Module Code & Module Title

CS5001NA Networks and Operating System

Assessment Weightage & Type

20% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Preksha Dahal

London Met ID: 20048964

College ID: np01cp4s210067

Assignment Due Date: 25th April

Assignment Submission Date: 25th April

Word Count (TASK B): 2065

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.

Contents

Task A 4

Introduction 4

Script 5

Testing 14

1. To see the result after giving only ID as the parameter. 14
2. To start the program without giving any parameters. 15
3. To check the response of the program by giving 3 incorrect password 16
4. To check the response of the program after entering the correct password 17
5. To check the response of the program after entering the band name instead of band code. .. 20
6. To check the output after entering the incorrect band code 21
7. To check the output after entering the correct band code 22
8. To pick the names of 4 members and test the program. 23
9. To check the response by picking same member name 24
10. To check the output after giving the wrong member ID. 25
11. To check the output by entering right user ID which contains an external file. 26
12. To choose the band member with no external file 27
13. To exit the program. 28
14. To deny to exit the program. 29

Contents of three files 30

Conclusion 30

Task B 32

Introduction 32

Aims and Objectives 33

Background 34

Process Architecture 34

Process Control Blocks 36

Process Hierarchies 38

Implementation of Process 39

Conclusion 40

References 42

Table of figures

Figure 1: Test 1 (Entering only ID)....................14
Figure 2: Test 2 (Starting the program without entering username and ID)....................15
Figure 3: Entering wrong passwords 3 times....................16
Figure 4: Checking the output after entering correct secret key (I)....................18
Figure 5: Checking the output after entering correct secret key (II)....................19
Figure 6: Entering the band name instead of band code....................20
Figure 7: Checking the output after entering incorrect band code.....................21
Figure 8: Checking the output after entering the correct band code.....................22
Figure 9: Entering four codes of band members instead of three....................23
Figure 10: Checking the output after picking the same member code multiple times.....................24
Figure 11: Checking the output after entering the same code multiple times.....................25
Figure 12: Entering the correct file code that is associated with an external file.....................26
Figure 13: Trying to enter the input as the member that is not associated to any external file.....................27
Figure 14: Exiting the program.....................28
Figure 15: Denying to exit the program....................29
Figure 16: Concept of deadlock....................33
Figure 17: Process architecture of an ongoing process....................36
Figure 18: Data items of PCB....................37

List of Tables

Table 1: To see the result after giving only ID as the parameter.....................14
Table 2: To start the program without giving any parameters.....................15
Table 3: Table 3: To check the response of the program by giving 3 incorrect password.....................16
Table 4: To check the response of the program after entering the correct password.....................17
Table 5: To check the response of the program after entering the band name instead of band code.....................20
Table 6: To check the output after entering the incorrect band code.....................21
Table 7: To check the output after entering the correct band code....................22
Table 8: To pick the names of 4 members and test the program.....................23
Table 9: To check the response by picking same member name....................24
Table 10: To check the output after giving the wrong member ID....................25
Table 11: To check the output by entering right user ID which contains an external file.....................26
Table 12: To choose the band member with no external file....................27
Table 13: To exit the program.....................28
Table 14: To deny to exit the program.....................29
Table 15: Difference between process and thread....................41

Task A

Introduction

A shell is a widely used term in UNIX. It is a layer that provides a user friendly interface to the Unix system. A shell gathers input from the user and then executes the program based on the given input. It is also defined as an surrounding where we can run our programs, commands, and shell scripts. A shell is further divided into the following sub-categories;

Bourne shell:

This shell is denoted by sh and uses $ as its default sign. This shell is the original Unix shell as it is faster when compared to other shell. Bourne shell is also the most preferred shell.

Bash shell:

Bash shell is basically the modified version of the Bourne shell and it stands for Bourne Again Shell. This shell is denoted by bash and is also the default shell in many Linux distributions.

C shell:

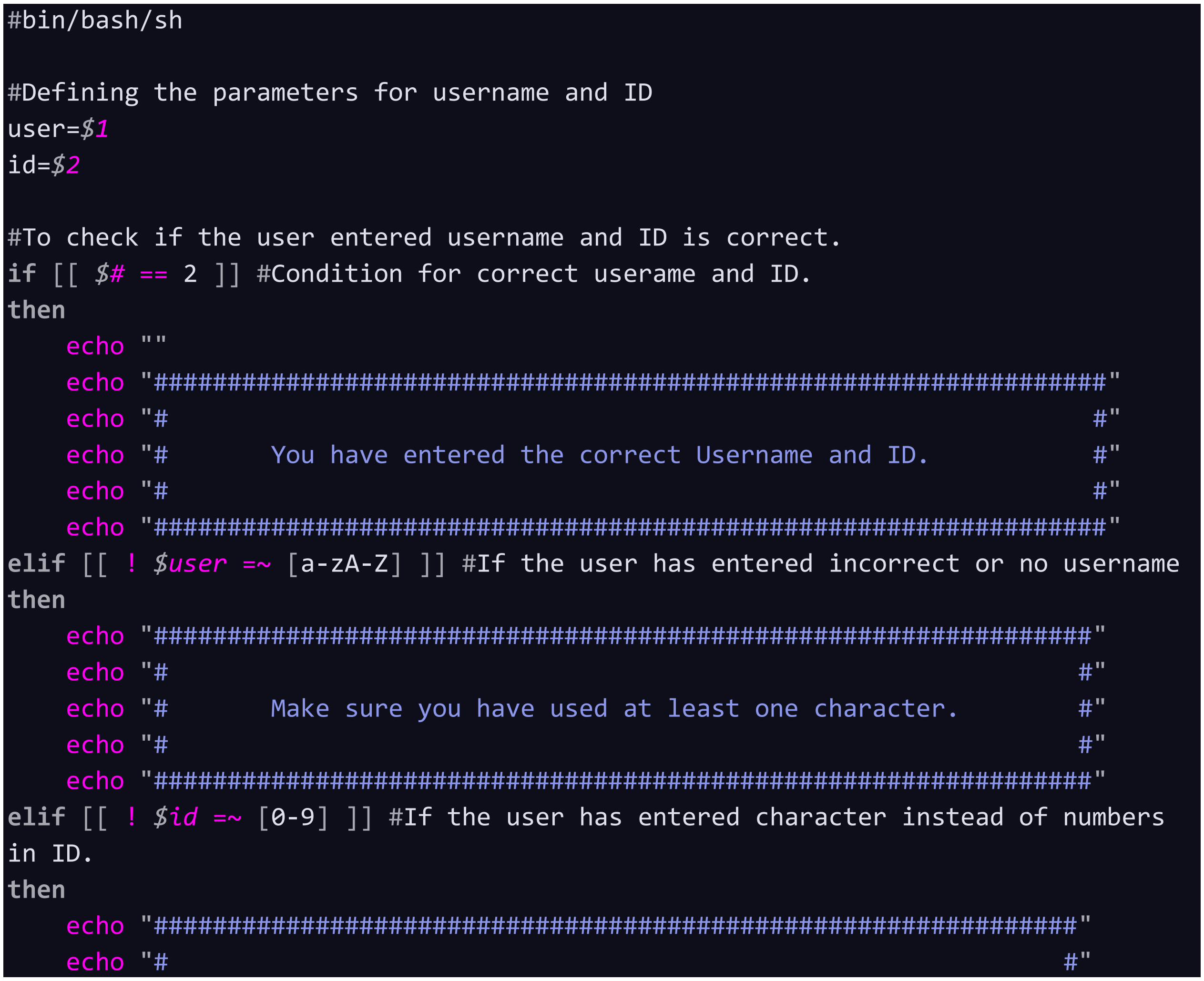
This shell is represented by csh and incorporated the features like aliases and command history. The % character is considered the default prompt in this shell.

Ksh shell:

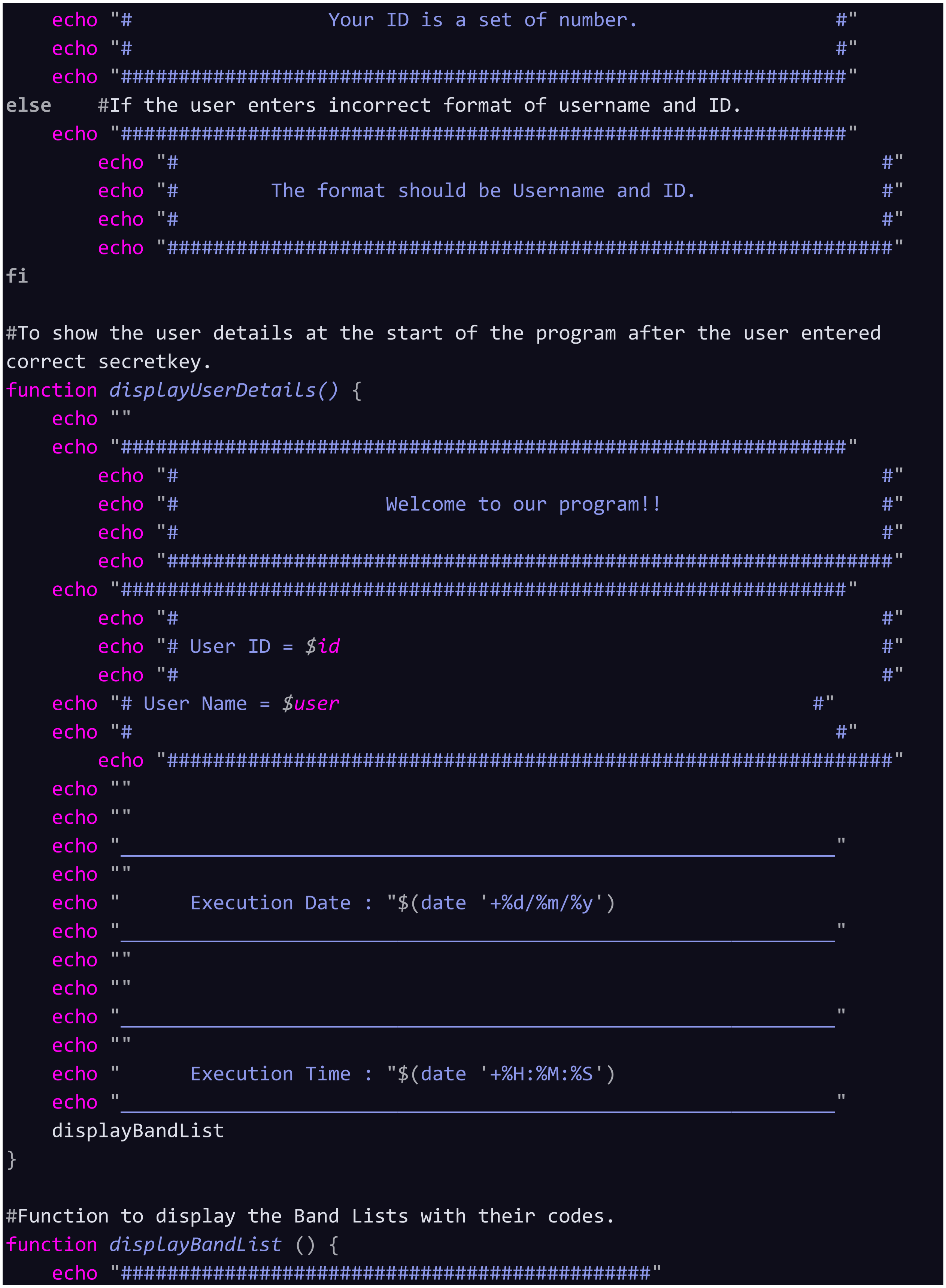
This shell is denoted by ksh itself and includes the interactive features like built-in functions and string manipulation facilities.

A shell script is a text file which contains a structure of commands and code for a UNIX based operating system. These scripts are widely used when a user has to perform multiple tasks at a particular time as it reduces the amount of time consumption. Scripting language is a programming language that ropes the writing scripts, programs transcribed for a software nature that automate the performance of tasks. The primary concept of shell script includes a list of commands that are usually listed in the order of execution. A proper shell script contains comments that proceeds by # sign. These comments explain the steps and flow of the program. Besides the comment, a proper shell script contains functions, conditional statements, variables, and user-friendly attributes. An example of a proper shell script that is ready for execution and compilation is displayed below.

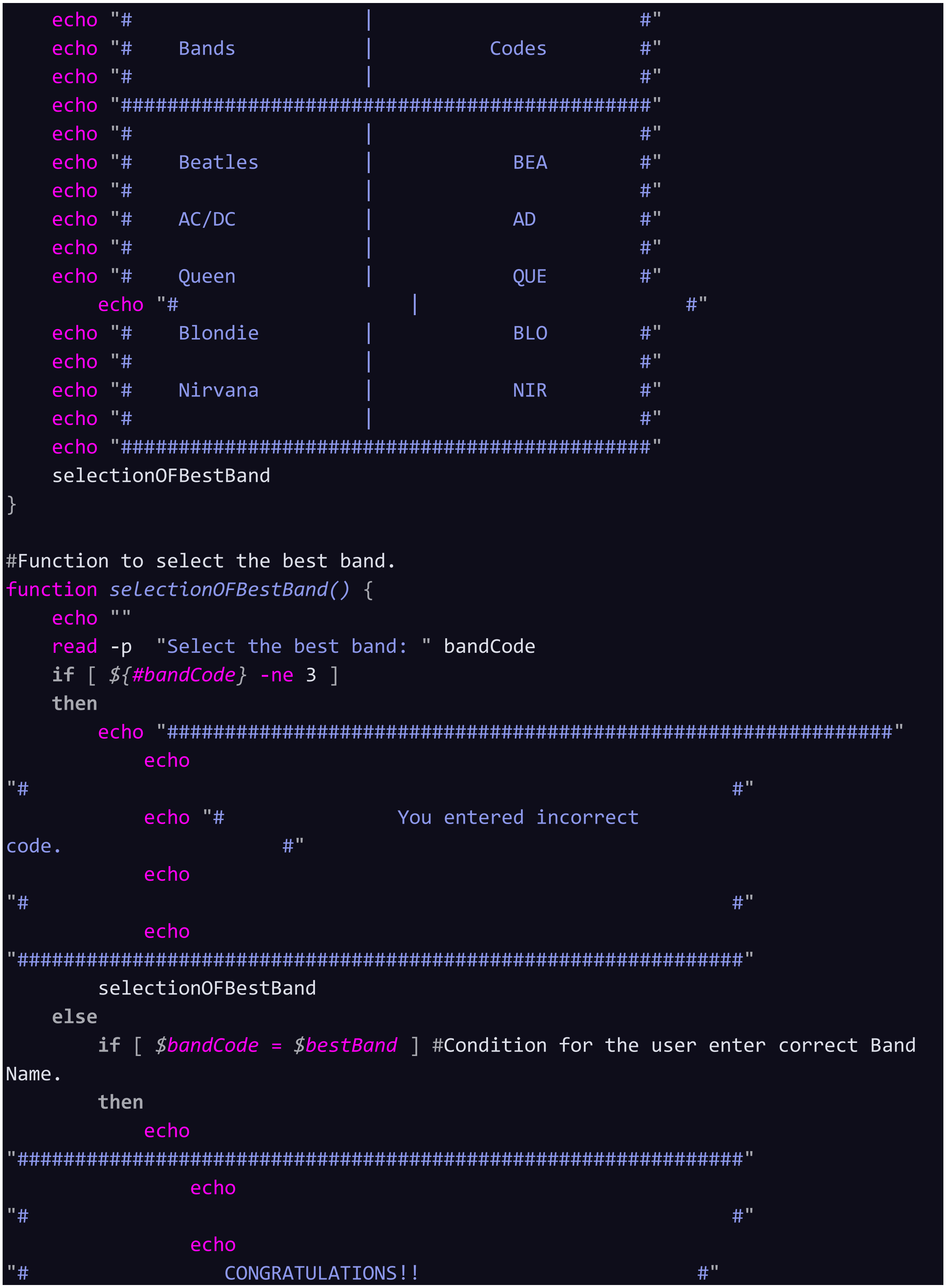
Script



```
#bin/bash/sh

#Defining the parameters for username and ID
user=$1
id=$2

#To check if the user entered username and ID is correct.
if [[ $# == 2 ]] #Condition for correct userame and ID.
then
    echo ""
    echo "#################################################################"
    echo "#                                                               #"
    echo "#       You have entered the correct Username and ID.           #"
    echo "#                                                               #"
    echo "#################################################################"
elif [[ ! $user =~ [a-zA-Z] ]] #If the user has entered incorrect or no username
then
    echo "################################################################"
    echo "#                                                              #"
    echo "#       Make sure you have used at least one character.        #"
    echo "#                                                              #"
    echo "################################################################"
elif [[ ! $id =~ [0-9] ]] #If the user has entered character instead of numbers
in ID.
then
    echo "###############################################################"
    echo "#                                                             #"
```

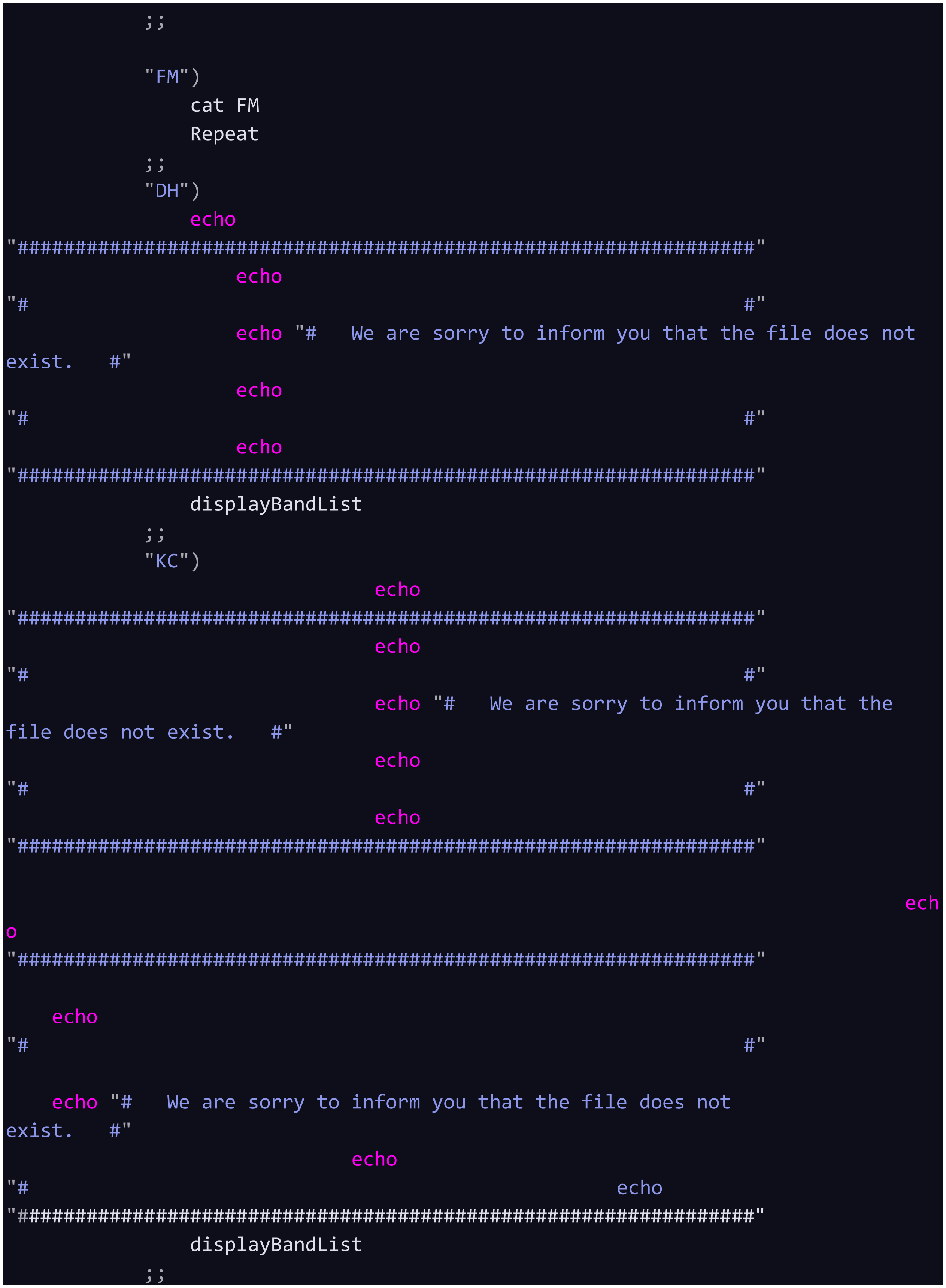


```
    echo "#                 Your ID is a set of number.                 #"
    echo "#                                                             #"
    echo "###############################################################"
else    #If the user enters incorrect format of username and ID.
    echo "###############################################################"
        echo "#                                                             #"
        echo "#        The format should be Username and ID.                #"
        echo "#                                                             #"
        echo "###############################################################"
fi

#To show the user details at the start of the program after the user entered
correct secretkey.
function displayUserDetails() {
    echo ""
    echo "###############################################################"
        echo "#                                                             #"
        echo "#                  Welcome to our program!!                   #"
        echo "#                                                             #"
        echo "###############################################################"
    echo "###############################################################"
        echo "#                                                             #"
        echo "# User ID = $id                                               #"
        echo "#                                                             #"
    echo "# User Name = $user                                         #"
    echo "#                                                             #"
        echo "###############################################################"
    echo ""
    echo ""
    echo "______________________________________________________________"
    echo ""
    echo "      Execution Date : "$(date '+%d/%m/%y')
    echo "______________________________________________________________"
    echo ""
    echo ""
    echo "______________________________________________________________"
    echo ""
    echo "      Execution Time : "$(date '+%H:%M:%S')
    echo "______________________________________________________________"
    displayBandList
}

#Function to display the Band Lists with their codes.
function displayBandList () {
    echo "##############################################"
```

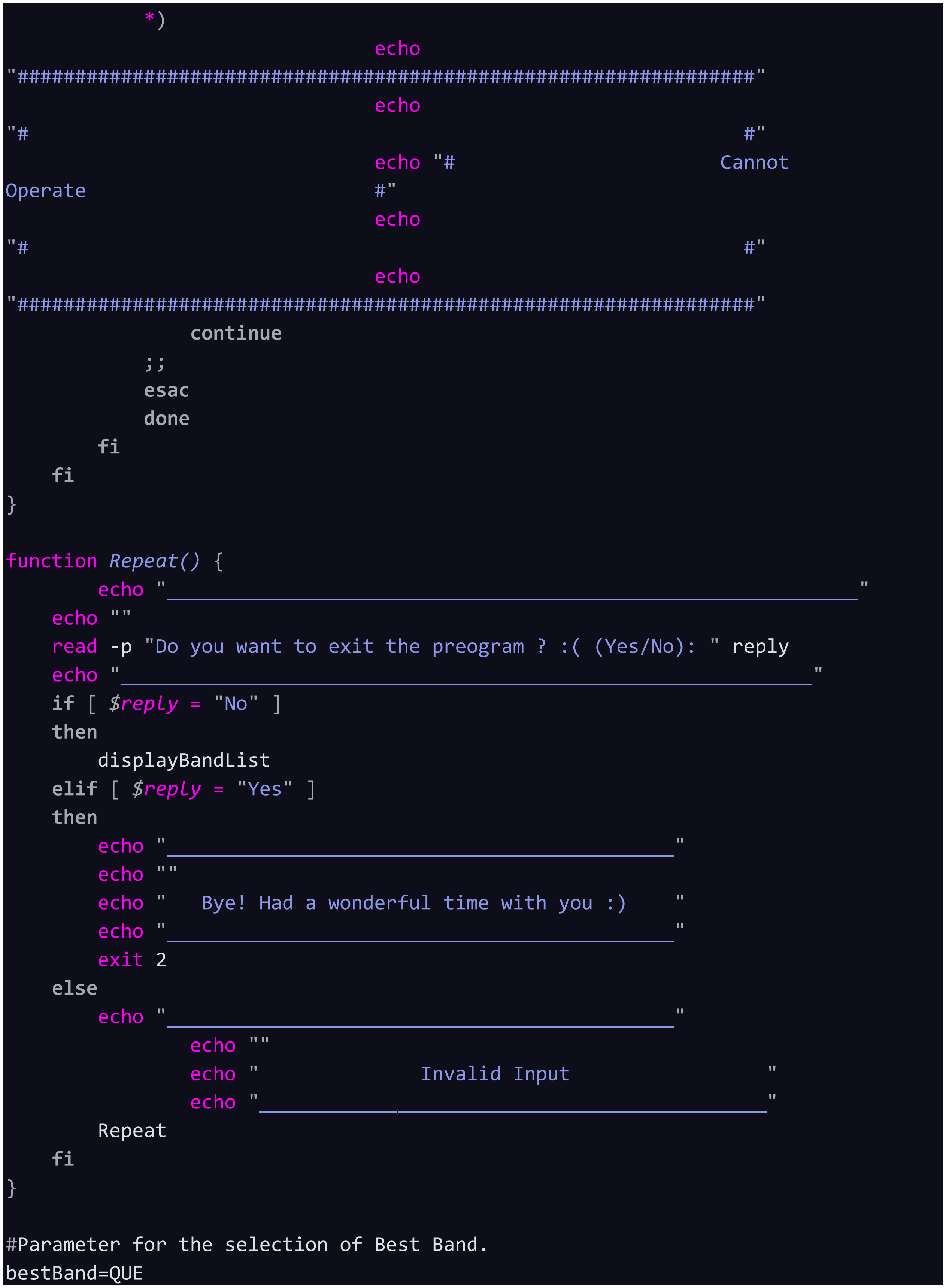


```
    echo "#                    |                       #"
    echo "#    Bands           |          Codes        #"
    echo "#                    |                       #"
    echo "##############################################"
    echo "#                    |                       #"
    echo "#    Beatles         |            BEA        #"
    echo "#                    |                       #"
    echo "#    AC/DC           |            AD         #"
    echo "#                    |                       #"
    echo "#    Queen           |            QUE        #"
        echo "#                    |                       #"
    echo "#    Blondie         |            BLO        #"
    echo "#                    |                       #"
    echo "#    Nirvana         |            NIR        #"
    echo "#                    |                       #"
    echo "##############################################"
    selectionOFBestBand
}

#Function to select the best band.
function selectionOFBestBand() {
    echo ""
    read -p  "Select the best band: " bandCode
    if [ ${#bandCode} -ne 3 ]
    then
        echo "################################################################"
            echo
"#                                                             #"
            echo "#               You entered incorrect
code.                   #"
            echo
"#                                                             #"
            echo
"###############################################################"
        selectionOFBestBand
    else
        if [ $bandCode = $bestBand ] #Condition for the user enter correct Band
Name.
        then
            echo
"###############################################################"
                echo
"#                                                             #"
                echo
"#                  CONGRATULATIONS!!                       #"
```

```
            echo
"#                                                            #"
                echo
"#                                                            #"
            echo "#         Your assumption about the best band is
correct.     #"
                echo
"##############################################################"
            bandMembers
        else  #Until condition to display the information about wrong assumption.
            until [ $bandCode = $bestBand ]
            do
                echo "Your assumption is incorrect."
                selectionOFBestBand
            done
        fi
    fi
}

#Function to display about the band members and their codes.
function bandMembers() {
    echo "***************************************************"
    echo "*                      !#!                        *"
    echo "*     Members          !#!         Codes          *"
    echo "***************************************************"
    echo "*                      !#!                        *"
    echo "*    John Lennon       !#!          JL            *"
    echo "*                      !#!                        *"
    echo "*    Angus Young       !#!          AY            *"
    echo "*                      !#!                        *"
    echo "*    Freddie Mercury   !#!          FM            *"
    echo "*                      !#!                        *"
    echo "*    Debbie Harry      !#!          DH            *"
    echo "*                      !#!                        *"
    echo "*    Kurt Coban        !#!          KC            *"
    echo "*                      !#!                        *"
    echo "***************************************************"
    bandMembersSelection
}

function bandMembersSelection() {
    read -p "Enter the code of any three band members: " Member1 Member2 Member3
    # If the user selects more or less than 3 numbers while selecting the band
member.
    if [[ ${#Member1} -ne 2 || ${#Member2} -ne 2 || ${#Member3} -ne 2 ]]
    then
        echo "###############################################################"
            echo
"#                                                            #"
            echo "#       You are supposed to use three member
codes.          #"
            echo
"#                                                            #"
            echo
"###############################################################"
        bandMembersSelection
    else
    # If the user enters same number for band member multiple times.
        if [[ $Member1 == $Member2 ]] || [[ $Member2 == $Member3 ]] || [[
$Member1 == $Member3 ]]
    then
        echo
"################################################################################
###########"
            echo
"#
          #"
            echo "#           Please note that you cannot enter the same number
multiple times.         #"
            echo
"#
          #"
            echo
"################################################################################
###########"
        bandMembersSelection
    else
        # To display the contents of the available files
        PS3="Choose one of the member from below; "
        select Member in $Member1 $Member2 $Member3
        do
            case $Member in
            "JL")
                cat JL
                Repeat
                ;;

            "AY")
                cat AY
                Repeat
```

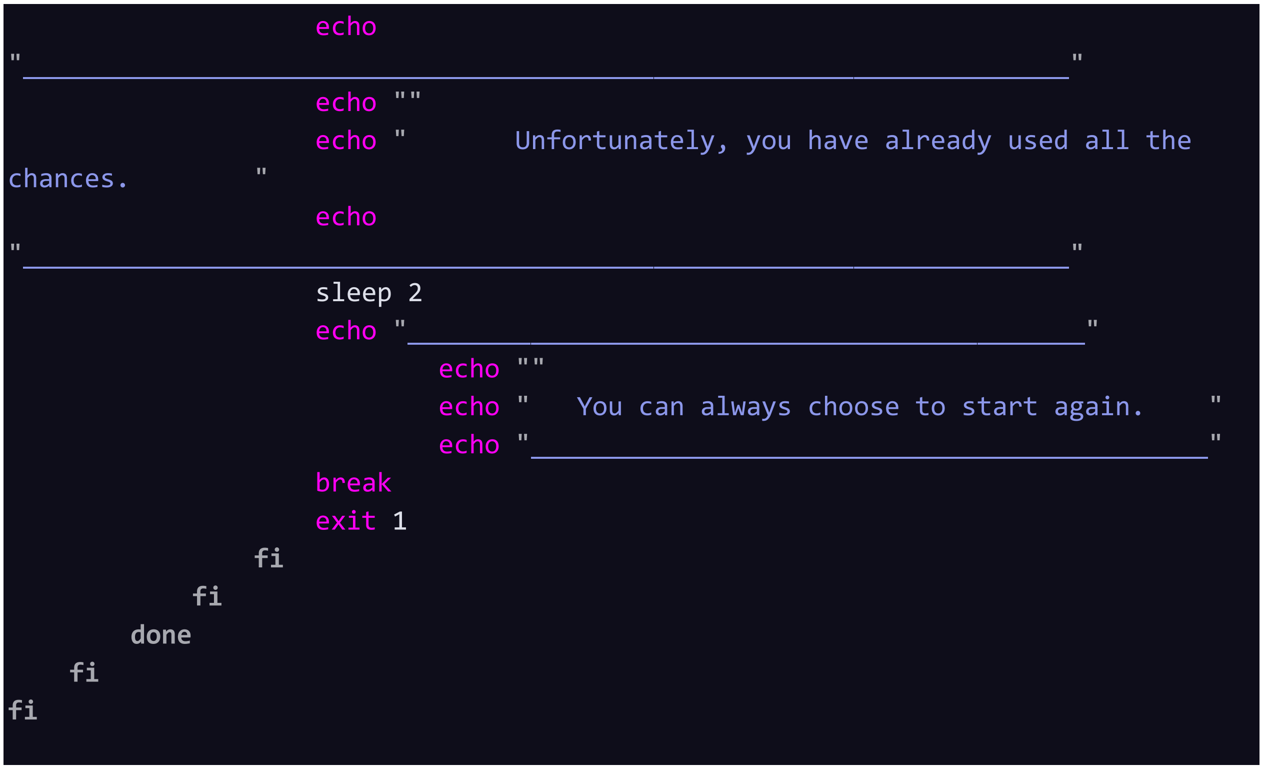


```
            ;;

            "FM")
                cat FM
                Repeat
            ;;
            "DH")
                echo
"################################################################"
                    echo
"#                                                              #"
                    echo "#   We are sorry to inform you that the file does not
exist.   #"
                    echo
"#                                                              #"
                    echo
"################################################################"
                displayBandList
            ;;
            "KC")
                                echo
"################################################################"
                                echo
"#                                                              #"
                                echo "#   We are sorry to inform you that the
file does not exist.   #"
                                echo
"#                                                              #"
                                echo
"################################################################"

                                                                              ech
o
"################################################################"

    echo
"#                                                              #"

    echo "#   We are sorry to inform you that the file does not
exist.   #"
                              echo
"#                                                   echo
"################################################################"
                displayBandList
            ;;
```



```
            *)
                                echo
"################################################################"
                                echo
"#                                                               #"
                                echo "#                        Cannot
Operate                         #"
                                echo
"#                                                               #"
                                echo
"################################################################"
                continue
            ;;
            esac
            done
        fi
    fi
}

function Repeat() {
        echo "____________________________________________________________"
    echo ""
    read -p "Do you want to exit the preogram ? :( (Yes/No): " reply
    echo "____________________________________________________________"
    if [ $reply = "No" ]
    then
        displayBandList
    elif [ $reply = "Yes" ]
    then
        echo "____________________________________________"
        echo ""
        echo "   Bye! Had a wonderful time with you :)    "
        echo "____________________________________________"
        exit 2
    else
        echo "____________________________________________"
                echo ""
                echo "             Invalid Input                 "
                echo "____________________________________________"
        Repeat
    fi
}

#Parameter for the selection of Best Band.
bestBand=QUE
```

```
#Defining the password for secret key.
secretKey="NOSp2"

#Initializing the value of check to count the number of chances to enter the
secret key.
check=1

# IF condition to provide 3 chances for the user to enter correct secret key.
if [ $# = 0 ]
then
    echo ""
    echo "Enter Parameters"
    echo ""
elif [ $# = 1 ]
then
    echo ""
    echo "You have not entered the username or ID"
    echo ""
else
    read -p "Enter the password/Secret Key: " -s value
    if [[ $value == $secretKey ]]
    then
        displayUserDetails
    else
        for (( tries=1; tries<4; tries++ ))
        do
            echo ""
            echo "You have entered wrong key $check time."
            if [ $check = 2 ]
            then
                echo ""
                echo "You have got one last chance to enter the correct key."
            fi
            read -p "Enter the secret key or password: " -s key
            if [[ $key == $secretKey ]]
            then
                displayUserDetails
                break
            else
                check=$((check+1))
                if [ $check = 3 ]
                then
                    echo ""
```



```
                    echo
"____________________________________________________________________"
                    echo ""
                    echo "       Unfortunately, you have already used all the
chances.        "
                    echo
"____________________________________________________________________"
                    sleep 2
                    echo "____________________________________________"
                            echo ""
                            echo "   You can always choose to start again.    "
                            echo "____________________________________________"
                    break
                    exit 1
                fi
            fi
        done
    fi
fi
```

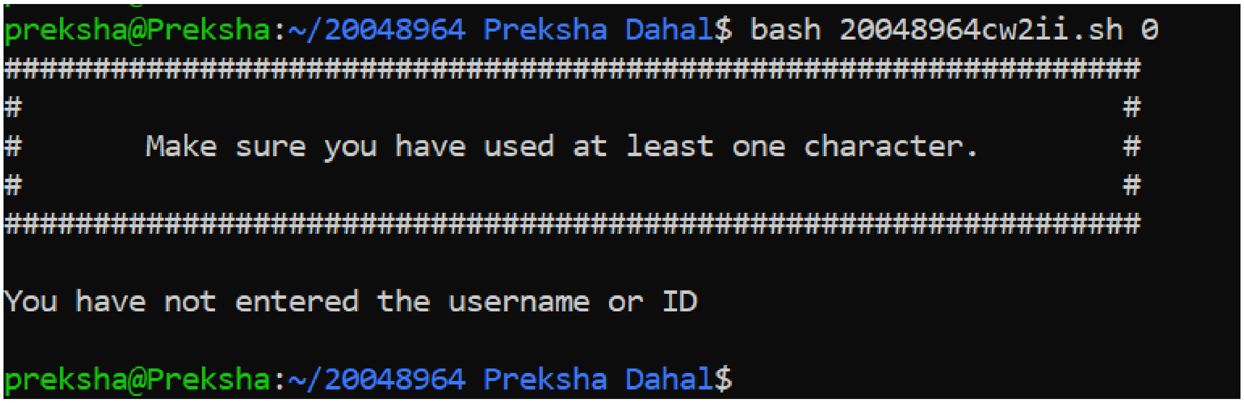
Testing

1. To see the result after giving only ID as the parameter.

Test No.	01
Input	bash 20048964cw2ii.sh
Expected Output	The program should not start and a message about the correct format should be displayed.
Actual Output	The program did not start and a message about the correct format was displayed.
Test Result	The test was successful.

Table 1: To see the result after giving only ID as the parameter.

Output of Test 1



```
preksha@Preksha:~/20048964 Preksha Dahal$ bash 20048964cw2ii.sh 0
################################################################
#                                                              #
#       Make sure you have used at least one character.        #
#                                                              #
################################################################

You have not entered the username or ID

preksha@Preksha:~/20048964 Preksha Dahal$
```

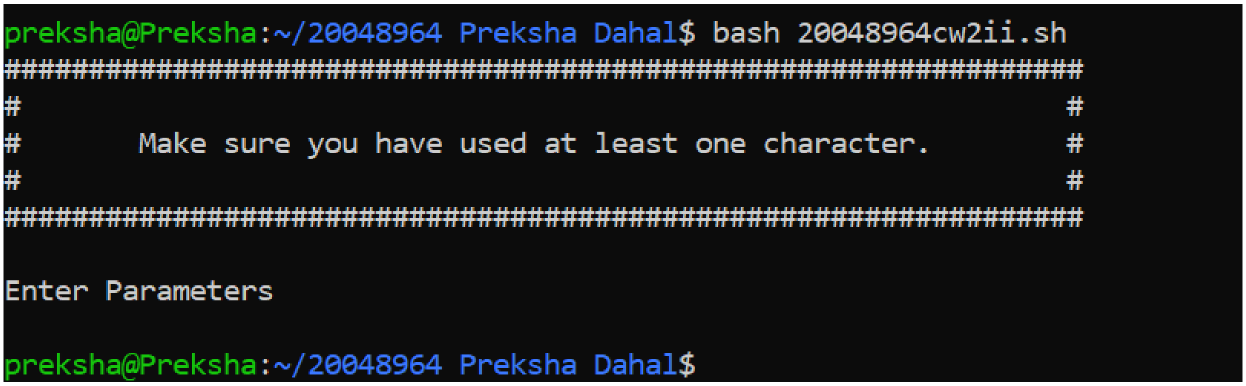
Figure 1: Test 1 (Entering only ID)

2. To start the program without giving any parameters.

Test No.	02
Input	bash 20048964cw2ii.sh
Expected Output	The program should give instruction about correct format.
Actual Output	The program displayed a statement that says “Make sure you have entered at least one parameter.”
Test Result	The test was successful.

Table 2: To start the program without giving any parameters.

Output of Test 2



```
preksha@Preksha:~/20048964 Preksha Dahal$ bash 20048964cw2ii.sh
################################################################
#                                                              #
#       Make sure you have used at least one character.        #
#                                                              #
################################################################

Enter Parameters

preksha@Preksha:~/20048964 Preksha Dahal$
```

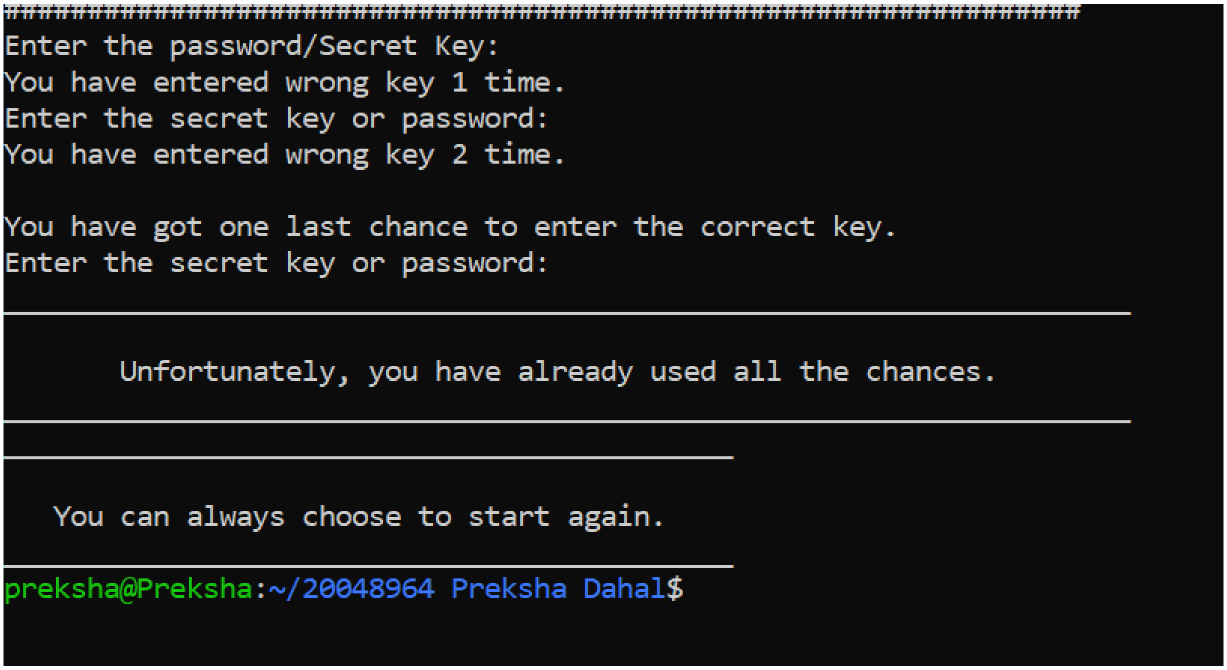
Figure 2: Test 2 (Starting the program without entering username and ID)

3. To check the response of the program by giving 3 incorrect password

Test No.	03
Input	bash 20048964cw2ii.sh Preksha 01 *3 incorrect passwords.*
Expected Output	The program should display the number of chances left after each try.
Actual Output	The program should displayed the number of chances left after each try.
Test Result	The test was successful.

Table 3: Table 3: To check the response of the program by giving 3 incorrect password.

Output of Test 3



```
#################################################################
Enter the password/Secret Key:
You have entered wrong key 1 time.
Enter the secret key or password:
You have entered wrong key 2 time.

You have got one last chance to enter the correct key.
Enter the secret key or password:
____________________________________________________________________

       Unfortunately, you have already used all the chances.
____________________________________________________________________
____________________________________________

   You can always choose to start again.
____________________________________________
preksha@Preksha:~/20048964 Preksha Dahal$
```

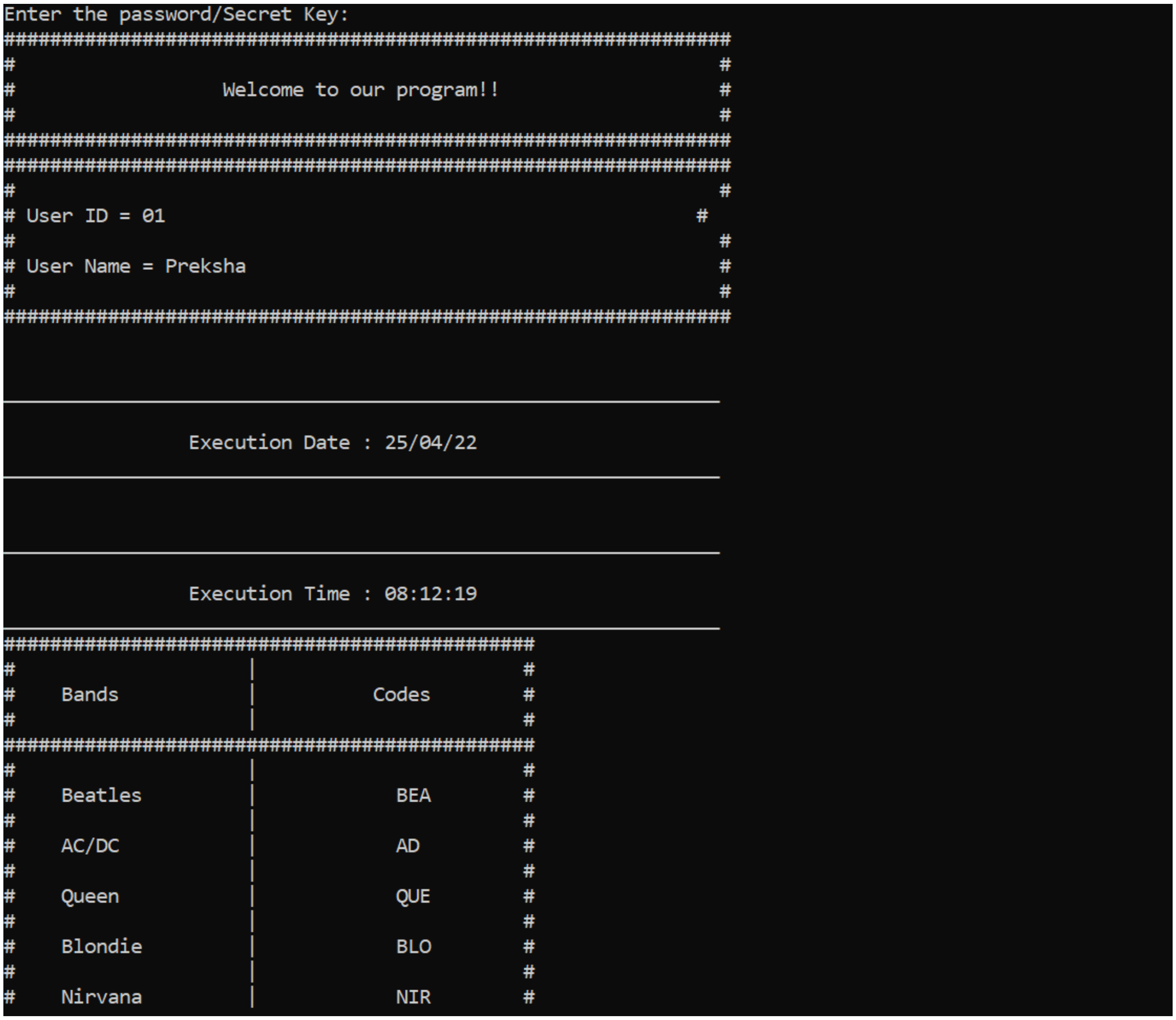
Figure 3: Entering wrong passwords 3 times

4. To check the response of the program after entering the correct password

Test No.	04
Input	NOSp2
Expected Output	The program should start by displaying the correct messages.
Actual Output	The program started displaying the information about under name, ID, the time of execution and the date of execution.
Test Result	The test was successful.

Table 4: To check the response of the program after entering the correct password.

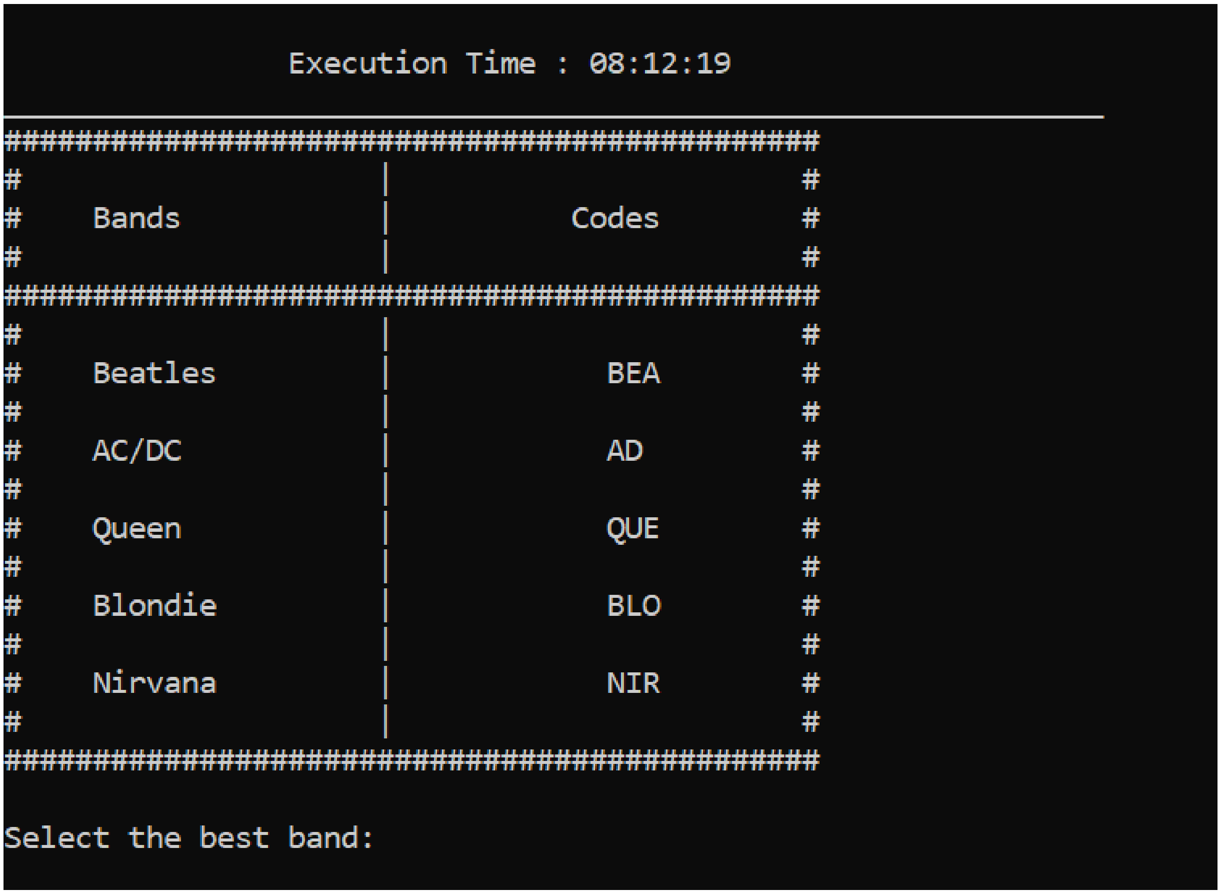
Output of Test 4



```
Enter the password/Secret Key:
###############################################################
#                                                             #
#                  Welcome to our program!!                   #
#                                                             #
###############################################################
###############################################################
#                                                             #
# User ID = 01                                              #
#                                                             #
# User Name = Preksha                                         #
#                                                             #
###############################################################


______________________________________________________________

                Execution Date : 25/04/22
______________________________________________________________


______________________________________________________________

                Execution Time : 08:12:19
______________________________________________________________
##############################################
#                    |                       #
#    Bands           |          Codes        #
#                    |                       #
##############################################
#                    |                       #
#    Beatles         |            BEA        #
#                    |                       #
#    AC/DC           |            AD         #
#                    |                       #
#    Queen           |            QUE        #
#                    |                       #
#    Blondie         |            BLO        #
#                    |                       #
#    Nirvana         |            NIR        #
```

Figure 4: Checking the output after entering correct secret key (I)

Output of Test 4 (Continued)



```
                Execution Time : 08:12:19

##############################################
#                    |                       #
#    Bands           |          Codes        #
#                    |                       #
##############################################
#                    |                       #
#    Beatles         |            BEA        #
#                    |                       #
#    AC/DC           |            AD         #
#                    |                       #
#    Queen           |            QUE        #
#                    |                       #
#    Blondie         |            BLO        #
#                    |                       #
#    Nirvana         |            NIR        #
#                    |                       #
##############################################

Select the best band:
```

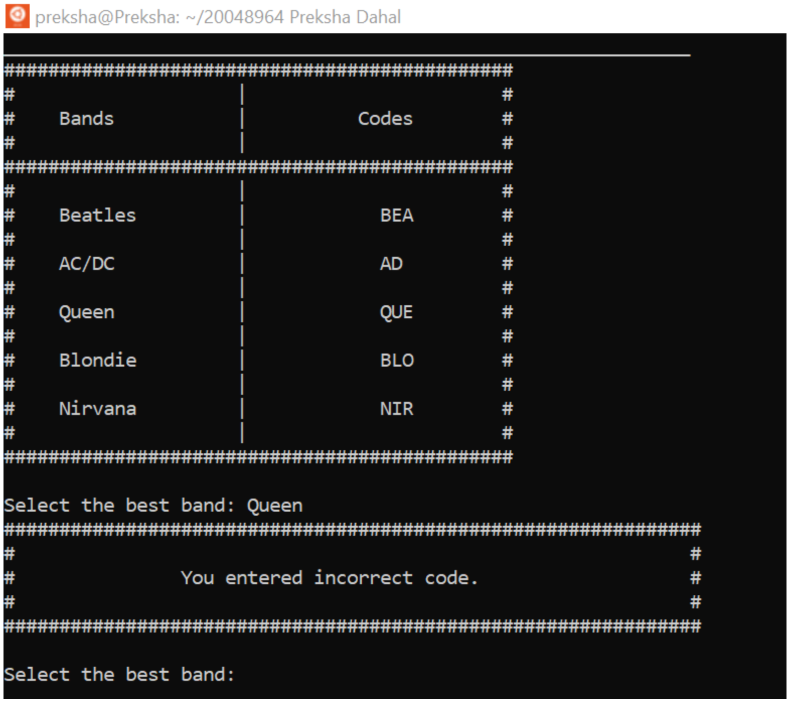
Figure 5: Checking the output after entering correct secret key (II)

5. To check the response of the program after entering the band name instead of band code.

Test No.	05
Input	Queen
Expected Output	The program will not proceed forward and inform the user about the incorrect code.
Actual Output	The program did not proceed forward and informed about the incorrect code.
Test Result	The test was successful.

Table 5: To check the response of the program after entering the band name instead of band code.

Output of Test 5



preksha@Preksha: ~/20048964 Preksha Dahal

```
##############################################
#                    |                       #
#    Bands           |          Codes        #
#                    |                       #
##############################################
#                    |                       #
#    Beatles         |            BEA        #
#                    |                       #
#    AC/DC           |            AD         #
#                    |                       #
#    Queen           |            QUE        #
#                    |                       #
#    Blondie         |            BLO        #
#                    |                       #
#    Nirvana         |            NIR        #
#                    |                       #
##############################################

Select the best band: Queen
###############################################################
#                                                             #
#               You entered incorrect code.                   #
#                                                             #
###############################################################

Select the best band:
```

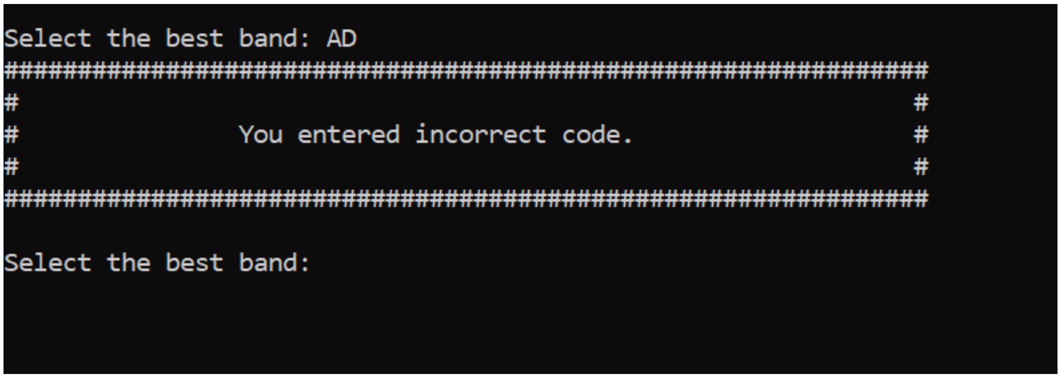
Figure 6: Entering the band name instead of band code

6. To check the output after entering the incorrect band code

Test No.	06
Input	AD
Expected Output	A message should be displayed about the incorrect code. The user will be given another chance to enter the band code.
Actual Output	A message was displayed about the incorrect code and the user was given another chance to enter the band code.
Test Result	The test was successful.

Table 6: To check the output after entering the incorrect band code.

Output of Test 5



```
Select the best band: AD
###############################################################
#                                                             #
#               You entered incorrect code.                   #
#                                                             #
###############################################################

Select the best band:
```

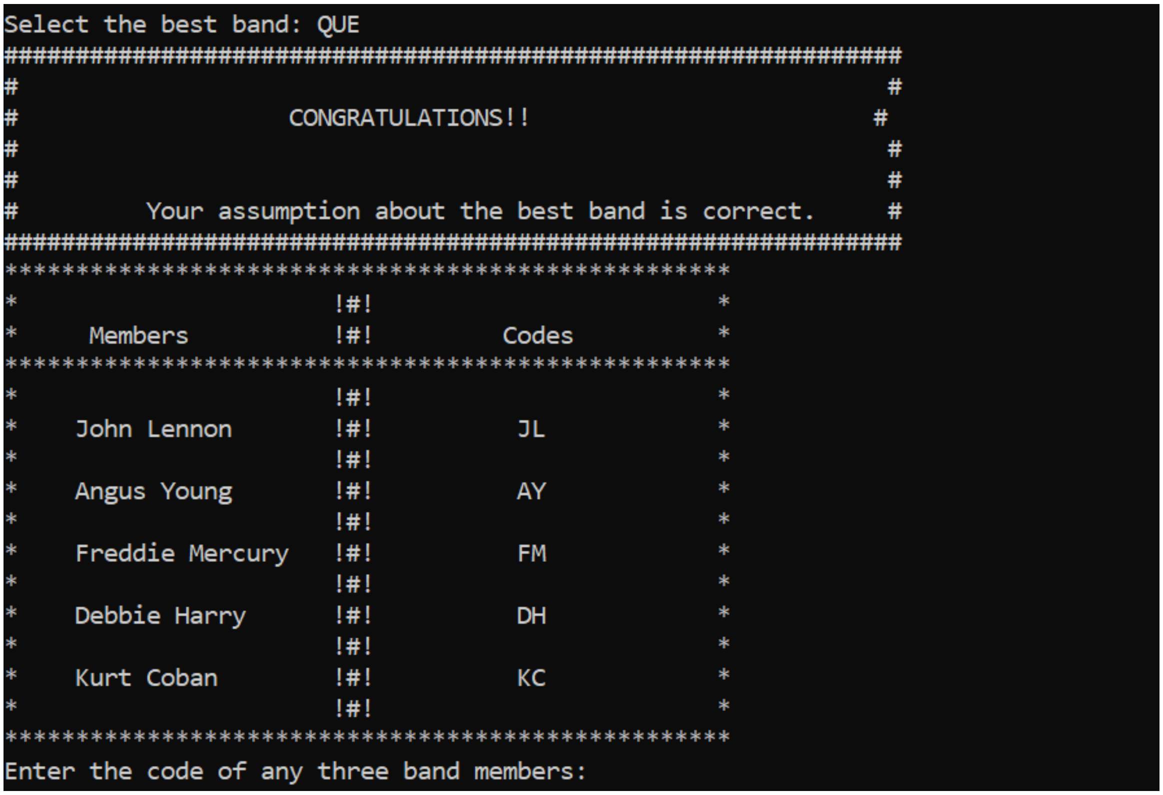
Figure 7: Checking the output after entering incorrect band code.

7. To check the output after entering the correct band code

Test No.	07
Input	QUE
Expected Output	A message should be displayed about the correct band name and the member of that band should be displayed along with their respective band member codes.
Actual Output	A message about entering the correct band name was displayed and the names of member of the band Queen band was displayed along with their respective band member codes.
Test Result	The test was successful.

Table 7: To check the output after entering the correct band code

Output of Test 7



```
Select the best band: QUE
###############################################################
#                                                             #
#                   CONGRATULATIONS!!                        #
#                                                             #
#                                                             #
#         Your assumption about the best band is correct.     #
###############################################################
***************************************************
*                      !#!                        *
*     Members          !#!         Codes          *
***************************************************
*                      !#!                        *
*    John Lennon       !#!          JL            *
*                      !#!                        *
*    Angus Young       !#!          AY            *
*                      !#!                        *
*    Freddie Mercury   !#!          FM            *
*                      !#!                        *
*    Debbie Harry      !#!          DH            *
*                      !#!                        *
*    Kurt Coban        !#!          KC            *
*                      !#!                        *
***************************************************
Enter the code of any three band members:
```

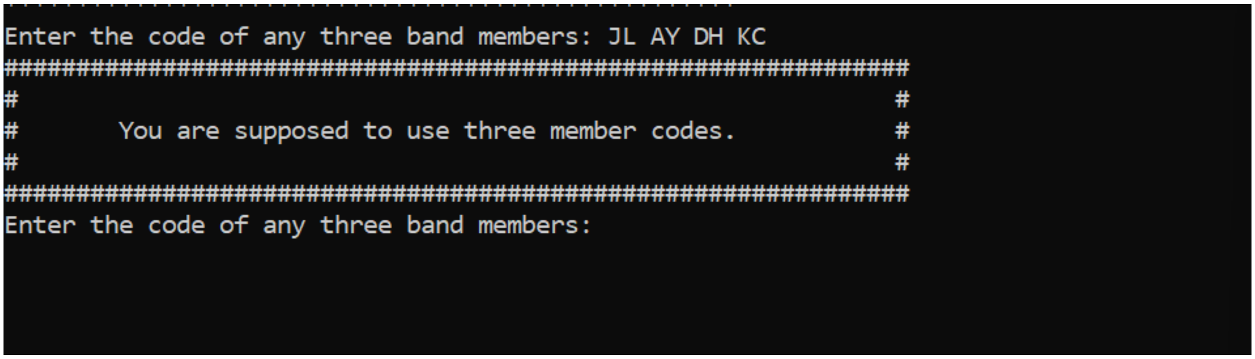
Figure 8: Checking the output after entering the correct band code.

8. To pick the names of 4 members and test the program.

Test No.	08
Input	JL AY DH KC
Expected Output	A message should be displayed about the incorrect format of entering names.
Actual Output	A message was displayed about the incorrect format of entering names and the user was offered to choose the names again.
Test Result	The test was successful.

Table 8: To pick the names of 4 members and test the program.

Output of Test 8



```
Enter the code of any three band members: JL AY DH KC
###############################################################
#                                                             #
#       You are supposed to use three member codes.           #
#                                                             #
###############################################################
Enter the code of any three band members:
```

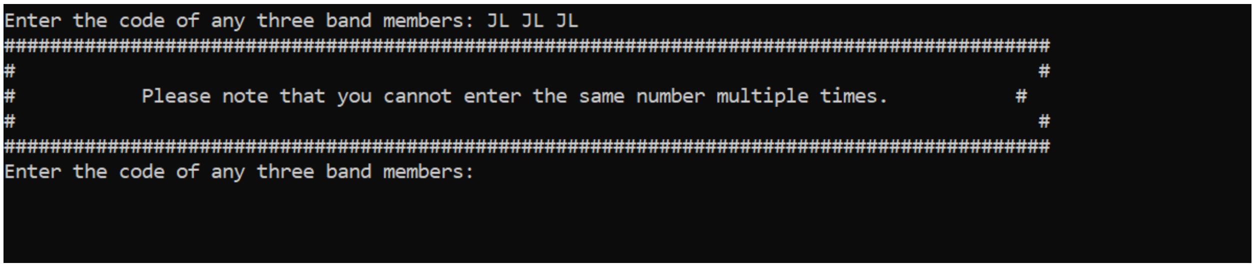
Figure 9: Entering four codes of band members instead of three.

9. To check the response by picking same member name

Test No.	09
Input	JL JL JL
Expected Output	The program should display some sort of information about inserting the same member code multiple times.
Actual Output	The program displayed a text saying, “Please note that you cannot enter the same number multiple times.”
Test Result	The test was successful.

Table 9: To check the response by picking same member name

Output of Task 9



```
Enter the code of any three band members: JL JL JL
###########################################################################################
#                                                                                         #
#           Please note that you cannot enter the same number multiple times.          #
#                                                                                         #
###########################################################################################
Enter the code of any three band members:
```

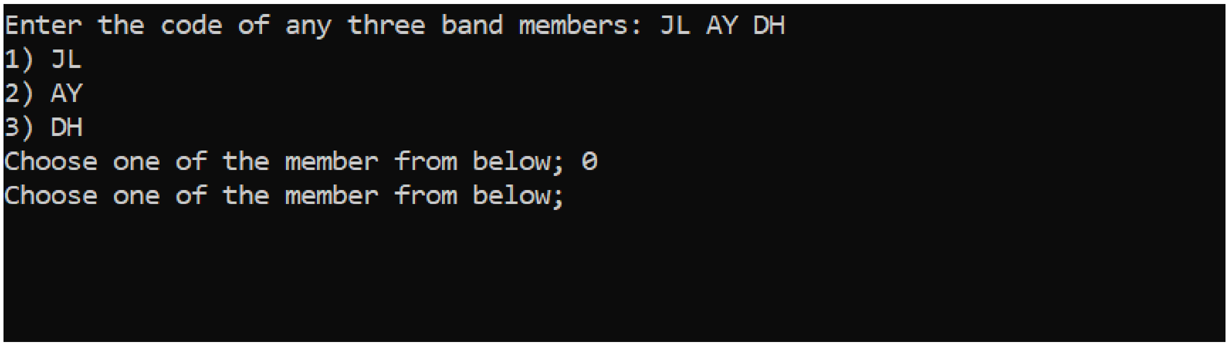
Figure 10: Checking the output after picking the same member code multiple times.

10.To check the output after giving the wrong member ID.

Test No.	10
Input	0
Expected Output	The program will allow the user to re-enter the number or code of the member.
Actual Output	The program allowed the user to re-enter the code of the member.
Test Result	The test was successful.

Table 10: To check the output after giving the wrong member ID.

Output of Task 10



```
Enter the code of any three band members: JL AY DH
1) JL
2) AY
3) DH
Choose one of the member from below; 0
Choose one of the member from below;
```

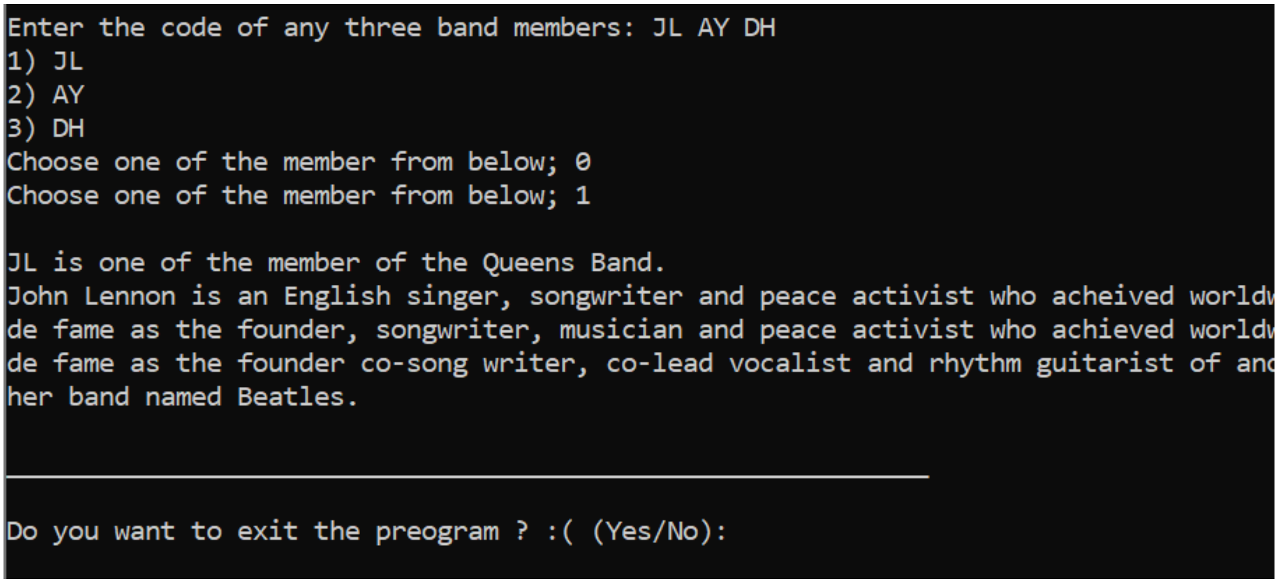
Figure 11: Checking the output after entering the same code multiple times.

11. To check the output by entering right user ID which contains an external file.

Test No.	11
Input	1
Expected Output	The content present inside the file of number 1 i.e. JL should be opened.
Actual Output	The contents of the JL file was displayed.
Test Result	The test was successful.

Table 11: To check the output by entering right user ID which contains an external file.

Output of Task 11



```
Enter the code of any three band members: JL AY DH
1) JL
2) AY
3) DH
Choose one of the member from below; 0
Choose one of the member from below; 1

JL is one of the member of the Queens Band.
John Lennon is an English singer, songwriter and peace activist who acheived worldw
de fame as the founder, songwriter, musician and peace activist who achieved worldw
de fame as the founder co-song writer, co-lead vocalist and rhythm guitarist of and
her band named Beatles.


____________________________________________________________

Do you want to exit the preogram ? :( (Yes/No):
```

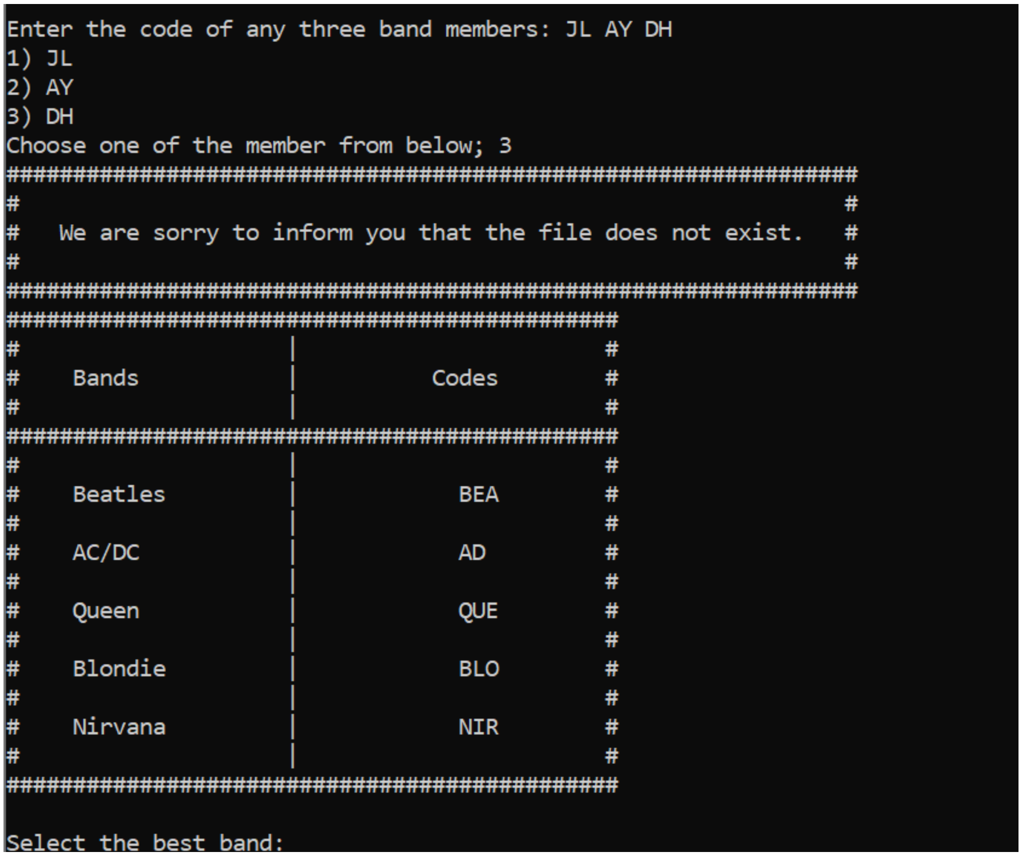
Figure 12: Entering the correct file code that is associated with an external file.

12. To choose the band member with no external file

Test No.	12
Input	3 (DH)
Expected Output	The program should display a message about the inconvenience and also display the list of all the bands again.
Actual Output	The program displayed a message about the no existing file and displayed the list of all the bands again.
Test Result	The test was successful.

Table 12: To choose the band member with no external file

Output of Task 12



```
Enter the code of any three band members: JL AY DH
1) JL
2) AY
3) DH
Choose one of the member from below; 3
################################################################
#                                                              #
#   We are sorry to inform you that the file does not exist.   #
#                                                              #
################################################################
##############################################
#                   |                        #
#    Bands          |          Codes         #
#                   |                        #
##############################################
#                   |                        #
#    Beatles        |            BEA         #
#                   |                        #
#    AC/DC          |            AD          #
#                   |                        #
#    Queen          |            QUE         #
#                   |                        #
#    Blondie        |            BLO         #
#                   |                        #
#    Nirvana        |            NIR         #
#                   |                        #
##############################################

Select the best band:
```

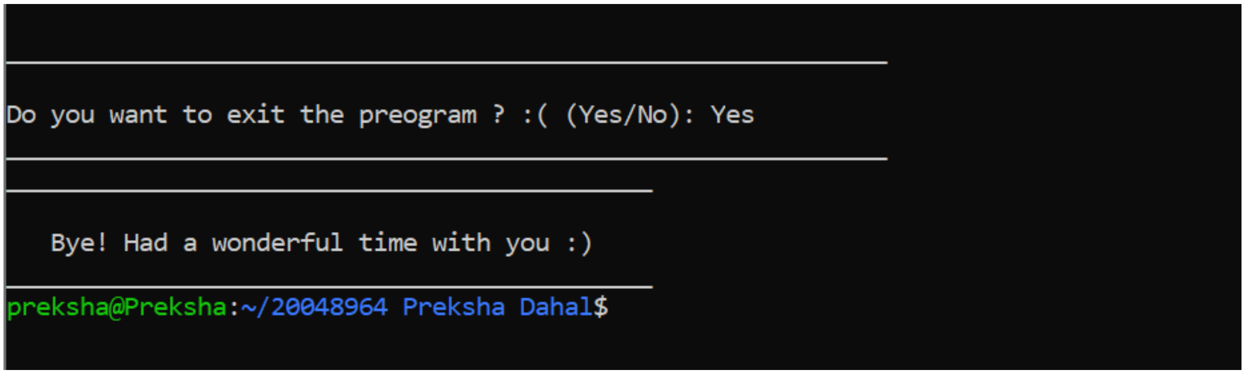
Figure 13: Trying to enter the input as the member that is not associated to any external file.

13. To exit the program.

Test No.	13
Input	Yes
Expected Output	The program should stop and the user should be sent to the parent directory.
Actual Output	The program was stopped, and the user was sent to the parent directory.
Test Result	The test was successful.

Table 13: To exit the program.

Output of Task 13



```
________________________________________________________
Do you want to exit the preogram ? :( (Yes/No): Yes
________________________________________________________
_________________________________________
   Bye! Had a wonderful time with you :)
_________________________________________
preksha@Preksha:~/20048964 Preksha Dahal$
```

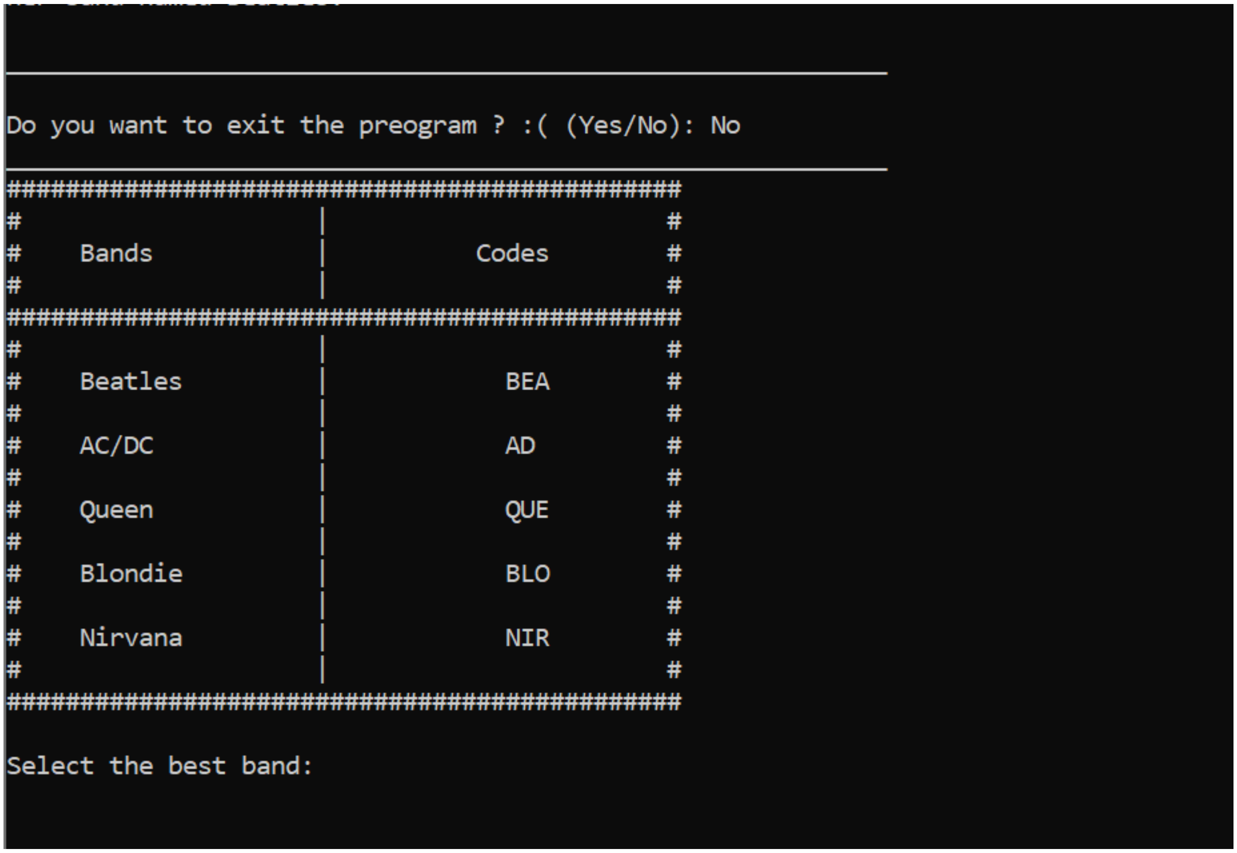
Figure 14: Exiting the program.

14. To deny to exit the program.

Test No.	14
Input	No
Expected Output	The program displays the list of band names and gets re-executed from there.
Actual Output	The program displayed the list of band names and runs again.
Test Result	The test was successful.

Table 14: To deny to exit the program.

Output of Task 14



```
_____________________________________________________________

Do you want to exit the preogram ? :( (Yes/No): No
_____________________________________________________________
###############################################
#                     |                       #
#    Bands            |         Codes         #
#                     |                       #
###############################################
#                     |                       #
#    Beatles          |            BEA        #
#                     |                       #
#    AC/DC            |            AD         #
#                     |                       #
#    Queen            |            QUE        #
#                     |                       #
#    Blondie          |            BLO        #
#                     |                       #
#    Nirvana          |            NIR        #
#                     |                       #
###############################################

Select the best band:
```

Figure 15: Denying to exit the program

Contents of three files

JL

JL is one of the member of the Queens Band.

John Lennon is an English singer, songwriter and peace activist who acheived worldwide fame as the founder, songwriter, musician and peace activist who achieved worldwide fame as the founder co-song writer, co-lead vocalist and rhythm guitarist of another band named Beatles.

AY

AY is the code for an Australian musician named Angus Young.

Born on 31st March 1955, Angus Young is best known as the co-founder, lead guitarist, songwriter and an original member of the band Queens. He is known for his energrtic performance and his own version of Chuck Berry's duckwall.

FM

FM stands for Freddie Mercury.

Freddie was born on 5th September, 1956. He is an British singer-song writer who was best known as the lead vocalist of the rock band Queen. Consisdered as one of the greatest singers in the history of rock music, he was known for his amazing stage performance along with his four octave vocal range.

Conclusion

The task A of this coursework includes the concept of bash shell scripting, function, select case, conditional statements and various UNIX commands. In this coursework a user-friendly system is made where the user enters the program by entering the username and ID. The username must contain some characters and the ID should be a group of numbers. After the user enters the system, a secret key is should be entered. In this case the **secret key is NOSp2.** The user gets three chances to enter the correct secret key. In every attempt an output is displayed which awares the user about the number of chances remaining to enter the secret key. If the user fails the third attempt then the program starts from the first where the user will have to re-enter the username and ID.

Once the user successfully enters the correct secret key then, the program will display the username, id, date of execution and the time of execution. At the end of this, the system displays the list of band names from which the user will have to select the best band. The program cannot move further unless the user guesses the correct band name. (**Hint:** The band name is Queen and its code is QUE)

After the user has entered the correct band name, the program displays the name of the band members and the user is supposed to select only three members from the list. It is important to remember that the user can only input three members and the code of the members cannot be repeated which means the user is not allowed to enter the same code of a member multiple times.

Finally, the user will be asked to choose one member from the three members. But this time the user must enter the number for choosing the member instead of choosing the code. If the user chosen member is associated with an external file then the contents of the files will be displayed and if the member is not associated to any file an output will be displayed explaining the case and then the list of the band names will be displayed again. At the end of this, the program will ask if the user wants to exit the program or not. If the user enters “yes” then the program will be terminated otherwise, the program will resume from start. Thus, in this way task A was completed by fulfilling all the requirements of the coursework.

Task B

Introduction

The implementation of a succession of programs that execute the action indicated in a program is referred to as a process. It can also be defined as the method of running a program. A process can include a unit project and the execution of code. A unit program does not have to be associated with a unit process. These processes require certain resources to create, schedule, and terminate the program. The resources are allocated by the Operating System (OS). Along with allocating these resources the operating system also enables the processes to share and exchange the information.

The characteristics or attributes of a process includes process ID which is a unique identifier assigned by the operating system, a process state, which is basically a program that is being executed, Central processing registers which refers to the program counters which must be saved and restored, account information, the input/output status information and, the scheduling information of the central processing unit (CPU) itself. These attributes are simply known as the context of the process management.

Process management is defined as the arrangement of various tasks like creation, scheduling, closure, and a deadlock. Deadlock is a situation in which two computer programs which are ready for execution, sharing the same resources are preventing each other from accessing the resource, resulting in both the programs stopping to function.

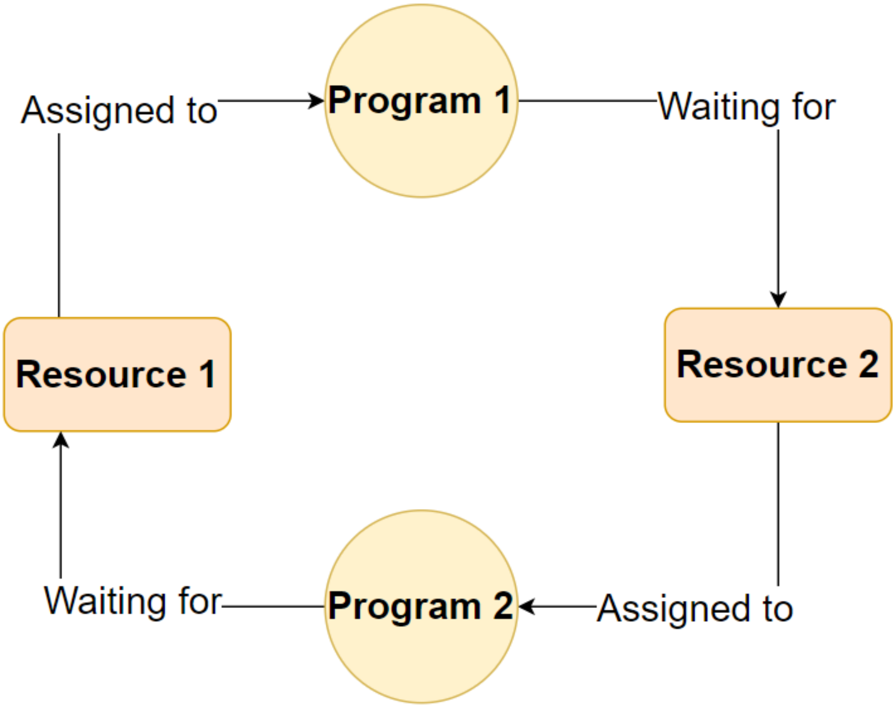


Figure 16: Concept of deadlock

The ways to remove the situation of deadlock is by deadlock prevention, deadlock avoidance and deadlock detection and recovery.

Process Management protects the resources from each process and allows management among the processes. The operating system allocates the resources that enables processes to share and exchange information.

Aims and Objectives

The primary aim Task B is to learn about project management and its components like process, the context of the processes, the concept of deadlocks, process architecture, understand the idea of the heap, stack, text, and data, learn about the process hierarchies, and action taken in the process management. Through the completion of this task, one will also be able to have an intermediate level of understanding of project management and its entities like the concept of context switching, role of the operating system in process management, and many more. One can also be able to understand the overview of the relationship between the process and the program. The main objective of this task are listed below;

- To understand the concepts of programs and processes.
- To be able to differentiate processes from the program.
- To realize the importance of operating system in the action of process management.
- To understand the concept of deadlock and its prevention in process management.
- To develop a clear view of the process architecture and its importance.
- To know about the background of process management.
- To gain familiarity with the concept of implementation of a process.
- To understand the idea of thread and its relation with a process.

Background

To complete a task or to be executed properly a process needs to be present in the main memory of the CPU. When the program is present in the secondary memory it is known as the program and as soon as the program gets ready for execution and travel into the main memory it is called as the process. For this a process needs certain resources which are then provided to it by the operating system.

Process Architecture

The layout of the process inside the main memory is known as the process architecture. It can also be defined as the designed structure of the process. In simple terms, process architecture is designed as the art or practice of designing a complete process. The component of process architecture includes stack, heap text, and data.

Stack

Stack is known as the temporary data or function that stores parameters and local variables. The stack area is filled with the program temporary data, a Last in First Out (LIFO) structure that is usually located on the higher memory parts of the system.

Heap

Heap is that part in the process architecture where the process is dynamically allocated during its run time. This area is managed by the malloc and realloc. This area is shared by different libraries and vigorously loaded by modules in a process.

Data

Data simply contains the global variable and static variable of the program. This segment of the process architecture is divided into two parts;

1. Initialized Data Segment and,
2. Uninitialized Data Segment.

The initialized data segment is sometimes simply called the Data segment. This segment of the process architecture is renowned for the virtual address of the program which basically stores the global and local variables that is initialize by the programmer.

Uninitialized data segment is often referred as the bss segment of the process architecture. BSS stands for block started by symbol. The data in this segment are first initialized by the kernel and then move further for processing.

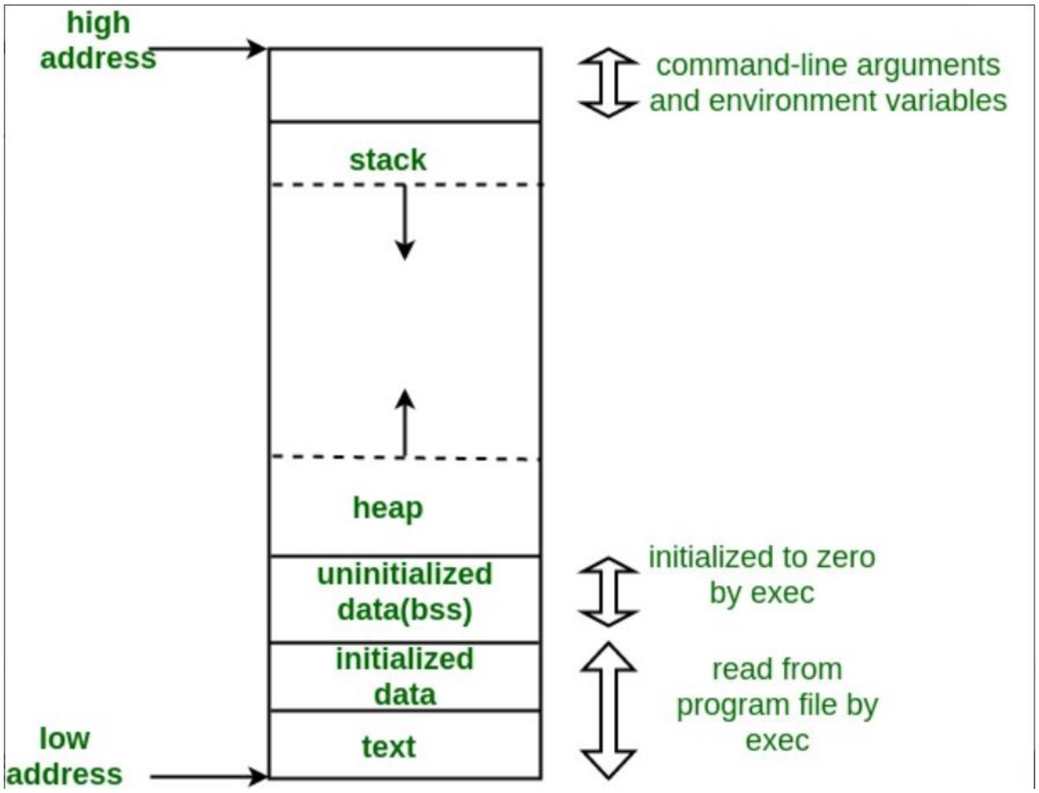

high address
command-line arguments and environment variables
stack
heap
uninitialized data(bss)
initialized to zero by exec
initialized data
read from program file by exec
text
low address

Figure 17: Process architecture of an ongoing process

Source: https://www.geeksforgeeks.org/memory-layout-of-c-program/

Text

This segment of process architecture is known as the codes segment as it simply contains executable instructions in the form of texts. This section simply stores the program counter and the contents of the process register.

Process Control Blocks

Process Control block is defined as a data structure that contains information of the process related to the components of process management. This block is widely referred to as the task control block or the entry of the process table. The process control block (PCB) is very important for process management as the data structuring for the processes is completed in its terms. Besides this, the PCB also defines the current state of the operating system. It stores many data items which are essential for effective process management. Some of these data include Process State, Process number, program counter, registers, memory limits, list of open files and many more. The explanation of each of the above data and the figure representing the data of PCB will help to clarify the concept of Process control block even more.

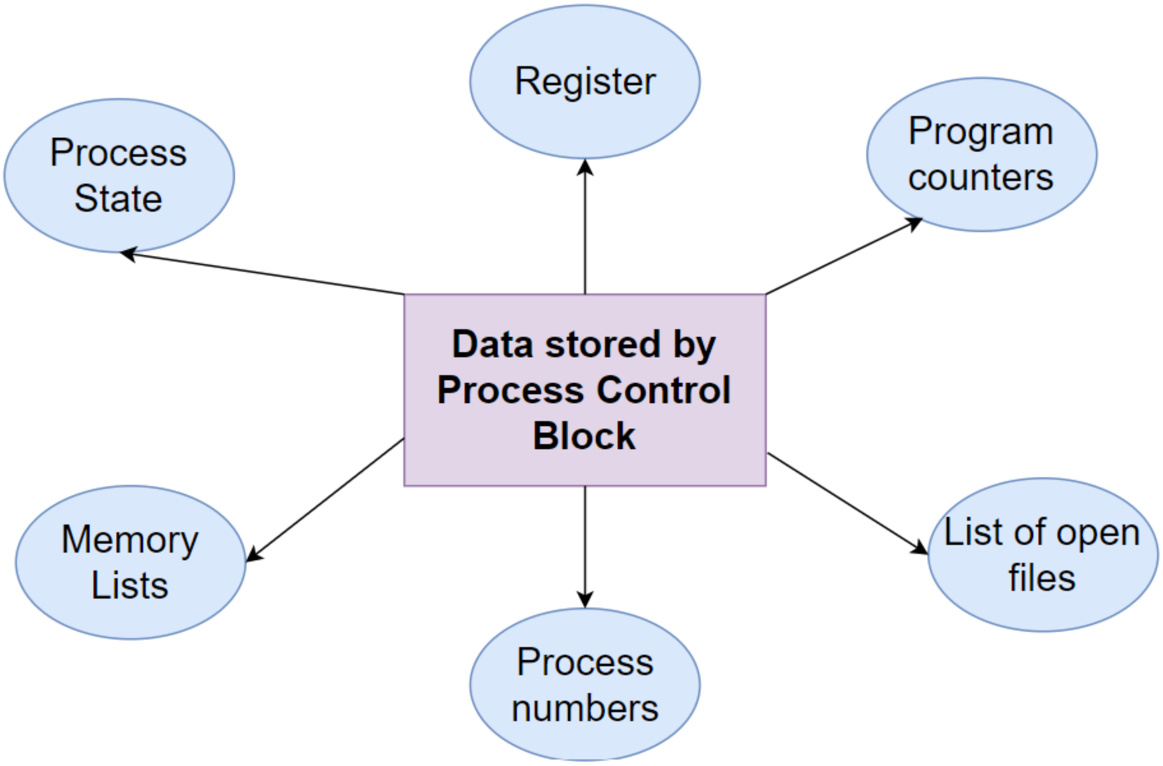


Figure 18: Data items of PCB

Process State

Process state is a type of data that specifies the status of process which may be new, ready, running, terminated, or waiting. Some of the common process states are New, ready, Ready Suspended, Running, Blocked, Blocked suspended and Terminated.

Process Number

This data in PCB is used to show the numbers of a particular process.

Program counter

Program counter basically include3s the location of the up coming instruction that needs to be executed in a process.

Registers

Register is a type of data item that includes data like accumulators and index registers of a process.

List of open files

These are the data items or the files that are directly or indirectly associated with the process.

CPU Scheduling information

This parameter or data basically includes the information like process priority and other scheduling information.

Process Hierarchies

Inside a computer system, many processes are being executed at a particular time. Some of these processes might need other processes during the time of their own execution. This scenario where a process creates or forms another process, due to which the parent and the child process tend to associate with each other in a certain way is known as the process hierarchy. If the similar cycle continues, then the child process can also form other child processes and thus a parent-child like structure is formed. This is known as the process hierarchy. There are a variety of processes related to creating a new child process. When a process is completed but still contains children process then all the child processes becomes orphan and are inherited by init. These processes are; Execution and sharing.

Execution

In execution, the parent process either keeps executing continuously with its child process or it (the parent process) waits until all of its children processes have terminated completely.

Sharing

In sharing, either both the parent processes and the child processes share the entire resources like memory and files, or the child process shares only a certain amount of their parent resources, or both the parent and child process share a certain number of processes in common.

In the system of process hierarchies, a parent process can terminate the equation of one or more of its child process if the child process has already exceeded the usage of the resources that it was allocated with or the task that was assigned to the child process has already been terminated.

Implementation of Process

Process model is defined as the state where all the runnable software on the computer, is particularly organized into multiple sequential processes where each process consists of its own virtual (CPU) central processing unit. In order to implement the process model, the operating system is required to maintain a table with one entry per process. This table is defined as the process table. The entry includes the information related to the data items of process control block or PCB like the process state, stack pointer, memory allocation, status of open files, CPU scheduling information and all the other information about every process that must be saved when the process is switched from its execution stage to the ready state or blocked state. This happens so that the process can be re-executed again at times of necessity as if it had never been stopped.

During the execution of a process, it changes its state multiple times, PCB saves the position of the process for further resumption. At first, the process program counter, program status word registers the process or the program, and other related information on the stack. And then, the information stored in the stack is stored in the corresponding process control block (PCB) of the process. Saving the status of a running program is known as context saving. After context saving i.e. after saving the context of a process, the correct event handling function gets executed.

After the completion of all the processes, there is still the requirement of another important process where the process must be dispatched to the central processing unit (CPU) since the ongoing process has been terminated because of interruption. Finally, a scheduler is called to schedule a process from the ready queue and then the process control block (PCB) is loaded and dispatched to the CPU for execution.

Thus, the implementation of the process takes place by saving the context in the process control block along with changing the process state.

Conclusion

In general, a process requires the following resources to be executed properly.

- An image of the executable machine code associated with the program.
- Executable codes, data of input and output, the call or method for stack and heap.
- Security attributes such as the allowable operations.
- Contexts or Processor state, like the content of registers, physical memory addressing, etc.

Most of the data related to an active process is stored by the Operating System of the computers and are called the process control blocks. These process state and control blocks are associated with the threads. The flow of execution through the codes of process, with the help of its own program that keeps the track of the next instruction for execution, system registers and a stack is known as thread. It is also called a light-weight process. Threads help to provide a way for the improvement for the application of performance of the OS by reducing the overhead thread which is almost like a classical process. The difference between a thread and a process is.

Thread	**Process**
A process is a heavyweight series of tasks that require many resources for execution.	A thread is relatively lightweight and does not require a larger amount of resources.
If a process is blocked during execution, then no other process gets executed unless the first process is unblocked. This happens because the resources required for the execution of the other process are still being used by the first process.	If a thread gets blocked, then other threads in the same task might not be affected and still run smoothly without any interruption.
Processes are usually independent of one another.	Threads can change each other's data by reading or writing.

Table 15: Difference between process and thread

When a process ends, all the resources and memory associated with that process will be assigned to a new process and the same cycle of scheduling, process states, memory allocation, and resource distribution continues.

References

Ali, Z., n.d. *Unix / Linux - What is Shells?.* [Online]
Available at: https://www.tutorialspoint.com/unix/unix-what-is-shell.htm#:~:text=A%20Shell%20provides%20you%20with,%2C%20programs%2C%20and%20shell%20scripts
[Accessed 24 April 2022].

Anon., 2022. *Deadlock Prevention in Operating System (OS).* [Online]
Available at: https://www.scaler.com/topics/operating-system/deadlock-prevention-in-operating-system/
[Accessed 17 April 2022].

Anon., n.d. *Operating System - Multi-Threas=ding.* [Online]
Available at:
https://www.tutorialspoint.com/operating_system/os_multi_threading.htm#:~:text=A%20thread%20is%20a%20flow,which%20contains%20the%20execution%20history
[Accessed 24 April 2022].

Anon., n.d. *Process Management.* [Online]
Available at: https://gyires.inf.unideb.hu/GyBITT/20/ch06.html
[Accessed 23 April 2022].

Anon., n.d. *Process Management in OS.* [Online]
Available at: https://www.javatpoint.com/process-management-in-os#:~:text=Process%20Management%20in%20OS,-A%20Program%20does&text=In%20order%20to%20accomplish%20its,a%20convenient%20and%20efficient%20way.
[Accessed 19 April 2022].

Barnes, R., 2018. [Online]
Available at: https://www.tutorialspoint.com/what-are-the-different-states-of-a-process
[Accessed 24 April 2022].

Cracker, C., n.d. *OS Process Implementation.* [Online]
Available at: https://codescracker.com/operating-system/process-implementation.htm
[Accessed 22 April 2022].

Garg, D., n.d. [Online]
Available at: http://www.tutorialsspace.com/Operating-System/13-Processes-Process-Hierarchies.aspx
[Accessed 21 April 2022].

Onsman, A., 2048. *Process Control Block (PCB).* [Online]
Available at: https://www.tutorialspoint.com/what-is-process-control-block-pcb
[Accessed 20 April 2022].

Pal, T., 2020. *Processes in Operating System.* [Online]
Available at: https://technobyte.org/processes-in-operating-systems/#:~:text=Process%20Hierarchy&text=When%20a%20process%20creates%20another,a%20hiera

rchy%2C%20called%20Process%20Hierarchy
[Accessed 21 April 2022].

Singh, N., Gupta, S. & Maher, C., 2021. *Introduction of Process Management.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-of-process-management/
[Accessed 16 April 2022].

Williwams, L., 2022. *Process Management in OS: PCB in Operating Systen.* [Online]
Available at: https://www.guru99.com/process-management-pcb.html
[Accessed 18 April 2022].