callin guessteam function
                        break
                fi
                if [ "$attempt" = "3" ] #closes  the program after 3 incorrect entries
                then
                        echo ""
                        echo "You have entered the incorrect secret key 3 times."
                        echo ""
                        echo "---------------Terminating the program------------------"
                        echo
                        exit
                fi
        done

}
guessteam(){
        echo ""
        echo -e "------------------------------------------------------------------------"
        echo -e "|                  Music Band       |          Codes             |"
        echo -e "------------------------------------------------------------------------"
        echo -e "|                  Beatles          |          BEA               |"
        echo -e "|                  AC/DC            |          AD                |"
        echo -e "|                  Queen            |          QUE               |"
        echo -e "|                  Blondie          |          BLO               |"
        echo -e "|                  Nirvana          |          NIR               |"
        echo -e "------------------------------------------------------------------------"
        userguess=false
        until [ $userguess = "true" ]
        do
                echo "Which is the best music band alive?"
                read userinput
                if [ "$userinput" != "BEA" ] && [ "$userinput" != AD ] && [ "$userinput" != "QUE" ] && [ "$userinput" != "BLO" ] && [ "$userinput" != "NIR" ]
                then
                        echo -e "Please enter the correct music band code."
                else
                        if [ "$userinput" == "QUE" ]
                        then
                            echo "Congratulation!You entered the correct music band."
                            echo "Queen is the best music band in the world."
                            member

                        else
                            echo "It seems like your guess is wrong. Please try again!!!."
                        fi
                exit
                fi
        done
}
member(){
        echo ""
        echo "---------------------------------------------------------"
        echo "|          Band member         |        Code       |"
        echo "---------------------------------------------------------"
        echo "|          John Lennon         |        JL         |"
        echo "|          Angus Young         |        AY         |"
        echo "|          Freddie Mercury     |        FM         |"
        echo "|          Debbie Harry        |        DH         |"
        echo "|          Kurt Cobain         |        KC         |"
        echo "---------------------------------------------------------"
        echo "Enter any three member codes from above menu.\n[ALL IN BLOCK LETTERS ONLY]"
        read membercode1 membercode2 membercode3 membercode4
        members=($membercode1 $membercode2 $membercode3)
        if [ ${#members[@]} -eq 3 ]
        then
                if [[ "$membercode1" = "$membercode2" || "$membercode1" = "$membercode3" || "$membercode2" = "$membercode3" ]]
                then
                        echo ""
                        echo "You have entered the same member code several times. Try again!!!"
                        echo ""
                        member #calling member function
                else
```

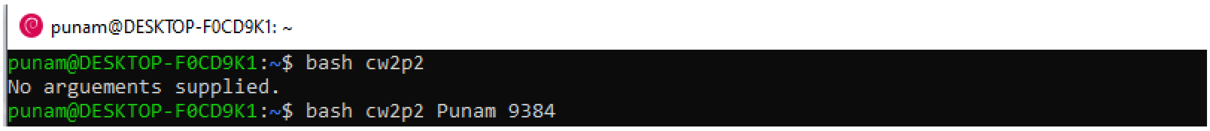
```
            PS3="Choose your option from above menu.[ALL IN BLOCK LETTER ONLY]"
            echo ""
            select member in $membercode1 $membercode2 $membercode3
            do
                case $member in #calls memberInfo funtion and pass the variable
                    $membercode1)
                    memberInfo "$membercode1"
                    break;;

                    $membercode2)
                    memberInfo "$membercode2"
                    break;;

                    $membercode3)
                    memberInfo "$membercode3"
                    break;;
                esac
            done
        fi
    else

        echo ""
        echo "---------------Incorrect option. Please try again------------"
        echo ""
        echo "Please enter three band member codes only!"
        echo ""
        member

    fi
}
memberInfo(){
    Name=$1
    if [[ $Name == "JL" || $Name == "AY" || $Name == "KC" ]]
    then
        if [[ $Name == "JL" && -f fileJL && -r fileJL ]] #displays details for JL file
        then
            echo "--------------------------"
            echo "You've choosen John Lennon"
            echo ""
            cat fileJL
            echo ""
            loop
        elif [[ $Name == "AY" && -f fileAY && -r fileAY ]] #displays details for AY file
        then
            echo "--------------------------"
            echo "You've choosen Angus Young"
            echo ""
            cat fileAY
            echo ""
            loop
        elif [[ $Name == "KC" && -f fileKC && -r fileKC ]] #displays details for KC file
        then
            echo "--------------------------"
            echo "You've choosen Kurt Cobain"
            echo ""
            cat fileKC
            echo ""
            loop
        else
            echo "The choice is not available"
        fi
    else
        echo ""
        echo "Sorry!! The file for the member $Name cannot be accessed."
        echo ""
        echo "You choose the wrong member name. Please try again."

    fi
}
loop(){
    echo -e "Do you want to start again?(Y/N)"
    read input
    if [[ "$input" == "Y" ||  "$input" == "y" ]]
    then
        guessteam
    elif [[ "$input" == "N" || "$input" == "n" ]]
    then
        echo -e "Thankyou!!"
        exit


    else
        echo -e "Please enter valid input(Y/N)"
        loop
    fi
}
entry $1 $2
```

Testing

Test 1: Run without a username

Table 1:Run without username

Objective	To check if the program runs without the username
Expected Output	The error message is displayed because we haven't entered any parameter
Actual Output	The message says no argument supplied.
Conclusion	Running without the username will not execute the program further.



punam@DESKTOP-F0CD9K1: ~

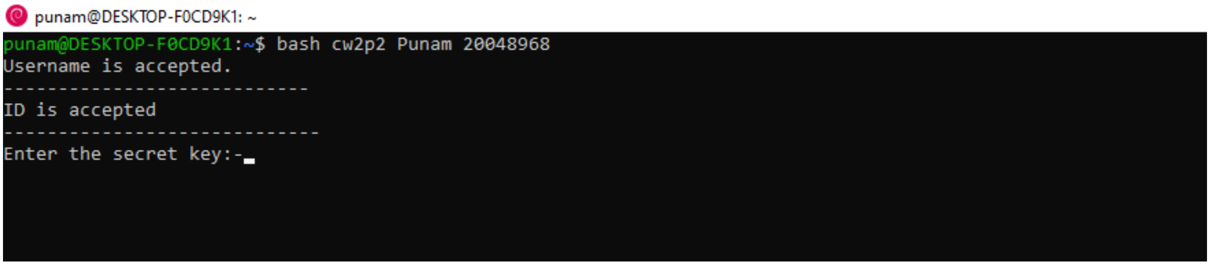
```
punam@DESKTOP-F0CD9K1:~$ bash cw2p2
No arguements supplied.
punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 9384
```

Figure 1: Run without username

Test 2: Run with username and ID

Table 2: Run with username and ID

Objectives	To run the program with the correct username and id.
Expected Output	The program should run without any error or interruption.
Actual Output	The program does not show any error and is continued to the next step which then asks user to enter his/her secret key.
Conclusion	The system does not show any error after running with username and id.



punam@DESKTOP-F0CD9K1: ~

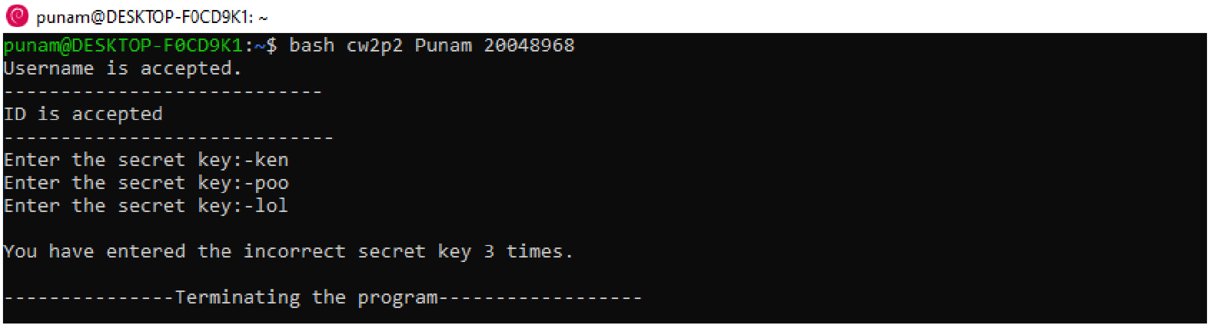
```
punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 20048968
Username is accepted.
---------------------------
ID is accepted
----------------------------
Enter the secret key:-
```

Figure 2:Run with username and ID

Test 3: Run incorrect password 3 times

Table 3: Run incorrect password three times

Objectives	To run the program with an incorrect password 3 times.
Expected Output	The error message should be displayed, and the program should be terminated.
Actual Output	The program was terminated after the input of the password 3 times in a row.
Conclusion	The program cannot be run after running an incorrect password three times.



punam@DESKTOP-F0CD9K1: ~

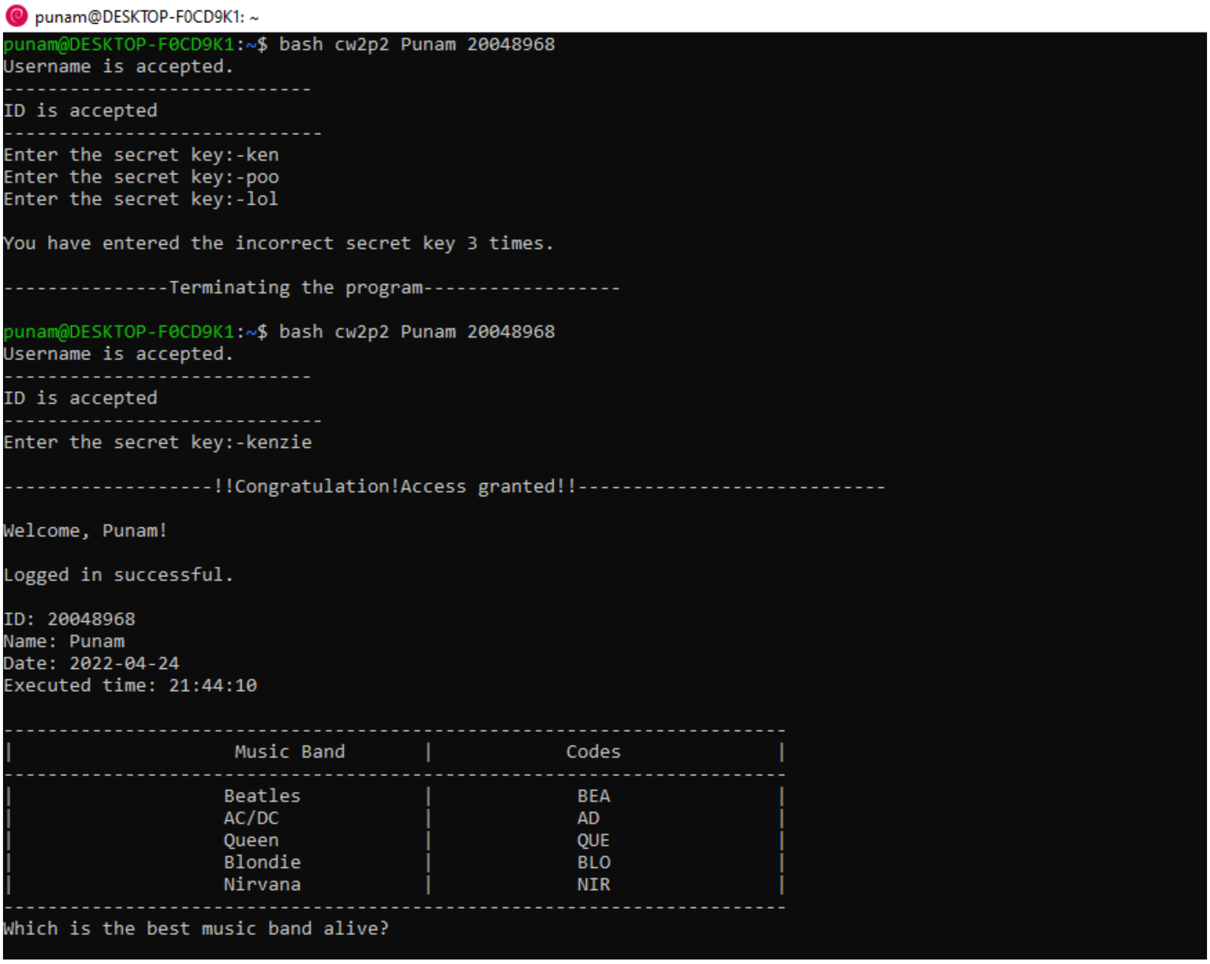
```
punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 20048968
Username is accepted.
----------------------------
ID is accepted
-----------------------------
Enter the secret key:-ken
Enter the secret key:-poo
Enter the secret key:-lol

You have entered the incorrect secret key 3 times.

---------------Terminating the program------------------
```

Figure 3:Run incorrect password 3 times

Test 4: Run the correct password

Table 4: Run the correct password

Objectives	To run the program with the correct password.
Expected output	The program should run smoothly after entering the correct password.
Actual Output	The program was run without interruption after it was run with the correct password.
Conclusion	Running with the correct password does not show any interruption in the program.



punam@DESKTOP-F0CD9K1: ~

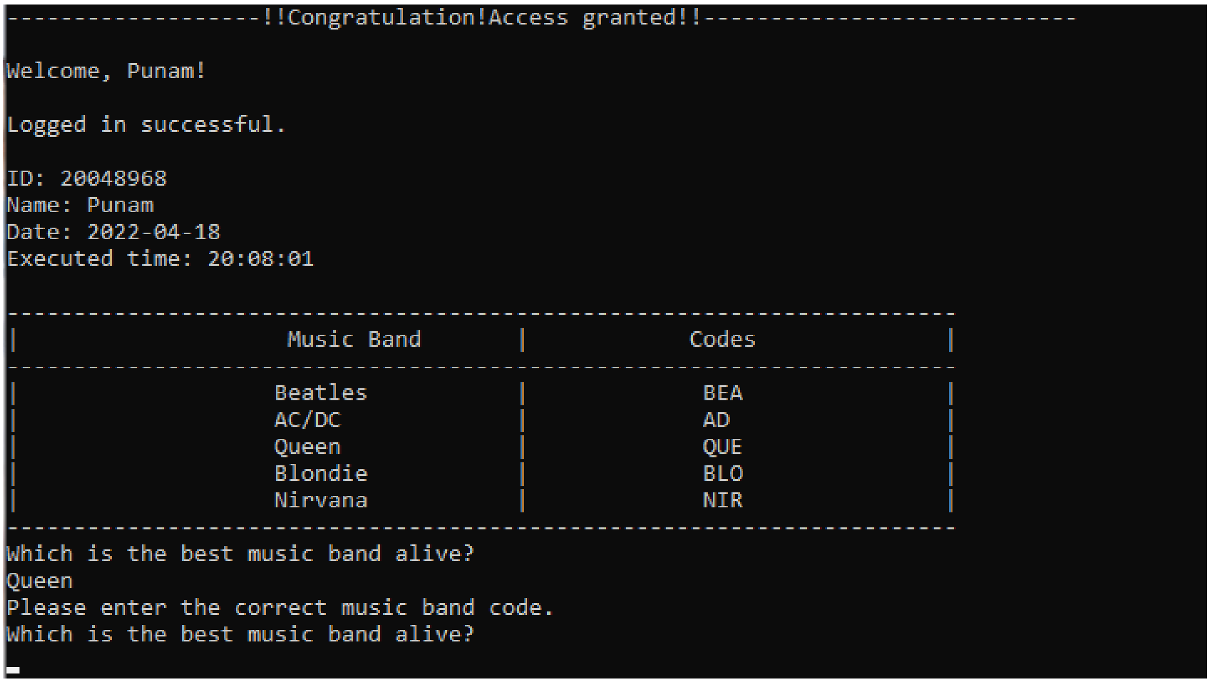
```
punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 20048968
Username is accepted.
----------------------------
ID is accepted
-----------------------------
Enter the secret key:-ken
Enter the secret key:-poo
Enter the secret key:-lol

You have entered the incorrect secret key 3 times.

---------------Terminating the program------------------

punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 20048968
Username is accepted.
----------------------------
ID is accepted
-----------------------------
Enter the secret key:-kenzie

-------------------!!Congratulation!Access granted!!----------------------------

Welcome, Punam!

Logged in successful.

ID: 20048968
Name: Punam
Date: 2022-04-24
Executed time: 21:44:10

-----------------------------------------------------------------------
|                    Music Band       |             Codes              |
-----------------------------------------------------------------------
|                   Beatles           |              BEA               |
|                   AC/DC             |              AD                |
|                   Queen             |              QUE               |
|                   Blondie           |              BLO               |
|                   Nirvana           |              NIR               |
-----------------------------------------------------------------------
Which is the best music band alive?
```

Figure 4: Run with correct password

Test 5: Band name

Table 5: Band name

Objectives	To check the output of the program after entering the band name instead of the band code.
Expected Output	The error message should be displayed after entering the band name.
Actual Output	The error message was shown saying, ‘Please enter the correct music band.’
Conclusion	The program does not execute after entering the band name instead of the band code and is asked to choose the band code again.



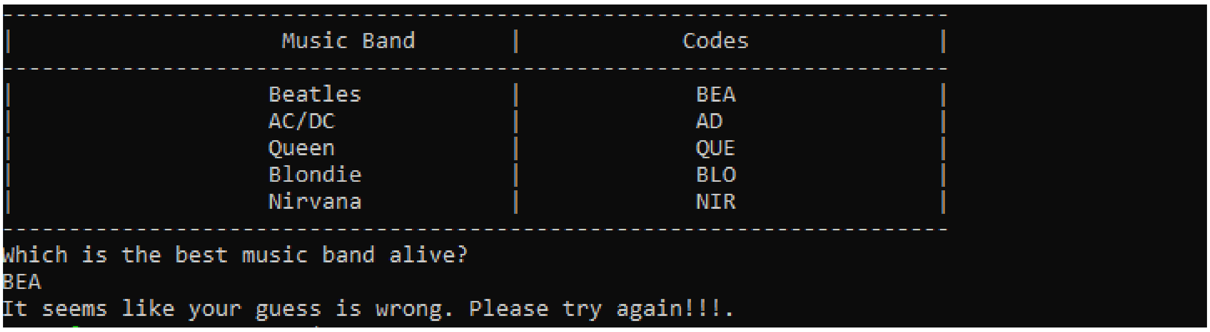
```
-------------------!!Congratulation!Access granted!!----------------------------

Welcome, Punam!

Logged in successful.

ID: 20048968
Name: Punam
Date: 2022-04-18
Executed time: 20:08:01

 ----------------------------------------------------------------------
|                    Music Band       |            Codes              |
 ----------------------------------------------------------------------
|                   Beatles           |             BEA               |
|                   AC/DC             |             AD                |
|                   Queen             |             QUE               |
|                   Blondie           |             BLO               |
|                   Nirvana           |             NIR               |
 ----------------------------------------------------------------------
Which is the best music band alive?
Queen
Please enter the correct music band code.
Which is the best music band alive?
_
```

Figure 5: Band name

Test 6: Incorrect Band code

Table 6: Incorrect Band code

Objectives	To run the program with incorrect band code
Expected Output	The program should display an error message and should not run further.
Actual Output	An error message was displayed saying, 'It seems like your guess is wrong. Please try again!!'
Conclusion	The program did not execute further after entering the wrong band code.



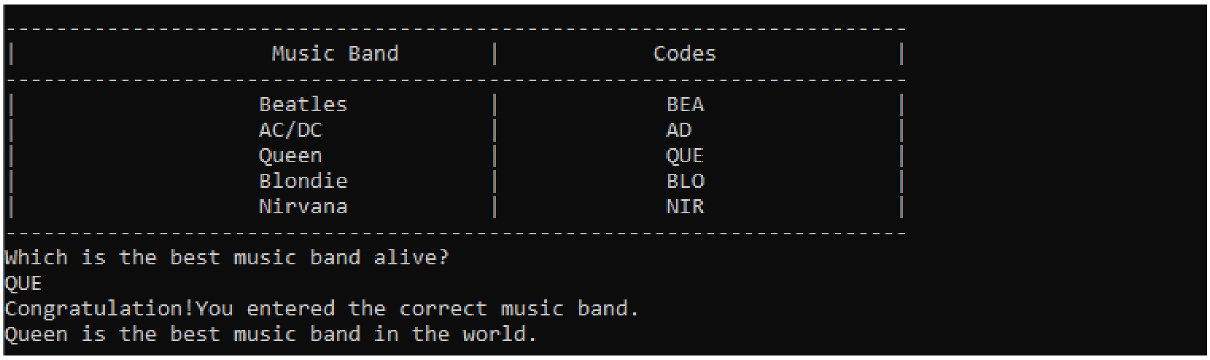
```
----------------------------------------------------------------------
|                    Music Band        |           Codes            |
----------------------------------------------------------------------
|                   Beatles            |            BEA             |
|                   AC/DC              |            AD              |
|                   Queen              |            QUE             |
|                   Blondie            |            BLO             |
|                   Nirvana            |            NIR             |
----------------------------------------------------------------------
Which is the best music band alive?
BEA
It seems like your guess is wrong. Please try again!!!.
```

Figure 6: Incorrect Band Code

Test 7: Correct Band Code

Table 7:Correct Band Code

Objectives	To run the program with the correct band code.
Expected Output	The program should proceed to the next step after accepting the correct band code.
Actual Output	The program asked the user to choose three band members after running with the correct band code.
Conclusion	When the correct band code is given, the program executes the next step.



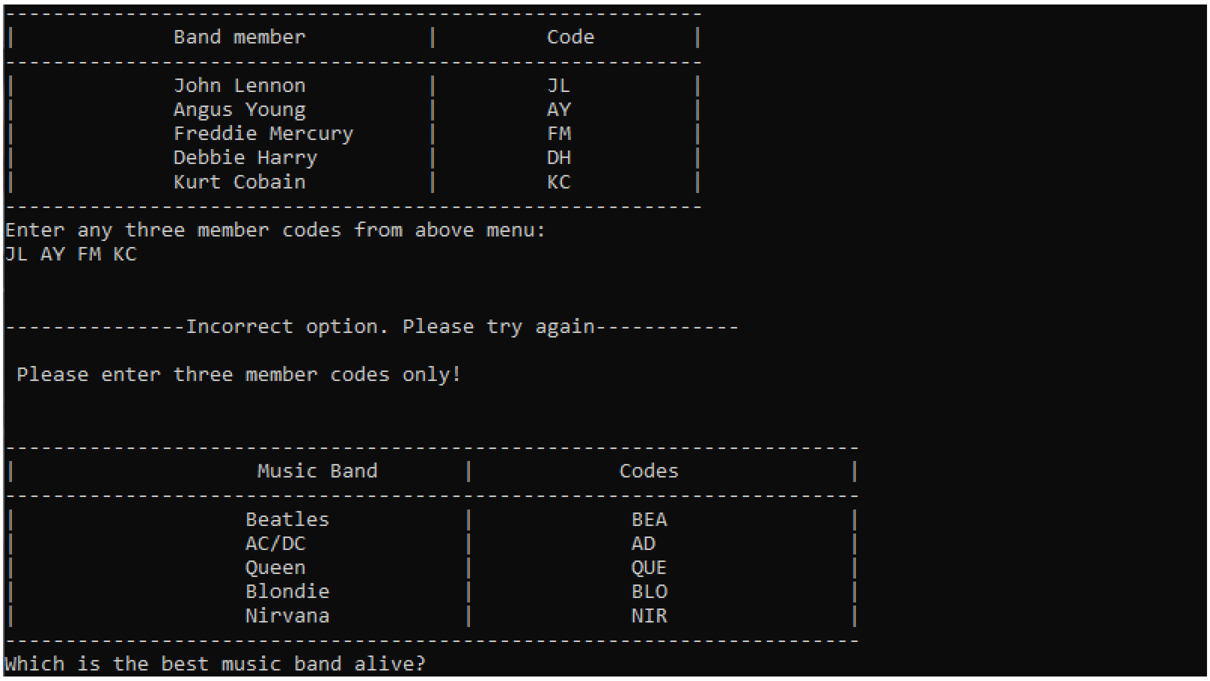
```
----------------------------------------------------------------------
|                    Music Band        |           Codes            |
----------------------------------------------------------------------
|                   Beatles            |            BEA             |
|                   AC/DC              |            AD              |
|                   Queen              |            QUE             |
|                   Blondie            |            BLO             |
|                   Nirvana            |            NIR             |
----------------------------------------------------------------------
Which is the best music band alive?
QUE
Congratulation!You entered the correct music band.
Queen is the best music band in the world.
```

Figure 7: Correct Band Code

Test 8: Pick four member's names

Table 8: Pick four member's names

Objectives	To run the program by picking four members' names.
Expected Output	The program should not run further and should display the error message.
Actual Output	The program displayed an error saying, 'incorrect option, please enter three-member codes only!'
Conclusion	The program cannot run further after selecting four members' names during the selection.



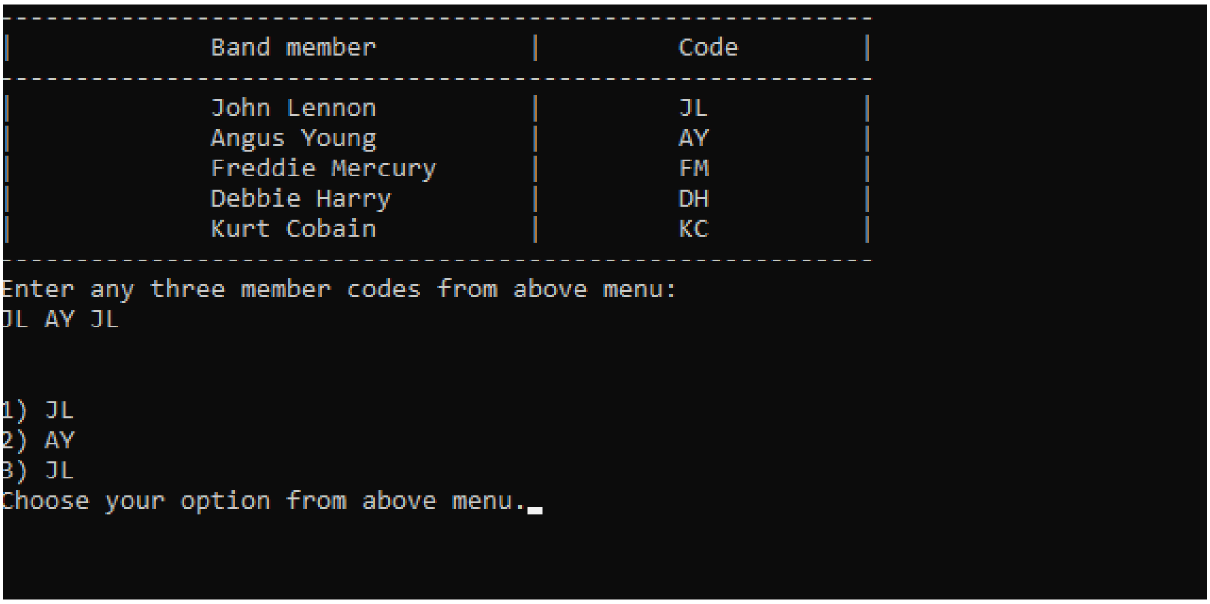
```
---------------------------------------------------------
|             Band member          |         Code        |
---------------------------------------------------------
|             John Lennon          |         JL          |
|             Angus Young          |         AY          |
|             Freddie Mercury      |         FM          |
|             Debbie Harry         |         DH          |
|             Kurt Cobain          |         KC          |
---------------------------------------------------------
Enter any three member codes from above menu:
JL AY FM KC


---------------Incorrect option. Please try again------------

 Please enter three member codes only!


----------------------------------------------------------------------
|                   Music Band       |               Codes            |
----------------------------------------------------------------------
|                  Beatles           |               BEA              |
|                  AC/DC             |               AD               |
|                  Queen             |               QUE              |
|                  Blondie           |               BLO              |
|                  Nirvana           |               NIR              |
----------------------------------------------------------------------
Which is the best music band alive?
```

Figure 8: Pick 4 member's name

Test 9: Pick the same member's name

Table 9: Pick the same member's name

Objectives	To run the program after picking the same member's name.
Expected Output	The program should display an error message saying selecting the same member's name is invalid.
Actual Output	The program does not show any error message and the program runs smoothly.
Conclusion	Picking the same member's name is accepted.



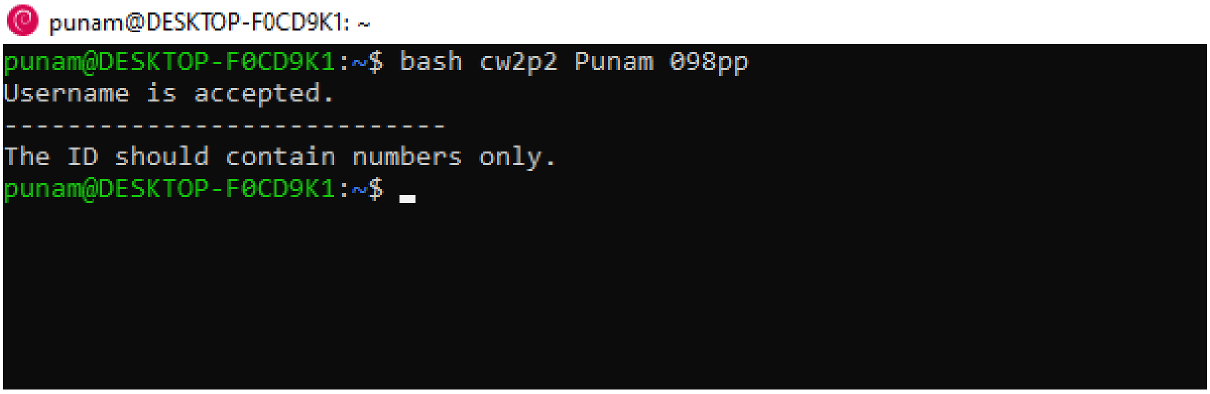
```
----------------------------------------------------------
|              Band member          |        Code        |
----------------------------------------------------------
|              John Lennon          |        JL          |
|              Angus Young          |        AY          |
|              Freddie Mercury      |        FM          |
|              Debbie Harry         |        DH          |
|              Kurt Cobain          |        KC          |
----------------------------------------------------------
Enter any three member codes from above menu:
JL AY JL


1) JL
2) AY
3) JL
Choose your option from above menu.
```

Figure 9: Pick the same member's name

Test 10: Pick the wrong user Id. (parameter verification)

Table 10: Pick the wrong user Id. (Parameter verification)

Objectives	To run the program after giving the wrong user Id.
Expected Output	The program should display an error message after giving the wrong user id.
Actual Output	The program showed an error message saying, 'The ID should contain numbers only.'
Conclusion	The program does not run after taking the wrong user id from the users.



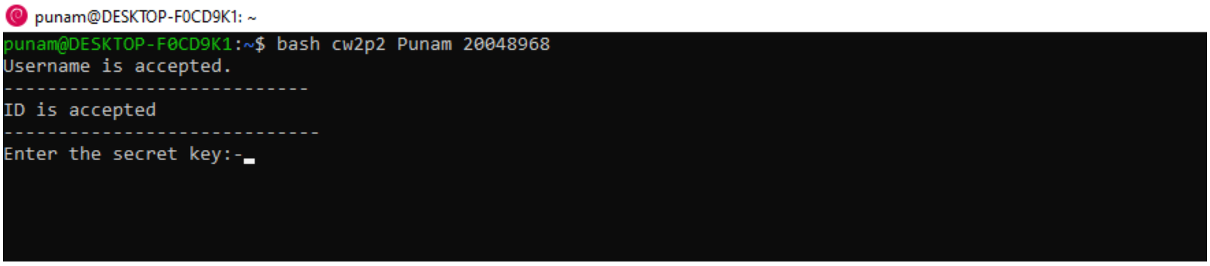
```
punam@DESKTOP-F0CD9K1: ~
punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 098pp
Username is accepted.
----------------------------
The ID should contain numbers only.
punam@DESKTOP-F0CD9K1:~$
```

Figure 10: Pick the wrong user Id. (Parameter verification)

Test 11: Right user ID.

Table 11: Right user ID

Objectives	To run the program after giving the right user id.
Expected Output	The program should run without any error message after taking the right user id.
Actual Output	The program accepts the user id and was run to the next step.
Conclusion	The program runs smoothly after taking the right user id from the user.



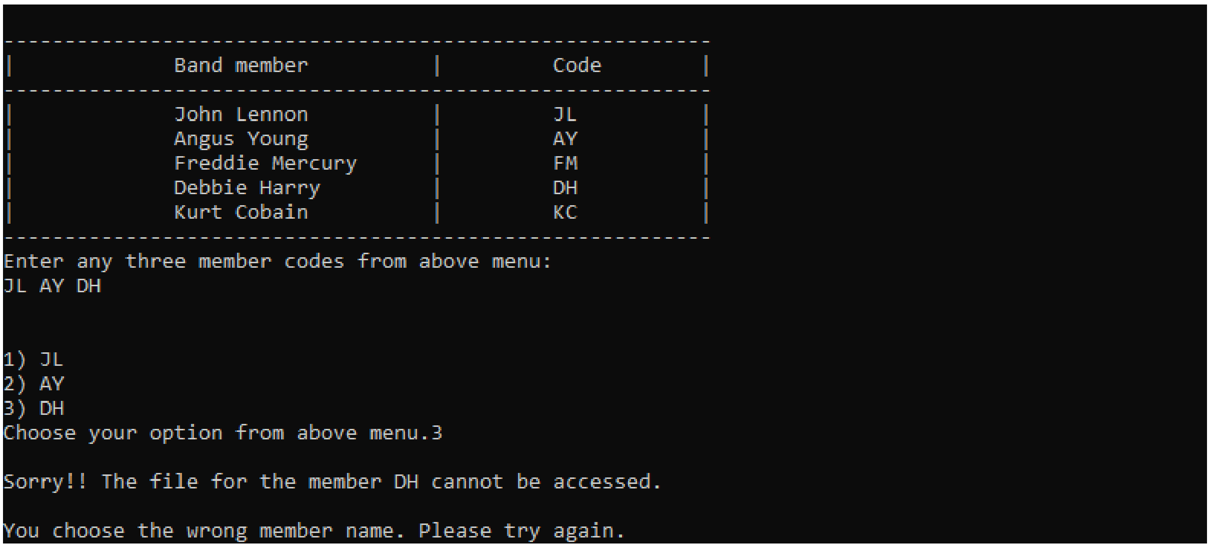
```
punam@DESKTOP-F0CD9K1: ~
punam@DESKTOP-F0CD9K1:~$ bash cw2p2 Punam 20048968
Username is accepted.
----------------------------
ID is accepted
-----------------------------
Enter the secret key:-
```

Figure 11: Right user Id

Test 12: No external file of member (except 3 profile band members that you have made or) or invalid band member ID.

Table 12: No external file of member (except 3 profile band members that you have made or) or invalid band member ID.

Objectives	To check the output after entering the invalid band code.
Expected Output	There should display an error message.
Actual Output	The error message was shown saying, 'The file for the band member cannot be accessed.'
Conclusion	The program shows an error message if the file with no external files or invalid band member ID is chosen.



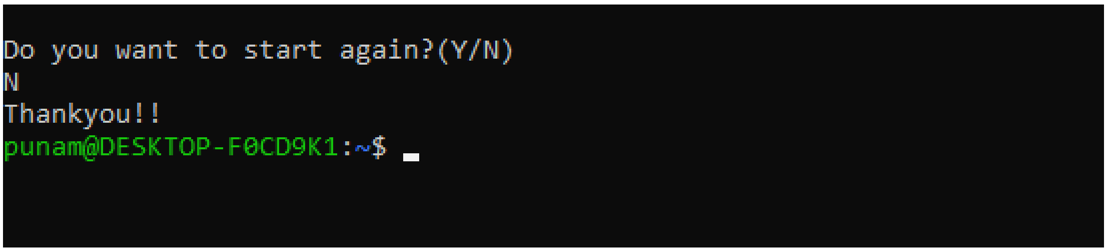
```
----------------------------------------------------------
|             Band member          |         Code        |
----------------------------------------------------------
|             John Lennon          |         JL          |
|             Angus Young          |         AY          |
|             Freddie Mercury      |         FM          |
|             Debbie Harry         |         DH          |
|             Kurt Cobain          |         KC          |
----------------------------------------------------------
Enter any three member codes from above menu:
JL AY DH


1) JL
2) AY
3) DH
Choose your option from above menu.3

Sorry!! The file for the member DH cannot be accessed.

You choose the wrong member name. Please try again.
```

Figure 12: No external file of member

Test 13: EXIT YES

Table 13: Exit yes

Objectives	To check the output after typing yes while exiting the program.
Expected Output	The program should terminate the program.
Actual Output	The program was terminated.
Conclusion	The program terminates after taking yes as an answer,



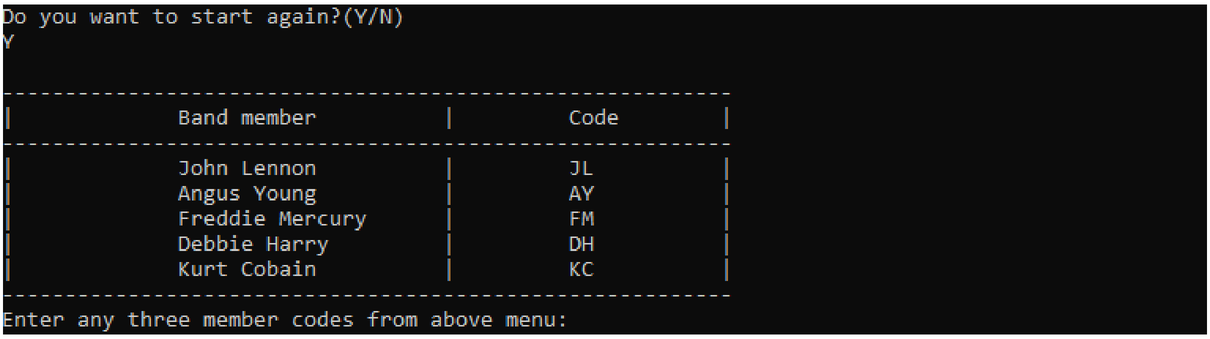
```
Do you want to start again?(Y/N)
N
Thankyou!!
punam@DESKTOP-F0CD9K1:~$ 
```

Figure 13:Exit yes

Test 14: EXIT NO

Table 14: Exit no

Objectives	To check the output after taking no while exiting the program.
Expected Output	The program should re-execute the program.
Actual Output	The program was re-executed.
Conclusion	The program restarts after taking no as an answer.



```
Do you want to start again?(Y/N)
Y

---------------------------------------------------------
|             Band member          |         Code        |
---------------------------------------------------------
|             John Lennon          |         JL          |
|             Angus Young          |         AY          |
|             Freddie Mercury      |         FM          |
|             Debbie Harry         |         DH          |
|             Kurt Cobain          |         KC          |
---------------------------------------------------------
Enter any three member codes from above menu:
```

Figure 14:Exit no

Contents of three files: (TEXTS)

- **JL:**

  John Winston Ono Lennon (born John Winston Lennon; 9 October 1940 -8 December 1980) was an English singer, songwriter, musician, and peace activist who achieve worldwide fame as the founder, co-songwriter, and co-lead vocalist and rhythm guitarist of the Beatles. Lennon was characterized by the rebellious nature and acerbic wit in his music, writing, and drawings, on film, and in interviews. His song writing partnership with Paul McCartney remain the most successful in history.

- **AY:**

  Angus McKinnon Young is an Australian musician, best known as the co-founder, lead guitarist, songwriter, and sole constant original member of the Australian hard rock band AC/DC. He is known for his energetic performances, schoolboy-uniform stage outfits, and his own version of Chuck Berry's duckwalk. Young was ranked 24th in Rolling Stone magazine's 100 greatest guitarists of all-time list. In 2003, Young and other members of AC/DC were inducted into the Rock and Roll Hall of Fame.

- **KC:**

  Kurt Donald Cobain (February 20, 1967 - April 5, 1994) was an American singer, songwriter, and artist. He was the guitarist, lead vocalist, and primary songwriter of the rock band Nirvana. Through his angst-fueled song writing and establishment persona, Cobain’s compositions widened the thematic conventions of mainstream rock. He was heralded as a spokesman of Generation X and is considered one of the most influential musicians in the history of alternative rock. In addition to "Smell Like Teen Spirit", "Come as You Are", "Lithium", "In Bloom", "Something in the Way", "Heart-Shaped Box", "All Apologies", "About a Girl", "Aneurysm", and "You know You're Right".

Conclusion

To whole up, this project given for coursework 2 was to make a program in the UNIX OS. Bash shell was used while developing the coursework. The program was developed and tested successfully. While doing this coursework, I needed to confront the challenges particularly in the coding segment as numerous mistakes occurred. I was likewise somewhat confused in regard to the syntax. Yet to conquer the confusion and troubles, a lot of exploration was finished with respect to the applicable syntax. Regular interaction with instructors, consistent exertion, and a lot of exploration, going through the lecture slides and surfing the web-assisted with acquiring sound information about UNIX OS. Moreover, this coursework helps me to boost my knowledge about Linux commands. I got the chance to learn how to create the bash file and work on it to develop the user interface program. It was a decent encounter to foster a program utilizing UNIX OS and it was enjoyable to chip away at this project as well.

Task B

Introduction

Process management involves the execution of different tasks like the creation of processes, scheduling of processes, termination of processes, and the management of deadlock. It is the liability of the operating system to deal with all the running processes of the system. The process is a program that is under execution, which is an important part of modern-day operating systems. The OS must allocate resources that enable processes to share and exchange information. It also protects the resources of each process from other methods and allows synchronization among processes. It is the job of the OS to manage all the running processes of the system. It handles operations by performing tasks like process scheduling and such as resource allocation. (Williams, 2022)

A request to the OS is typically in the form of a system call, (for example, a call from the running process to a function that uses part of the OS code). For instance, requesting a document from disk or saving a part of code or information from memory to a record on disk.

Aims and Objectives

The aim of task B is to compose a report on the given theme ‘Process Management’ which incorporates various topics such as process architecture, process control blocks, process states, process hierarchies, and process implementation. In like manner, the principle point of this report is to give detail information about the process management, its responsibilities, and background.

The objectives of this report are:

- Acquiring information on the given topic.
- Research and data gathering.
- Neutral presentation of data.

Background

Process Architecture

The hierarchical design of processes and systems that are utilized to change inputs into outputs is alluded to as process architecture. The expression can be applied to an assortment of things, including computing, business processes, and project management, to refer to a couple. It can be used to describe any process or set of operations. We can see the reason why this term is utilized in countless settings since we have seen that significant piece of importance. There are many advantages to characterizing and examining process architecture. They are:

- This works with rearrangements.
- It speeds up response time.
- Aids in cost-cutting.
- Recognizing automation opportunities
- Helps with arranging and anticipating the impacts of system changes.

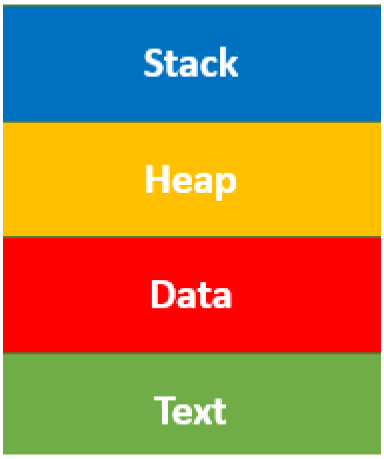


Figure 15: Process architecture (williams, 2022)

- **Stack:**

  The Stack stores temporary data like function parameters, returns addresses, and local variables.

- **Heap:**

  Heap allocates memory, which may be processed during its run time.

- **Data:**

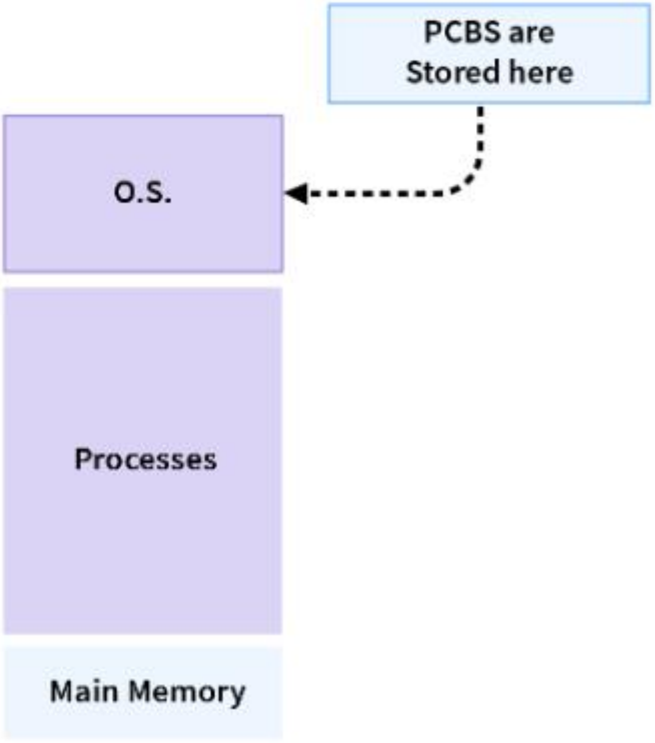
  Data contains the variable.

- **Text:**

  The text section includes the current activity, which is represented by the value of the program counter.
  (Silwal, 2022)

Process Control Blocks

Whenever the process is made by the operating system it makes a data structure to store the data of a process, known as a process control block (PCB).



(Vaisnav, 2022)

PCBs are put away in uniquely held memory for the operating system known as kernel space. Besides, PCB is extraordinary for each process which comprises different attributes such as process ID, priority, registers, process counters, process states, list of open files, etc. The role of the process control block emerges as an ID card of each process. The operating system does not realize which process is which, until the operating system alludes through the PCB of each process. For instance, there are MS word processes, pdf processes, printing processes, and many background processes running as of now on the CPU.

Structure of Process Control Block

The process control block (PCBs) contains many attributes such as process state, program counter, CPU registers, CPU scheduling information, accounting, and business information, memory-management information, I/O status information, etc for each process.

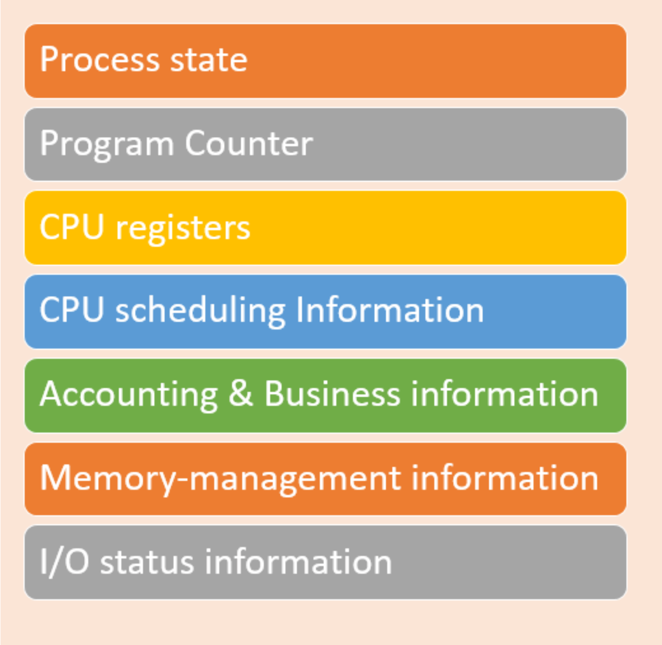

Process state
Program Counter
CPU registers
CPU scheduling Information
Accounting & Business information
Memory-management information
I/O status information

Figure 16: Process Control Block (PCB) (Silwal, 2022)

The description of each field of the process control block is given below:

1. **Process State:**

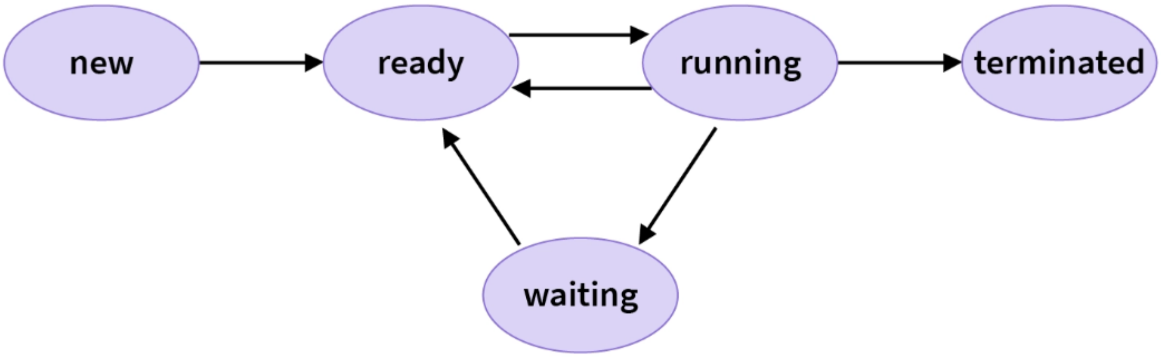
   Although each process is an independent entity, with its program counter registers, stack, open files, alarms, and other internal states, processes often need to interact, communicate, and synchronize with other processes. One process may create some result that another process utilizes as contribution or input, for instance, all thing considered, the data needs to be moved between processes.

   In the shell command; cat Tanjiro Inosuke Zenitsu | grep tree

   The first process, running cat, concatenates and outputs three files. Likewise, the second process, running grep, selects all lines containing the word “tree.”

Depending on the relative speeds of the two processes (which depends on both the relative complexity of the programs and how much CPU time each one has had), it may happen that grep is ready to run, but there is no input waiting for it. It must then block until some input is available. When a process blocks, it does so because logically it cannot continue, typically because it is waiting for input that is not yet available. It is also possible for a process that is conceptually ready and able to run to be stopped because the operating system has decided to allocate the CPU to another process for a while. (Woodhull, 2006)

Generally, a process may be present in one of the five states during its execution:



- **New:** This state contains the processes which are ready to be stacked by the operating system into the main memory.

- **Ready:** This state contains the process which is both ready to be executed and is currently in the main memory of the system. The operating system brings the processes from secondary memory (hard disk) to main memory (RAM). As these processes are present in the main memory and are waiting to be assigned to the CPU, the state of these processes is known as the ready state. (Vaisnav, 2022)

- **Running:** This state contains the processes which are currently executed by the CPU in our system. In that even that there is an absolute x CPU in our system, then a most extreme number of running processes for a specific time is also x.

- **Waiting:** A process from its running state may transition to a block or wait for state-contingent upon the scheduling algorithm or on account of the internal conduct of the process (process explicitly needs to pause).

- **Termination:** A process that finishes its execution comes to its termination state. Every one of the contents in that process (Process control block) will likewise be erased by the operating system.

2. **Program Counter:**

The program counter shows the location of the next instruction to be executed for the process.

3. **CPU registers**

The CPU registers vary in number and type, depending on the computer architecture. It is a quickly accessible small-sized location available to the CPU. Furthermore, these registers are stored in virtual memory (RAM) and like CPU registers include accumulators, index registers, stack pointers, and general-purpose registers, plus any condition code information. Along with the program counter, this state information must be saved when an interrupt occurs, to allow the process to be continued correctly afterward. (Masood, 2016)

4. **CPU-scheduling information**

The CPU-scheduling information incorporates a process priority, pointers to scheduling queues, and some other scheduling parameters.

5. Accounting and Business Information

The accounting and business information incorporates the amount of CPU and real-time used, time limits, account numbers, job or process numbers, and so on,

6. Memory-management information

Memory-management information may include such data, for example, the worth of the base and limit registers, the page tables, or the section tables, contingent upon the memory system utilized by the operating system.

7. I/O status information

The I/O status information includes the rundown of I/O devices dispensed to the process, a list of open files, etc.

Process States

The process state is already described in the above section of Process control block.

Continue reading…..

Process Hierarchies

In some systems, when an interaction makes another process, the parent and child keep on being related in some ways. The child can itself generate more processes, building a process hierarchy. Not at all like plants and animals that use sexual reproduction, a process has just a single parent (but zero, one, two, or more children). In a process, its children, and further descendants together may form a process group. At the point when a user conveys a message from the console, the signal may be conveyed to all individuals from the process group as of now connected with the keyboard (typically all processes that were made in the ongoing window). This is what we call signal-dependent. Assuming that a signal is shipped to off a group, each process can get the sign, disregard the sign, or make the default move, which is to be killed by the signal.

As a simple example of how process trees are used. Two special processes, the reincarnation server and init are present in the boot image. The reincarnation server's job is to restart drivers and servers. It begins by blocking, waiting for a message telling it what to create. In contrast, init executes the /etc/rc script that causes it to issue commands to the reincarnation server to start the drivers and servers not present in the boot image. This procedure makes the drivers and servers so started children of the reincarnation server, so if any of them ever terminate, the reincarnation server will be informed and can restart (i.e. reincarnate) them again. When init has finished this, it reads a configuration file /etc/ttytab) to see which terminals and virtual terminals exist. Init forks a getty process for each one, displays a login prompt on it, and then waits for input. When a name is typed, getty execs a login process with the name as its argument. If the user succeeds in logging in, login will exec the user's shell. So, the shell is a child of init. User commands create children of the shell, which are grandchildren of init. This sequence of events is an example of how process trees are used. (Woodhull, 2006)

Implementation of Process

To carry out the process model, the operating system maintains a table (an array of structures), called the process table, with one entry per process. (A few writers call these entries process control blocks.) This entry contains data about the process state, its program counter, stack pointer, memory allocation, the status of its open files, its accounting and scheduling information, alarms, and other signals, and all the other things about the process that must be saved when the process is changed from running to ready state so that it can be restarted later as though it has never been halted.

The table given below shows some of the important fields in a typical system.

Process Management	Memory Management	File Management
Registers	Pointer to text segment	Root directory
Program counter	Pointer to the data segment	Working directory
Program status word	Pointer to stack segment	File descriptors
Stack pointer		User Identity
Process state		Group Identity
Priority		
Scheduling parameters		
Process Identity		
Parent process		
Process group		
Signals		
The time when the process started		
Central Processing Unit time used		
Central Processing Unit time of Children		
Time of next alarm		

(Codescracker, 2022)

The fields in segment one connected with process management, the fields in section two connected with memory management, and the fields in section three connected with file management.

Conclusion

From this task, a ton of information on Process Management was gained. This errand helps to boost the knowledge of process management in an operating system. For instance, what does the process management truly concerns about? All things considered, through research, I get to know that it concerns creating processes, satisfying the resource requirements, scheduling them for utilization of a CPU, implementing less synchronization to control their collaboration, avoid deadlock so that they do not sit tight for one another endlessly and lastly process management concerns about terminating them when they complete their operation. The research was done regarding some of the process management fields (the Process States, Process Scheduling, and Priority Scheduling) which helped a lot to achieve the tasks. I likewise, became more acquainted with the different fields and the way it works in the computer, I knew about various terms alongside their capabilities.

References

Codescracker, 2022. *OS Process Implementation.* [Online] Available at: https://codescracker.com/operating-system/process-implementation.htm#:~:text=OS%20Process%20Implementation.%20To%20implement%20the%20process%20model%2C,open%20files%2C%20its%20accounting%20and%20the%20scheduling%20
[Accessed 14 04 2022].

Data-flair, 2022. *Process Scheduling in Operating System.* [Online] Available at: https://data-flair.training
[Accessed 15 April 2022].

JavaTpoint, 2022. *Priority Scheduling.* [Online] Available at: https://www.javatpoint.com
[Accessed 04 April 2022].

Masood, S., 2016. *Process Control Block.* [Online] Available at: https://sajidullah786.wordpress.com/2016/03/11/process-control-block/
[Accessed 13 04 2022].

Silwal, D., 2022. *Process Control Block.* Kathmandu: Islington College.

tutorialspoint, 2022. *Operating System - Process Scheduling.* [Online] Available at: https://www.tutorialspoint.com
[Accessed 15 April 2022].

tutorialspoint, 2022. *What is Unix ?.* [Online] Available at: https://www.tutorialspoint.com/unix/unix-getting-started.htm#:~:text=The%20Unix%20operating%20system%20is%20a%20set%20of,kernel%20through%20a%20program%20known%20as%20the%20shell.
[Accessed 03 April 2022].

Vaisnav, N., 2022. *What is Process Control Block in OS?.* [Online] Available at: https://www.scaler.com/topics/operating-system/process-control-block-in-os/
[Accessed 12 04 2022].

williams, L., 2022. *Process Management in OS: PCB in Operating System.* [Online] Available at: https://www.guru99.com/process-management-pcb.html [Accessed 09 April 2022].

Williams, L., 2022. *Process Management in OS: PCB in Operating System.* [Online] Available at: https://www.guru99.com/process-management-pcb.html [Accessed 05 April 2022].

Woodhull, A. S. T. a. A. S., 2006. Process Hierarchies. In: A. S. Andrew S.Tanenbaum, ed. *Operating Systems-Design and Implementation.* Upper Saddie River,New Jersey 07458,USA: PearsonEducation,inc.,publishing as Pearson Prentice Hall, p. 1054.

Woodhull, A. S. T. a. A. S., 2006. Process States. In: A. S. W. Adrew S. Tanenbaum, ed. *Operating Systems- Design and Implementation.* Upper Saddle River, New Jersey 07458,USA: PearsonEducation, Inc.,publising as Pearson Prentice Hall, p. 1054.

Appendix

Appendix -A (Glossary)

This is a glossary section where some meanings of words are explained in detail below:

- Synchronization: The operation or activity of two or more things at the same time or rate.
- Reincarnation: a new version of something from the past.
- Descendants: a machine, artifact, system, etc., that has developed from an earlier, more rudimentary version.
- Dispatched: send off to a destination or for a purpose.

As there were no such difficult words, Appendix A is quite brief

Appendix -B (Process Scheduling)

Process scheduling is the activity of the process manager that handles the expulsion of the running process from the CPU and the determination of another process based on a specific methodology. Furthermore, process scheduling is a fundamental piece of a Multiprogramming operating system. Such operating systems permit more than one process to be stacked into the executable memory at a time and the stacked process shares the CPU utilizing time multiplexing.

There are various states a process must go through during execution. The OS keeps a different queue for each state alongside the process control blocks (PCB) for all processes. The PCB moves to a new state queue, in the wake of being unlinked from its ongoing queue when the state of a process changes.

The Operating System maintains the following important process scheduling queues:

1. Job queue:

   This queue makes sure that processes stay in the system.

2. Ready queue:

   This queue stores a set of all processes in the main memory, ready and waiting for execution. The ready queue stores any new process.

3. Device queue:

   This queue consists of the processes blocked due to the unavailability of an I/O device.

(Data-flair, 2022)

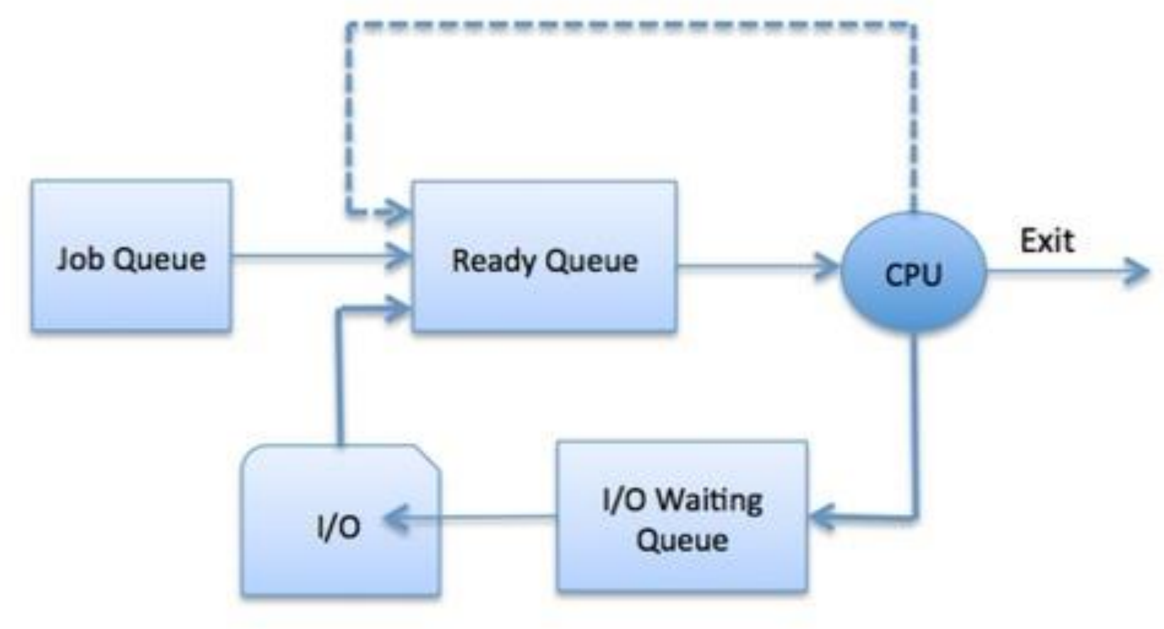


Figure 17:Process scheduling (tutorialspoint, 2022)

The stages a process goes through are:

- A new process first goes in the Ready queue, where it waits for execution or to be dispatched.
- The CPU gets allocated to one of the processes for execution.
- The process issues an I/O request, after which an OS places it in the I/O queue.
- The process then creates a new subprocess and waits for its termination.
- If removed forcefully, the process creates an interrupt. Thus, once this interrupt completes, the process goes back to the ready queue.
  (Data-flair, 2022)

Following are the objectives of process scheduling:

1. It maximizes the number of interactive users within acceptable response times.
2. It achieves a balance between response and utilization.
3. It makes sure that there is no postponement for an unknown time and enforces priorities.
4. It gives reference to the processes holding the key resources.
   (tutorialspoint, 2022)

Appendix -C (Priority Scheduling)

In priority scheduling, there is a priority number assigned to each process. In some systems, the lower the number, the higher the priority. While, in the others, the higher the number, the higher will be the priority. The process with the higher priority among the available processes is given in the CPU. There are two types of priority scheduling algorithm that exists.

1. Preemptive priority scheduling
2. Non-preemptive priority scheduling

The priority number assigned to each of the processes may or may not vary. If the priority number does not change itself throughout the process, it is called static priority, while if it keeps changing itself at the regular intervals, it is called dynamic priority.

(JavaTpoint, 2022)