Module Code & Module Title

CS5001NA Networks and Operating System

Assignment 3

Assessment Weightage & Type

20% Individual Coursework

Year and Semester

2018-19 Spring

Student Name: Rajat Shrestha

London Met ID: 17030954

College ID: np01cp4a170021

Assignment Due Date: 15th April 2019

Assignment Submission Date: 15th April 2019

Word Count (Part B): 1571

Table of Contents

Abstract **1**

Task A **2**

1. Introduction **2**

2. Script **2**

3. Contents of the player files **8**

4. Testing **9**

5. Conclusion **23**

Task B **24**

1. Introduction **24**

1.1. Aims 24

1.2. Objectives 24

2. Body **25**

2.1. Categories of I/O Devices 25

2.2. Device Drivers and Device Controllers 26

2.3. I/O subsystem 27

2.4. DMA 28

3. Conclusion **29**

References **30**

Appendix **32**

Table of Figures:

Figure 1: Running the script without any parameters.......... 9
Figure 2: Running the script with valid parameters but incorrect password 10
Figure 3: Entering the correct password.......... 11
Figure 4: Invalid input on country field.......... 12
Figure 5: Entering the wrong country code.......... 13
Figure 6: Valid country code Input.......... 14
Figure 7: Duplicate player code handling 15
Figure 8: Invalid input on player codes.......... 16
Figure 9: More than 3 player codes as input 17
Figure 10: Inputting valid players 18
Figure 11: User input handling on the favorite player chooser... 19
Figure 12: Favorite player with no description file.......... 20
Figure 13: Starting over.......... 21
Figure 14: Terminating.......... 22
Figure 15: Bus connecting vital parts of a Computer System (Sexton, 1996) 28
Figure 16: Communication in UNIX (TekSlate, 2019).......... 35

Table of tables:

Table 1: Running the script without any parameters 9
Table 2: Running the script with valid parameters but incorrect password 10
Table 3: Entering correct password 11
Table 4: Invalid input on country field 12
Table 5: Entering the wrong country code 13
Table 6: Valid country code Input 14
Table 7: Duplicate player code handling 15
Table 8: Invalid input on player codes 16
Table 9: More than 3 player codes as input 17
Table 10: Inputting valid players 18
Table 11: User input handling on the favorite player chooser. 19
Table 12: Favorite player with no description file 20
Table 13: Starting over 21
Table 14: Terminating 22

Abstract

This third coursework of the Networking and Operating System Module consists of two parts, Task A and task B. Task A is a report based on creating a Bash Script which includes the specification of the project, script, files, testing and then the conclusion of the report. Task B consists of a research-based report which goes in-depth of the UNIX as an Operating System. It details how the UNIX is used as a network Operating System, the I/O Operations on the UNIX and the Communication of the Users and the peripherals on the UNIX. This is done by doing extensive research on that matter and the contributors for the essential information provided are all referenced at the end.

Task A

1. Introduction

This part of the report consists of a Script file being developed which is designed to run on Bourne Again Shell to various tasks as specified. The specified tasks were completed and the resulting script file was tested extensively as a guide and a proof of the mechanism of the script file. The shell scripting used in this course work is based on the Bourn Again Shell in Linux which is derived from the Bourne shell developed by Stephen Bourne in the 7th version of UNIX developed in bell labs (Shotts Jr., 2019). However, to develop a script which can do the listed tasks various programming methods were utilized such as if-else statements, case statements, passing parameters, while loops, until loops, functions and many more. This whole process is documented and presented in this report, the code, testing, and contents are presented below.

2. Script

```
#!/bin/sh

#1

if [[ $# != 2 ]]; then

  echo -e "Error! [0] Username or id missing"

  echo -e "please execute again with both username and ID"

  exit

fi

if ! [[ "$2" =~ ^[+-]?[0-9]+$ ]]; then

  echo -e "Error! [1] enter ID as a number"

  exit

fi

username=$1

id=$2
```

```
echo -e "------------------------------------------------------------"
echo -e "\t\t\t Welcome \n"
#2
echo -e "Enter secret key!"
password=noway
key=a
count=0
until [[ "$key" = "$password" ]]; do
  if [[ count -lt 3 ]]; then
    if [[ count -gt 0 ]]; then
      echo -e "Error! [2] incorrect password attempt $count"
    fi
    echo -e "Password: \c"
    read key
    count=`expr $count + 1`
    else
      echo -e "Error! [3] login attempts exceeded"
      echo -e "\nprogram terminated\n\t***"
      exit
  fi
done
#3
echo -e "\v************************************************************"
echo -e "\t\t  login sucessfull \n"
echo -e "\tID: $id\t\t\t\t\tUsername: $username"
echo -e "\t\tAccess Date: \c"
date +%D
echo -e "\t\tAccess Time: \c"
date +%T
echo -e "*************************************************************"
echo -e "\t"
country(){
  #4
  echo -e "\n\t SELECT A COUNTRY"
  echo -e "____________________________________________________________"
  echo -e "| Country                    | code                        |"
  echo -e "------------------------------------------------------------"
  echo -e "| Japan                      | JPN                         |"
  echo -e "| Argentina                  | ARG                         |"
  echo -e "| Germany                    | GER                         |"
  echo -e "| France                     | FRA                         |"
  echo -e "| Brazil                     | BRZ                         |"
  echo -e "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^"
  #5
  echo -e "\nWhich is the best football team?:-\c"
  read wc
  if [[ "$wc" = "FRA" ]]; then
    echo -e "************************************************************"
    echo -e "\t\tFrance is the best team"
    echo -e "The France national football team represents France in international football "
    echo -e "and is controlled by the French Football Federation, also known as FFF, "
    echo -e "or in French: Fédération française de football. The team's colours are"
    echo -e "blue, white and red, and the coq gaulois its symbol. France are"
    echo -e "colloquially known as Les Bleus."
    echo -e "************************************************************"
  elif [[ "$wc" = "JPN" || "$wc" = "ARG" || "$wc" = "GER" || "$wc" = "BRZ" ]]; then
      echo -e "Error! [4] Wrong Country!"
      echo -e "$wc is not the world champion"
      country
  else
    echo -e "Error! [5] Invalid Input!"
    country
  fi
}
val(){
  # to check if the players are valid
  if [[ $1 != "KAG" && $1 != "MES" && $1 != "REU" && $1 != "LOR" && $1 != "COU" ]];
then
    echo -e "Error! [6] Player doesnt exist Try again"
    player_selection
  fi
}
numval(){
  # to check if there are exactly 3 inputs
  if [[ $# != 3 ]]; then
  echo -e "Error! [7] Invalid number of inputs, Enter only 3 players"
    player_selection
  fi
  if [[ $1 = $2 || $2 = $3 || $1 = $3  ]]; then
    echo -e "Error! [8] Duplicate inputs!"
    player_selection
  fi
}
player_selection(){
  #6
  a=a
  b=b
  c=c
  echo -e "The Five Star players at the world cup are:"
  echo -e "____________________________________________________________"
  echo -e "| Players                    | code                        |"
  echo -e "------------------------------------------------------------"
  echo -e "| Kagawa                     | KAG                         |"
  echo -e "| Messi                      | MES                         |"
  echo -e "| Reus                       | REU                         |"
  echo -e "| Loris                      | LOR                         |"
  echo -e "| Coutinho                   | COU                         |"
  echo -e "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^"
  #7
  echo -e "Choose three players represented by codes separated by spaces"
  #8
  read a b c
  numval $a $b $c
  val "$a"
  val "$b"
  val "$c"
  echo -e " "
}
fun(){
  PS3="Enter a number as per list for choosing a player you want:"
  select player in $a $b $c
    #10
    do
      if [ -z $player ]
        then
          echo -e "Error! [9] Invalid Input!"
      elif [ -r $player ]
        then
        #11
          echo " "
          cat $player
          break
      else
        echo -e "Error! [10] Cant find file!"
      fi
    done
}
execute(){
country
player_selection
fun
echo -e "Do you want to start over?(y):-\c"
read y
case $y in
 "YES"|"yes"|"Yes"|"Y"|"y")
   execute
   ;;
 *)
  echo "Program Terminated"
  exit
  ;;
esac
}
execute;
```

3. Contents of the player files

The contents of the description files of the players

KAG

```
Shinji Kagawa is a Japanese professional footballer who plays as a midfielder for Turkish club Beşiktaş, on loan from German club Borussia Dortmund and the Japan national team. Kagawa began his professional career in his homeland with Cerezo Osaka before joining Borussia Dortmund in 2010.
```

MES

```
Lionel Andrés Messi Cuccittini is an Argentine professional footballer who plays as a forward and captains both Spanish club Barcelona and the Argentina national team.
```

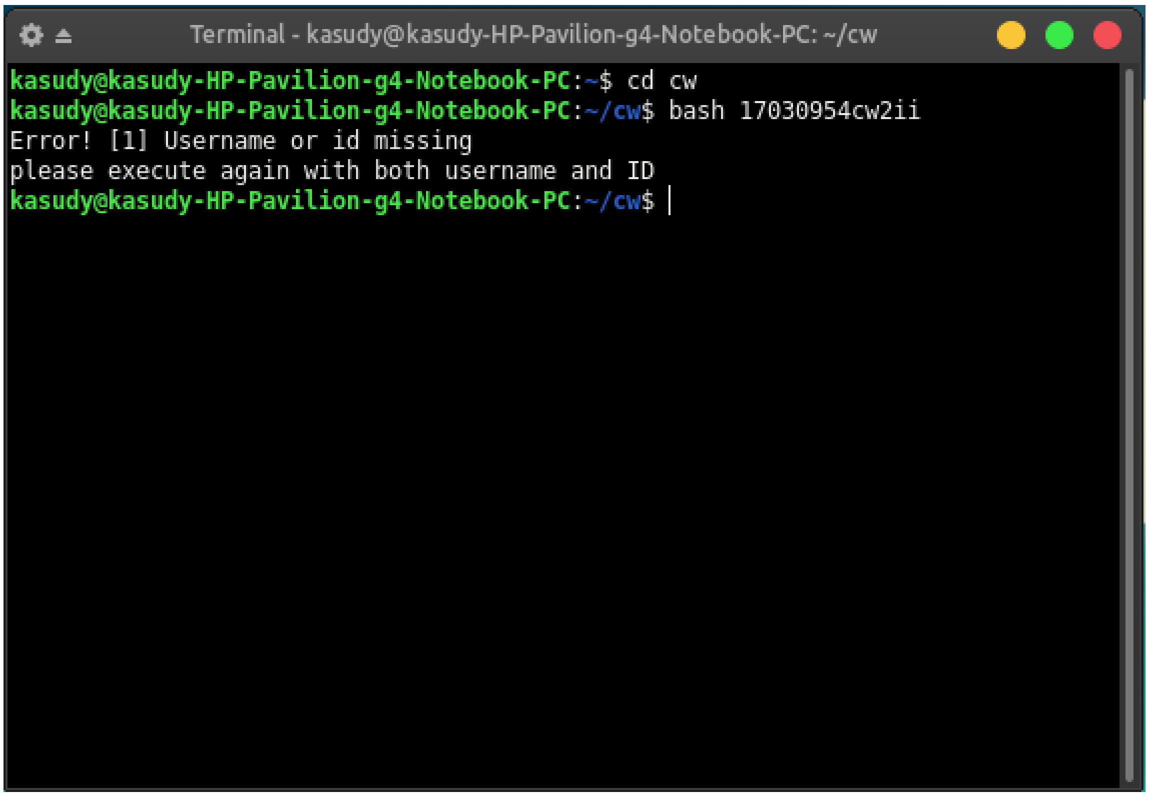
LOR

```
Hugo Hadrien Dominique Lloris is a French professional footballer who plays as a goalkeeper and is the captain of both English club Tottenham Hotspur and the French national team.
```

4. Testing

Table 1: Running the script without any parameters

Objective	Checking if the script file runs without parameters or not
Action	Running the script without any parameter
Expected Result	The Script file when run would show error and doesn't execute the program.
Actual Result	The Script file showed error and did not execute
Conclusion	Test Successful.

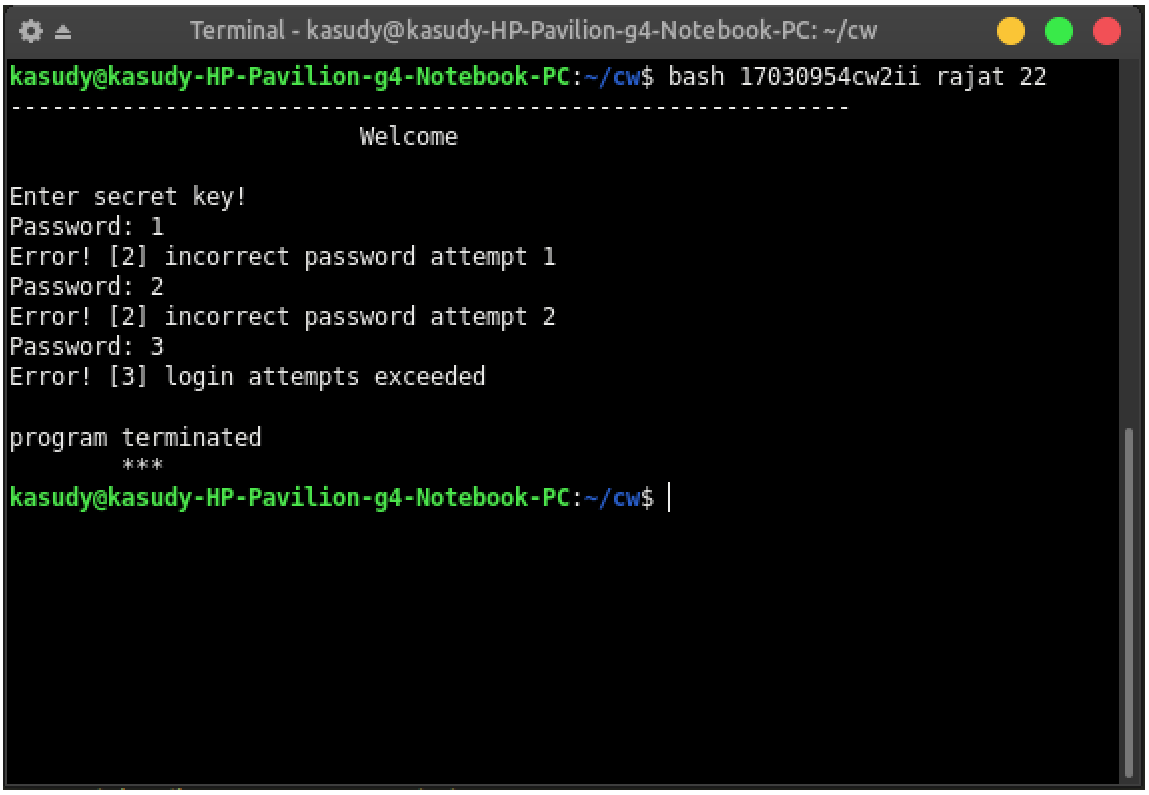


```
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~$ cd cw
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~/cw$ bash 17030954cw2ii
Error! [1] Username or id missing
please execute again with both username and ID
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~/cw$ 
```

Figure 1: Running the script without any parameters

Table 2: Running the script with valid parameters but incorrect password

Objective	Checking if the script file runs with parameters or not
Action	Running the script with Name and ID. Input incorrect password.
Expected Result	The script runs successfully but terminated due to failed password inputs.
Actual Result	The Script executed successfully but terminates as the incorrect password is been given as an input.
Conclusion	Test Successful.

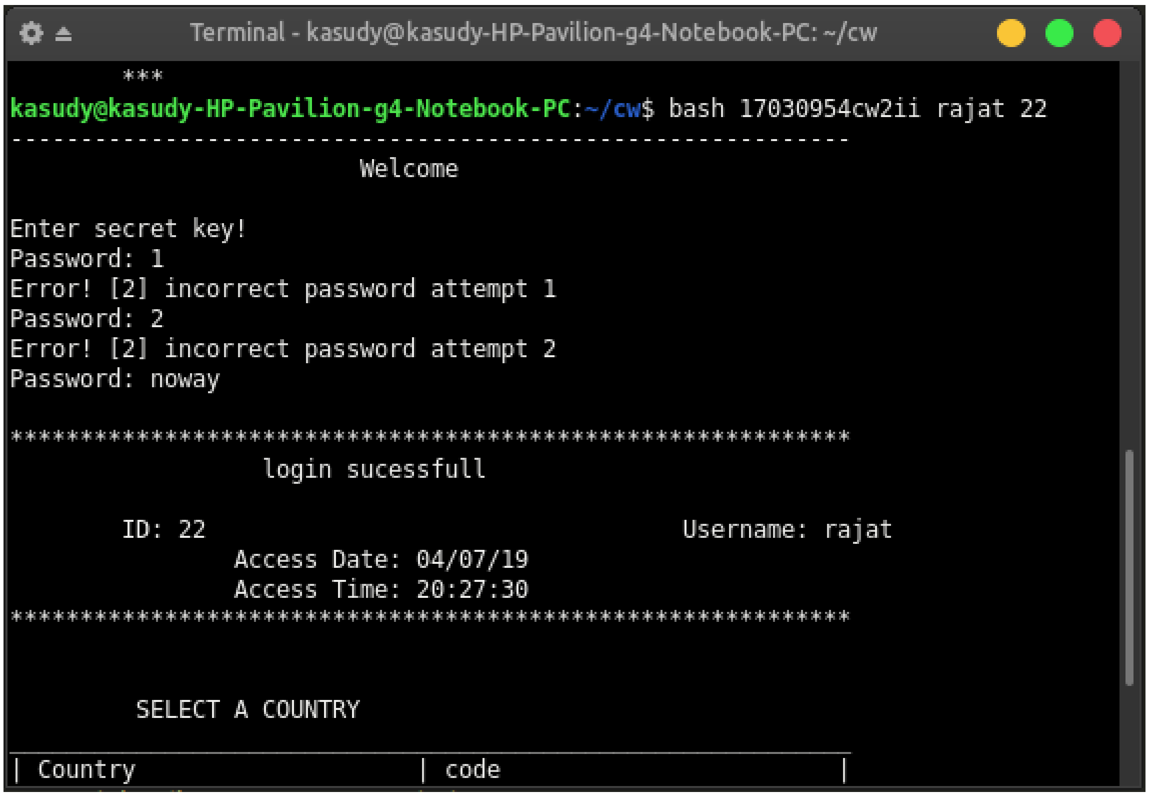


```
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~/cw$ bash 17030954cw2ii rajat 22
------------------------------------------------------------
                         Welcome

Enter secret key!
Password: 1
Error! [2] incorrect password attempt 1
Password: 2
Error! [2] incorrect password attempt 2
Password: 3
Error! [3] login attempts exceeded

program terminated
        ***
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~/cw$ 
```

Figure 2: Running the script with valid parameters but incorrect password

Table 3: Entering the correct password

Objective	Checking if the script file runs without parameters or not
Action	Running the script. Entering the Wrong password for the first two times. Entering the Correct password for the last attempt.
Expected Result	The Script would run after the correct password is entered
Actual Result	The Script file ran the next step after the correct password
Conclusion	Test Successful.

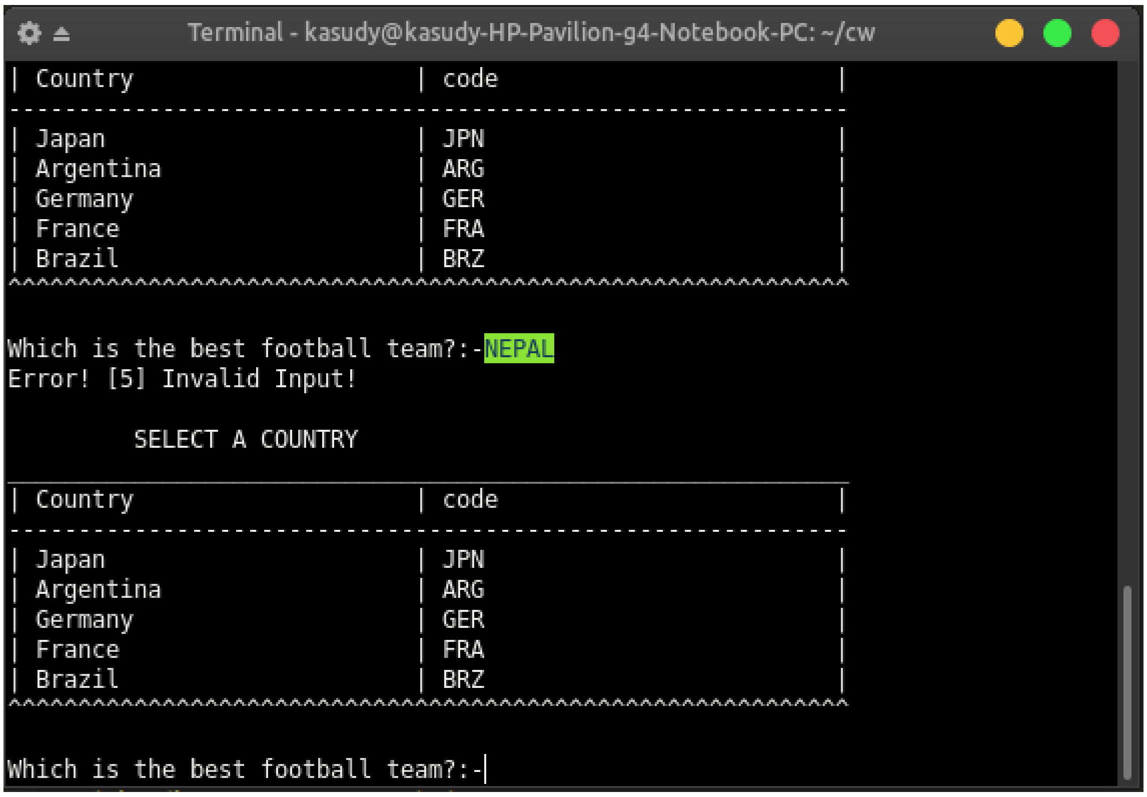


```
       ***
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~/cw$ bash 17030954cw2ii rajat 22
------------------------------------------------------------
                         Welcome

Enter secret key!
Password: 1
Error! [2] incorrect password attempt 1
Password: 2
Error! [2] incorrect password attempt 2
Password: noway

************************************************************
                  login sucessfull

        ID: 22                               Username: rajat
                Access Date: 04/07/19
                Access Time: 20:27:30
************************************************************


         SELECT A COUNTRY

| Country                     | code                         |
```

Figure 3: Entering the correct password

Table 4: Invalid input on country field

Objective	Checking how the country code input handles invalid country codes or other invalid inputs
Action	Inputting invalid inputs
Expected Result	The Script file when run would show error and then ask the user to enter the code again
Actual Result	The Script file showed error and looped as valid data wasn't entered.
Conclusion	Test Successful.

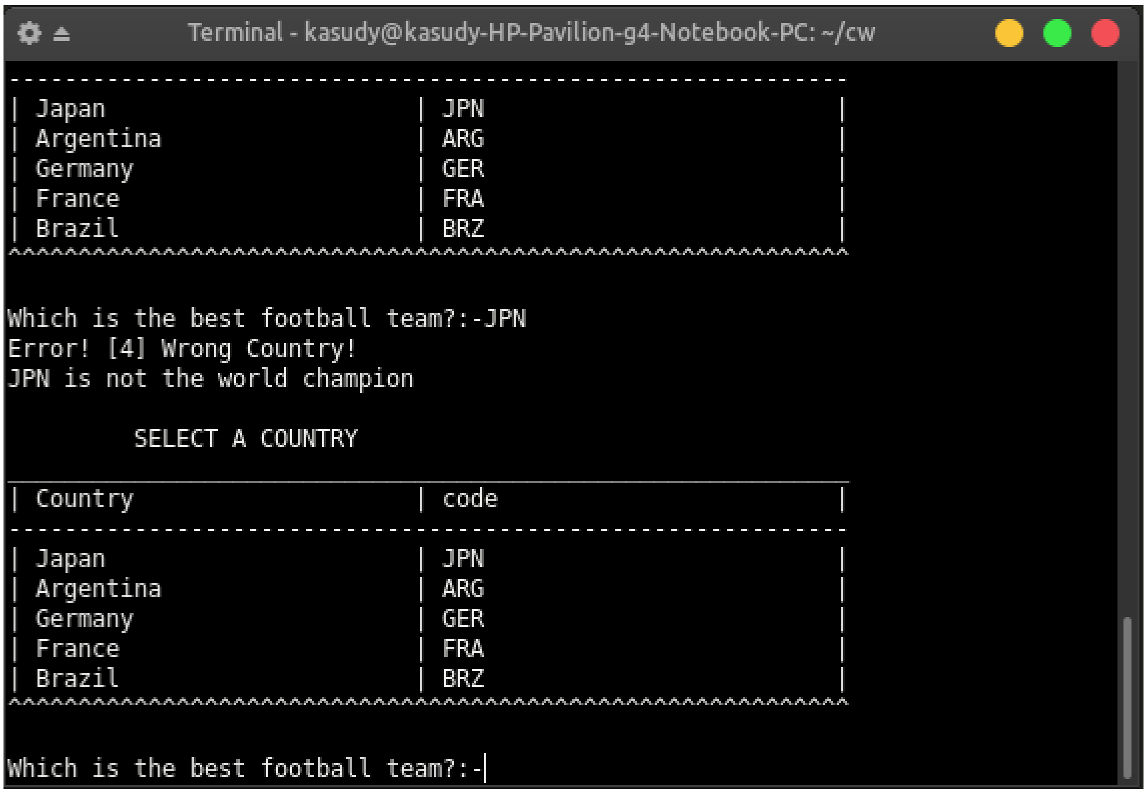


```
| Country                    | code                        |
-------------------------------------------------------------
| Japan                      | JPN                         |
| Argentina                  | ARG                         |
| Germany                    | GER                         |
| France                     | FRA                         |
| Brazil                     | BRZ                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Which is the best football team?:-NEPAL
Error! [5] Invalid Input!

         SELECT A COUNTRY
_____________________________________________________________
| Country                    | code                        |
-------------------------------------------------------------
| Japan                      | JPN                         |
| Argentina                  | ARG                         |
| Germany                    | GER                         |
| France                     | FRA                         |
| Brazil                     | BRZ                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Which is the best football team?:-
```

Figure 4: Invalid input on country field

Table 5: Entering the wrong country code

Objective	Checking how the country code input handles entry of other countries than the best ones
Action	Inputting other country code
Expected Result	The Script file when run would show error and then ask the user to enter the code again
Actual Result	The Script file showed error and looped until the correct code was entered
Conclusion	Test Successful.



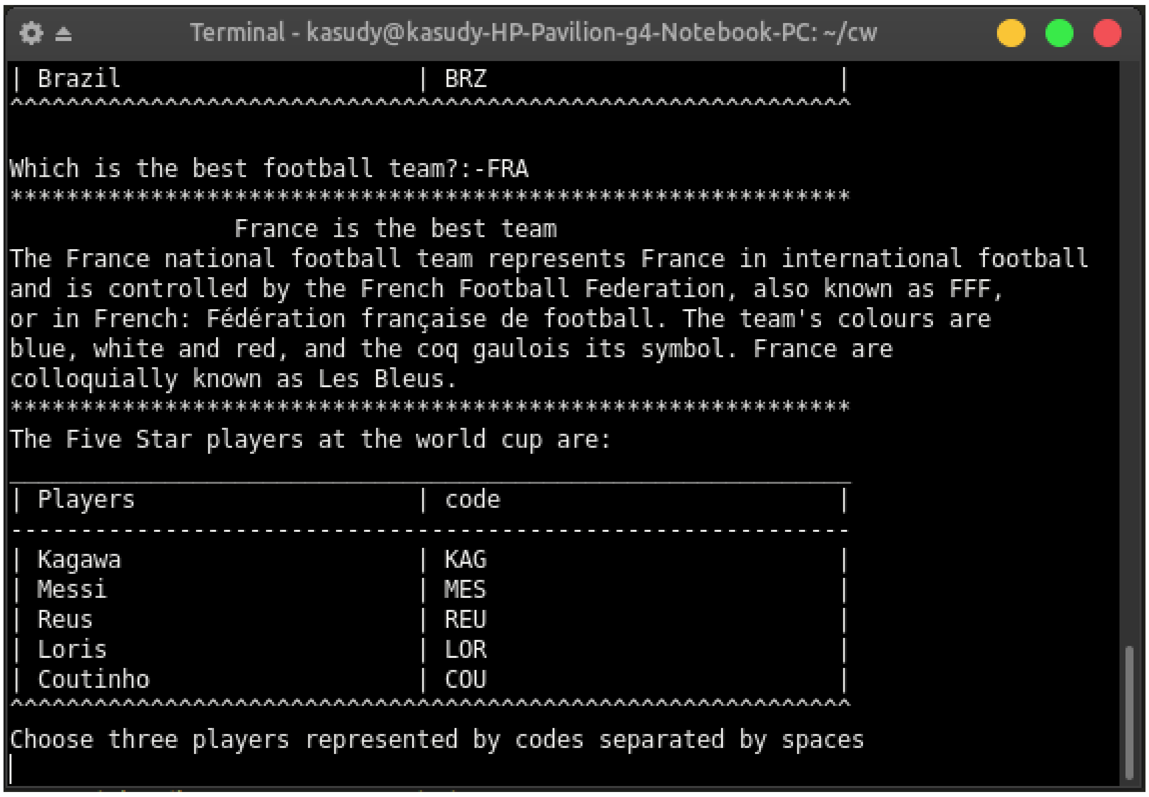
Terminal - kasudy@kasudy-HP-Pavilion-g4-Notebook-PC: ~/cw

```
------------------------------------------------------------
| Japan                         | JPN                        |
| Argentina                     | ARG                        |
| Germany                       | GER                        |
| France                        | FRA                        |
| Brazil                        | BRZ                        |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Which is the best football team?:-JPN
Error! [4] Wrong Country!
JPN is not the world champion

         SELECT A COUNTRY

____________________________________________________________
| Country                       | code                       |
------------------------------------------------------------
| Japan                         | JPN                        |
| Argentina                     | ARG                        |
| Germany                       | GER                        |
| France                        | FRA                        |
| Brazil                        | BRZ                        |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Which is the best football team?:-
```

Figure 5: Entering the wrong country code

Table 6: Valid country code Input

Objective	Checking how the country code input handles entry of a correct country
Action	Inputting correct country code
Expected Result	The Script file when run would show the details of the country and proceed to the next step
Actual Result	The Script file showed the details of the country and proceeded to the next step
Conclusion	Test Successful.

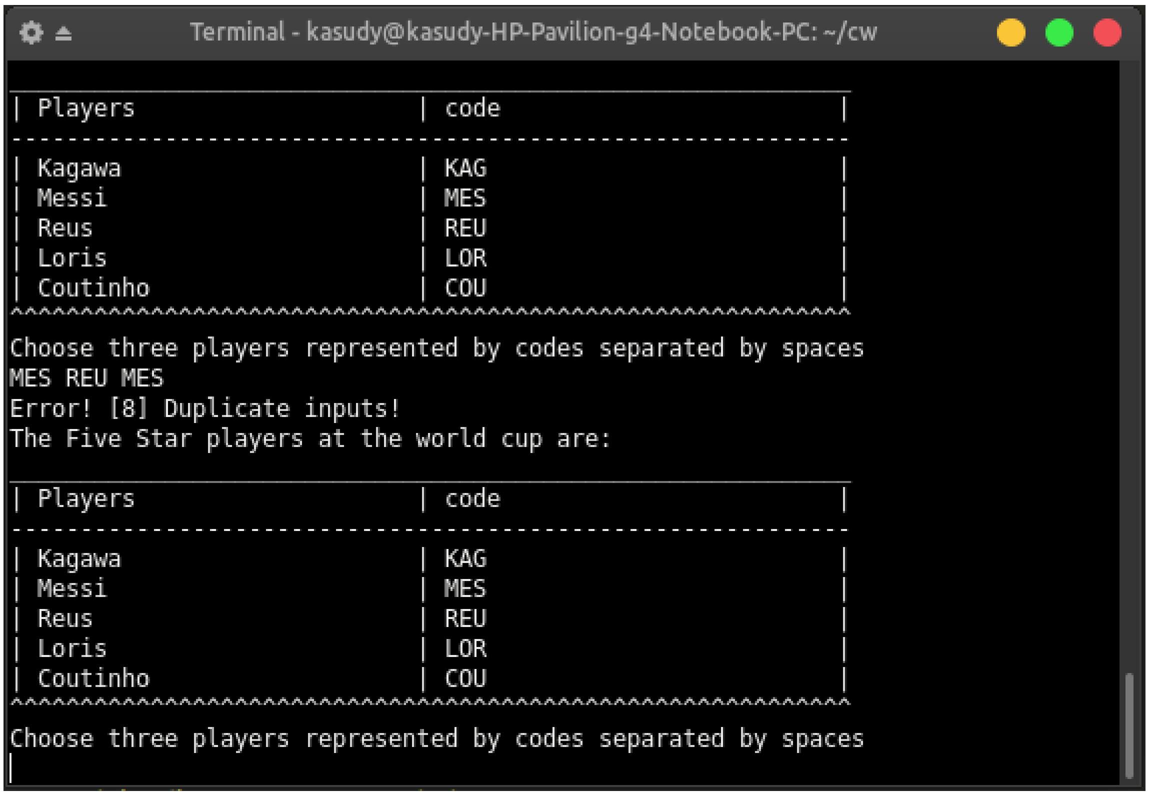


```
| Brazil                       | BRZ                          |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Which is the best football team?:-FRA
************************************************************
                France is the best team
The France national football team represents France in international football
and is controlled by the French Football Federation, also known as FFF,
or in French: Fédération française de football. The team's colours are
blue, white and red, and the coq gaulois its symbol. France are
colloquially known as Les Bleus.
************************************************************
The Five Star players at the world cup are:
____________________________________________________________
| Players                      | code                         |
------------------------------------------------------------
| Kagawa                       | KAG                          |
| Messi                        | MES                          |
| Reus                         | REU                          |
| Loris                        | LOR                          |
| Coutinho                     | COU                          |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
|
```

Figure 6: Valid country code Input

Table 7: Duplicate player code handling

Objective	Checking how the duplicate player inputs are handled
Action	Inputting other country code
Expected Result	The code will show an error and loop
Actual Result	The Script file showed error and looped.
Conclusion	Test Successful.

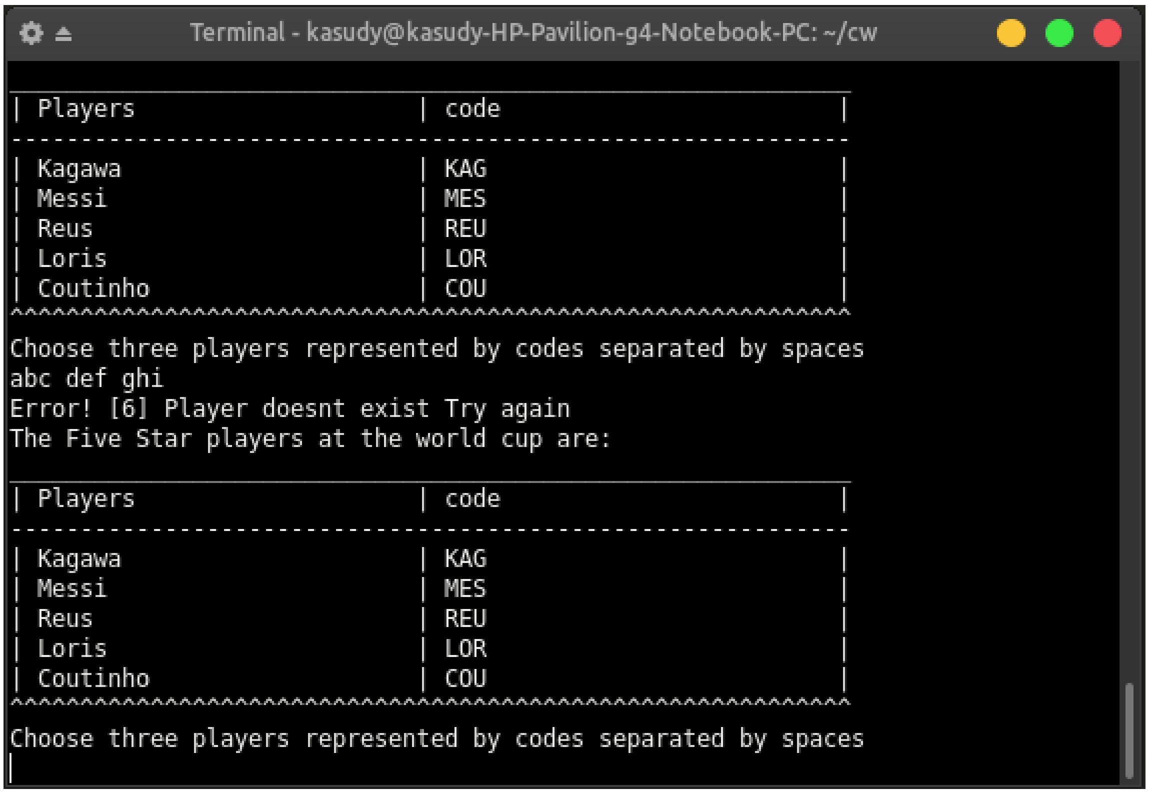


```
___________________________________________________________
| Players                     | code                        |
-----------------------------------------------------------
| Kagawa                      | KAG                         |
| Messi                       | MES                         |
| Reus                        | REU                         |
| Loris                       | LOR                         |
| Coutinho                    | COU                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
MES REU MES
Error! [8] Duplicate inputs!
The Five Star players at the world cup are:
___________________________________________________________
| Players                     | code                        |
-----------------------------------------------------------
| Kagawa                      | KAG                         |
| Messi                       | MES                         |
| Reus                        | REU                         |
| Loris                       | LOR                         |
| Coutinho                    | COU                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
|
```

Figure 7: Duplicate player code handling

Table 8: Invalid input on player codes

Objective	Checking how the invalid data are handled in the player code input
Action	Inputting invalid data
Expected Result	The code will show an error and loop
Actual Result	The Script file showed error and looped.
Conclusion	Test Successful.

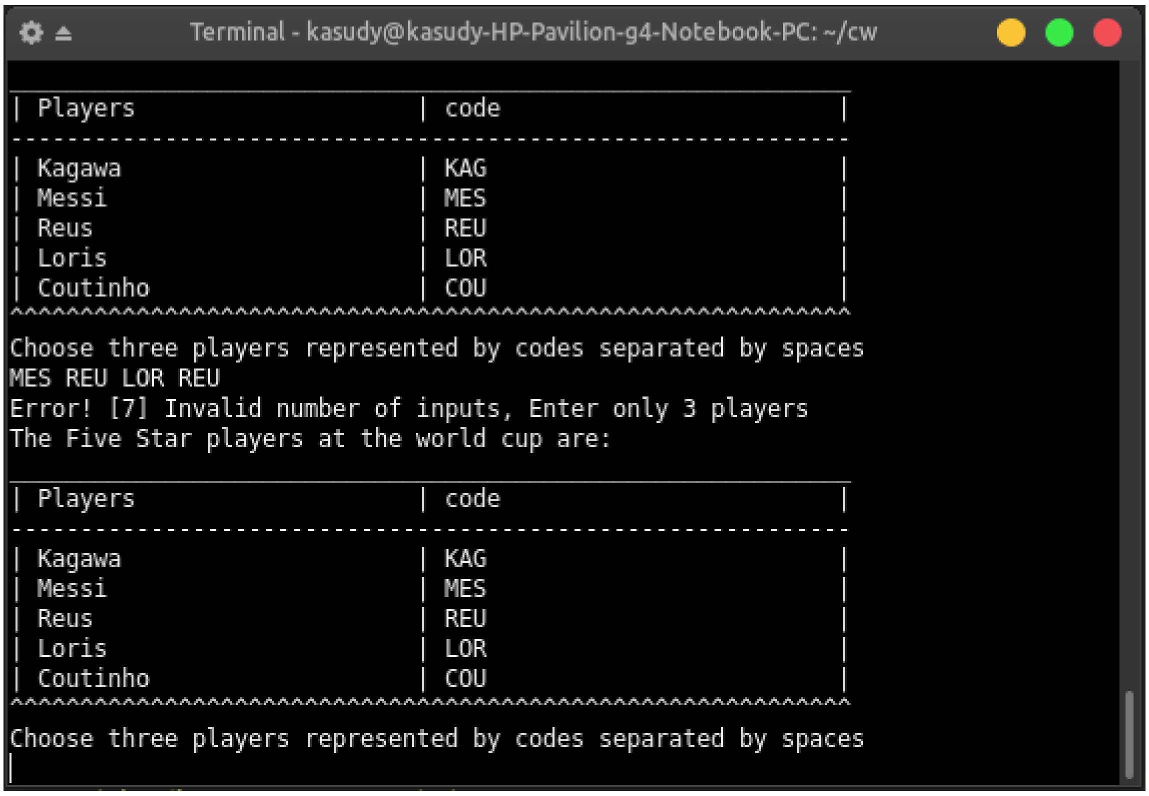


```
___________________________________________________________
| Players                     | code                        |
-----------------------------------------------------------
| Kagawa                      | KAG                         |
| Messi                       | MES                         |
| Reus                        | REU                         |
| Loris                       | LOR                         |
| Coutinho                    | COU                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
abc def ghi
Error! [6] Player doesnt exist Try again
The Five Star players at the world cup are:
___________________________________________________________
| Players                     | code                        |
-----------------------------------------------------------
| Kagawa                      | KAG                         |
| Messi                       | MES                         |
| Reus                        | REU                         |
| Loris                       | LOR                         |
| Coutinho                    | COU                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
|
```

Figure 8: Invalid input on player codes

Table 9: More than 3 player codes as input

Objective	Checking how more than 3 player codes are handled.
Action	Inputting more than 3 player codes
Expected Result	The code will show an error and loop
Actual Result	The Script file showed error and looped.
Conclusion	Test Successful.

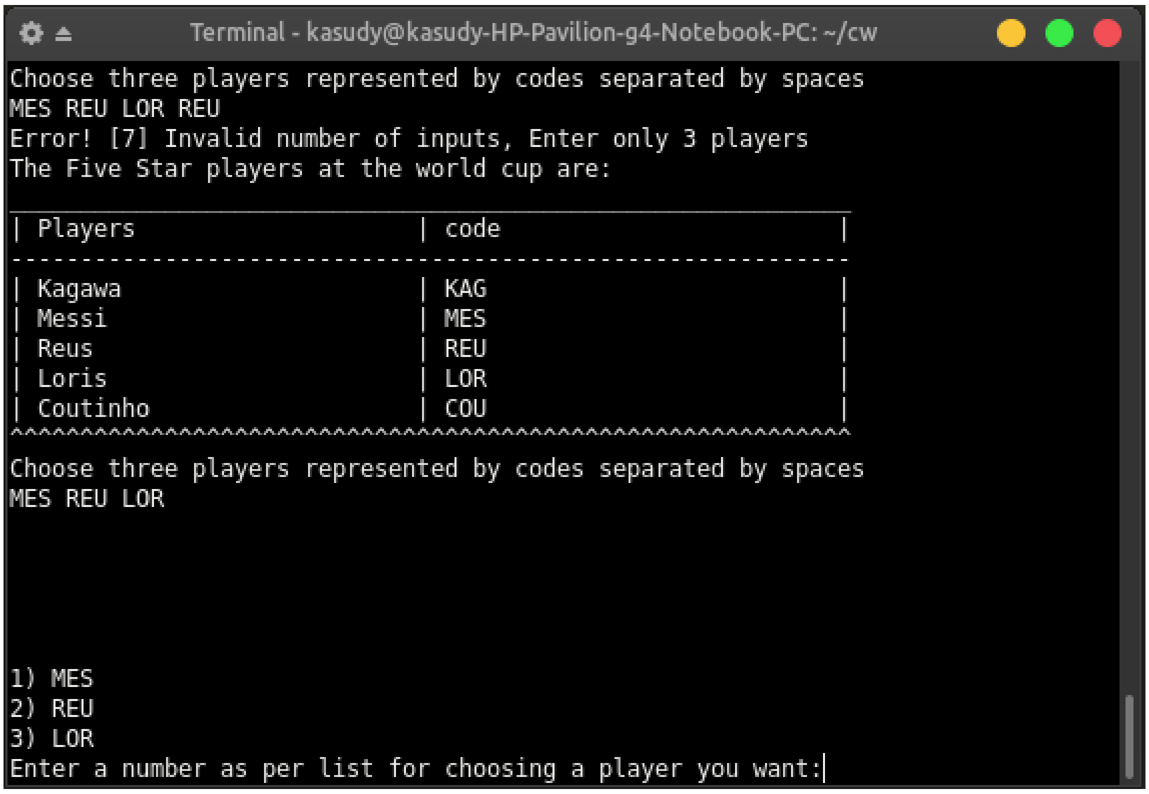


```
_____________________________________________________________
| Players                     | code                        |
-------------------------------------------------------------
| Kagawa                      | KAG                         |
| Messi                       | MES                         |
| Reus                        | REU                         |
| Loris                       | LOR                         |
| Coutinho                    | COU                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
MES REU LOR REU
Error! [7] Invalid number of inputs, Enter only 3 players
The Five Star players at the world cup are:
_____________________________________________________________
| Players                     | code                        |
-------------------------------------------------------------
| Kagawa                      | KAG                         |
| Messi                       | MES                         |
| Reus                        | REU                         |
| Loris                       | LOR                         |
| Coutinho                    | COU                         |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
|
```

Figure 9: More than 3 player codes as input

Table 10: Inputting valid players

Objective	Checking what the script does after three correct player code inputs
Action	Inputting three valid player codes
Expected Result	The Script file when run would run as intended and proceed to the next step
Actual Result	The Script file ran successfully and proceeded to the next step
Conclusion	Test Successful.

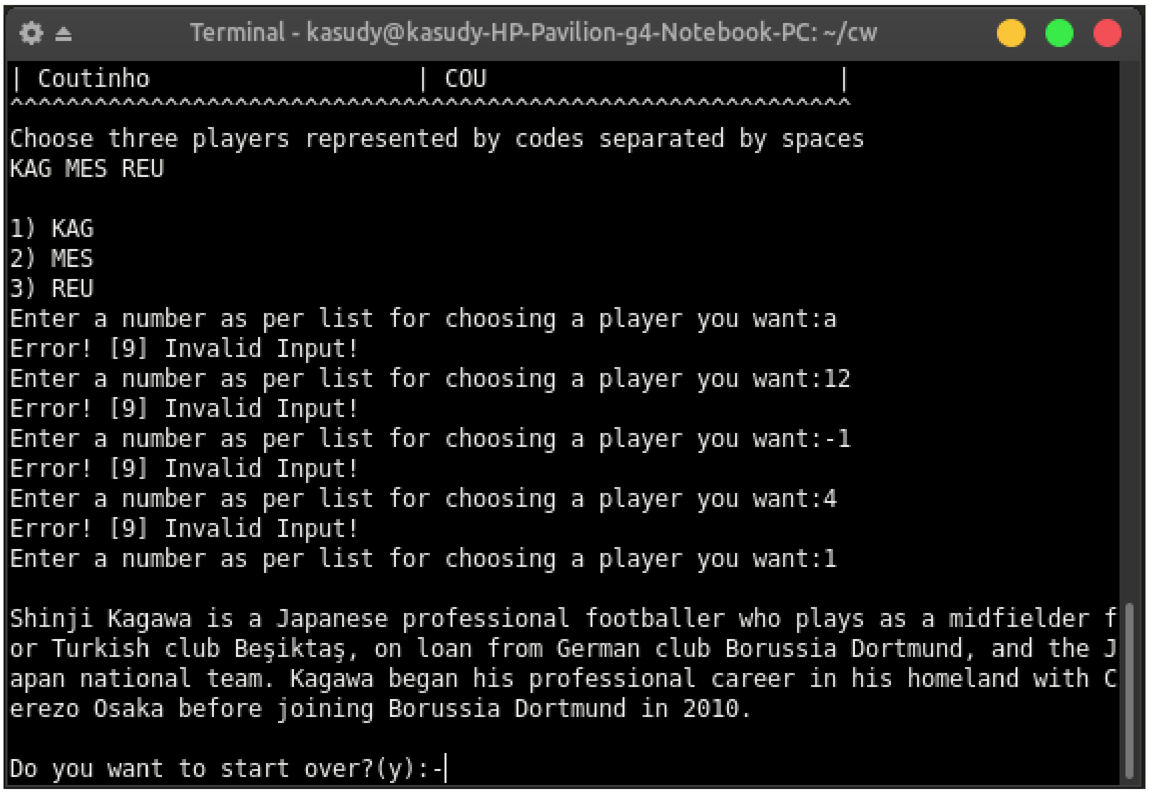


```
Terminal - kasudy@kasudy-HP-Pavilion-g4-Notebook-PC: ~/cw
Choose three players represented by codes separated by spaces
MES REU LOR REU
Error! [7] Invalid number of inputs, Enter only 3 players
The Five Star players at the world cup are:
____________________________________________________________
| Players                       | code                       |
------------------------------------------------------------
| Kagawa                        | KAG                        |
| Messi                         | MES                        |
| Reus                          | REU                        |
| Loris                         | LOR                        |
| Coutinho                      | COU                        |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
MES REU LOR





1) MES
2) REU
3) LOR
Enter a number as per list for choosing a player you want:
```

Figure 10: Inputting valid players

Table 11: User input handling on the favorite player chooser.

Objective	Checking how the script file handles invalid inputs on the favorite player chooser and if it runs if the correct code is entered
Action	Inputting invalid entries
Expected Result	The Script file when run would show error and then ask the user to enter the code again until a valid input is given
Actual Result	The Script file showed error and looped until the correct code was entered and proceeded to the next step
Conclusion	Test Successful.

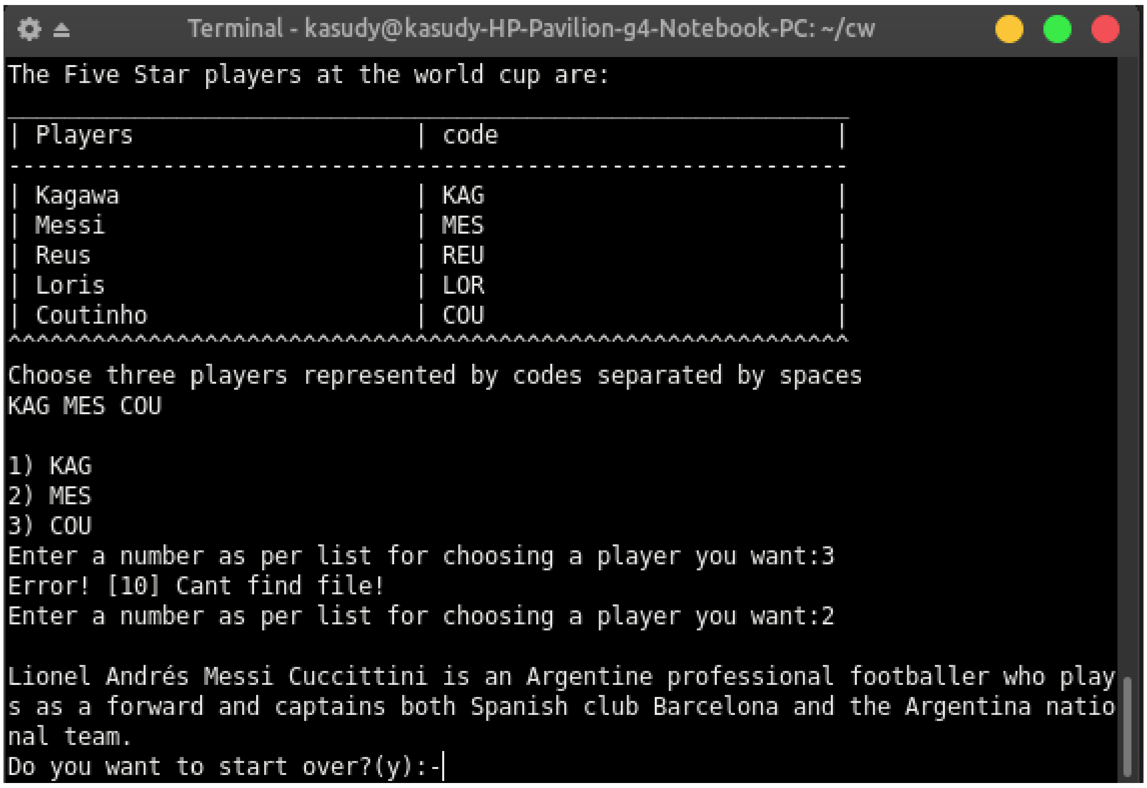


```
| Coutinho                        | COU                             |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
KAG MES REU

1) KAG
2) MES
3) REU
Enter a number as per list for choosing a player you want:a
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:12
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:-1
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:4
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:1

Shinji Kagawa is a Japanese professional footballer who plays as a midfielder f
or Turkish club Beşiktaş, on loan from German club Borussia Dortmund, and the J
apan national team. Kagawa began his professional career in his homeland with C
erezo Osaka before joining Borussia Dortmund in 2010.

Do you want to start over?(y):-
```

Figure 11: User input handling on the favorite player chooser.

Table 12: Favorite player with no description file

Objective	Checking how the script file handles the entry of valid favorite player with no file present for their description
Action	Inputting favorite player code whose description doesn't exist
Expected Result	The Script file when run would show error and then ask the user to enter the code again
Actual Result	The Script file showed error and looped until the correct code was entered
Conclusion	Test Successful.

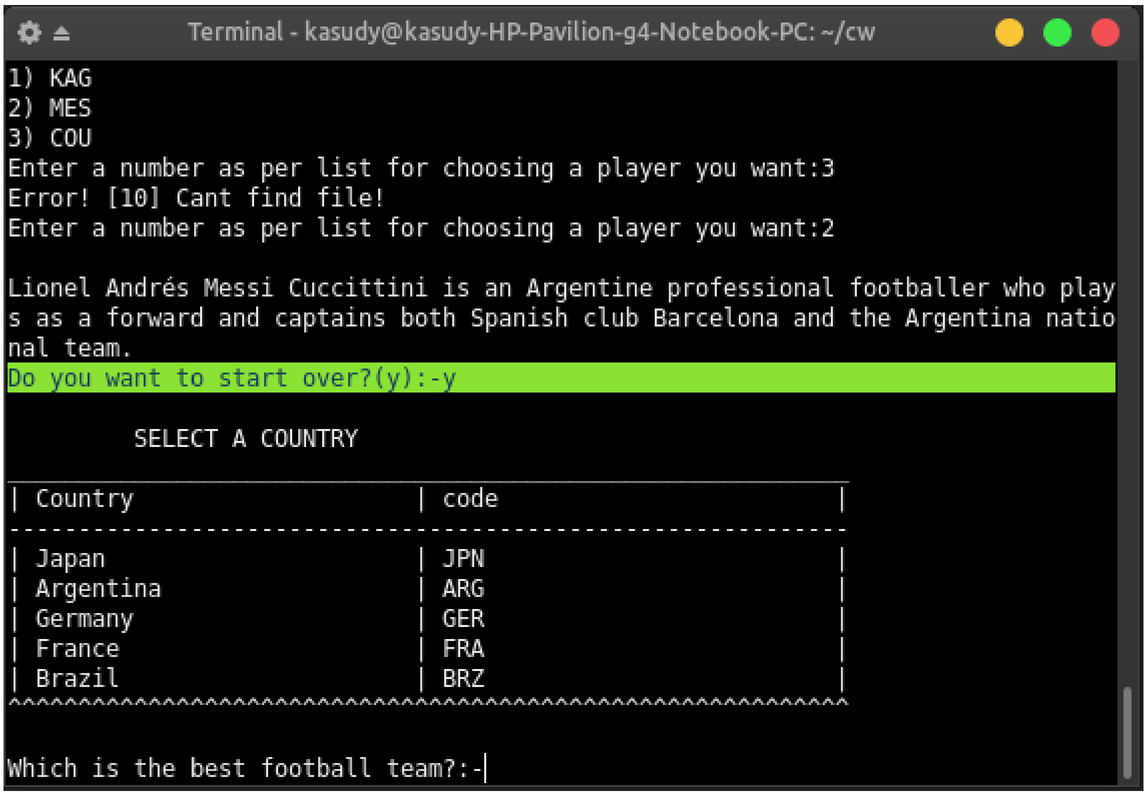


```
Terminal - kasudy@kasudy-HP-Pavilion-g4-Notebook-PC: ~/cw
The Five Star players at the world cup are:
____________________________________________________________
| Players                      | code                      |
------------------------------------------------------------
| Kagawa                       | KAG                       |
| Messi                        | MES                       |
| Reus                         | REU                       |
| Loris                        | LOR                       |
| Coutinho                     | COU                       |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Choose three players represented by codes separated by spaces
KAG MES COU

1) KAG
2) MES
3) COU
Enter a number as per list for choosing a player you want:3
Error! [10] Cant find file!
Enter a number as per list for choosing a player you want:2

Lionel Andrés Messi Cuccittini is an Argentine professional footballer who play
s as a forward and captains both Spanish club Barcelona and the Argentina natio
nal team.
Do you want to start over?(y):-
```

Figure 12: Favorite player with no description file

Table 13: Starting over

Objective	The start over function to repeat the previous codes
Action	Inputting y or 'YES' or 'Yes or 'Y'
Expected Result	The script re-runs from guessing the best team
Actual Result	The Script ran again from the expected part
Conclusion	Test Successful.



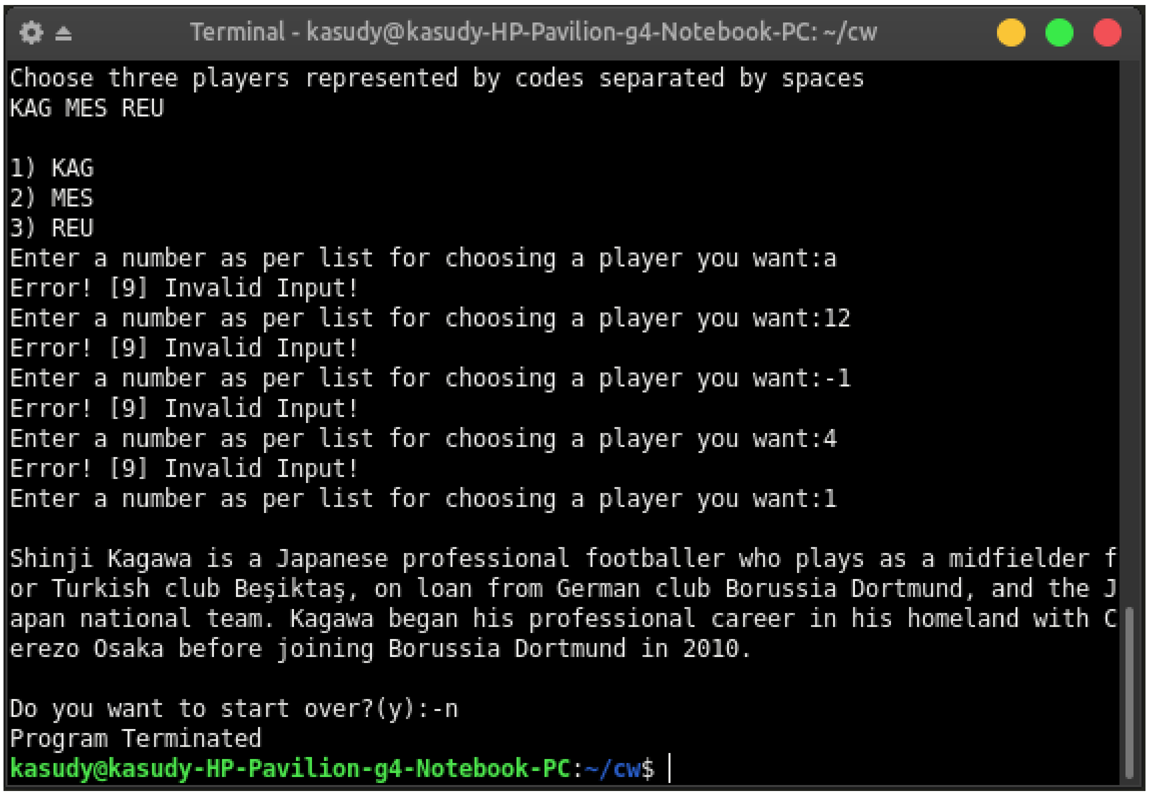
Terminal - kasudy@kasudy-HP-Pavilion-g4-Notebook-PC: ~/cw

```
1) KAG
2) MES
3) COU
Enter a number as per list for choosing a player you want:3
Error! [10] Cant find file!
Enter a number as per list for choosing a player you want:2

Lionel Andrés Messi Cuccittini is an Argentine professional footballer who play
s as a forward and captains both Spanish club Barcelona and the Argentina natio
nal team.
Do you want to start over?(y):-y

         SELECT A COUNTRY

____________________________________________________________
| Country                      | code                      |
------------------------------------------------------------
| Japan                        | JPN                       |
| Argentina                    | ARG                       |
| Germany                      | GER                       |
| France                       | FRA                       |
| Brazil                       | BRZ                       |
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Which is the best football team?:-
```

Figure 13: Starting over

Table 14: Terminating

Objective	Terminating the program after all the tasks are completed
Action	Inputting values other than y or 'YES' or 'Yes or 'Y'
Expected Result	The Script file terminates
Actual Result	The Script file terminated
Conclusion	Test Successful.



Terminal - kasudy@kasudy-HP-Pavilion-g4-Notebook-PC: ~/cw

```
Choose three players represented by codes separated by spaces
KAG MES REU

1) KAG
2) MES
3) REU
Enter a number as per list for choosing a player you want:a
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:12
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:-1
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:4
Error! [9] Invalid Input!
Enter a number as per list for choosing a player you want:1

Shinji Kagawa is a Japanese professional footballer who plays as a midfielder f
or Turkish club Beşiktaş, on loan from German club Borussia Dortmund, and the J
apan national team. Kagawa began his professional career in his homeland with C
erezo Osaka before joining Borussia Dortmund in 2010.

Do you want to start over?(y):-n
Program Terminated
kasudy@kasudy-HP-Pavilion-g4-Notebook-PC:~/cw$
```

Figure 14: Terminating

5. Conclusion

The tasks as specified in the assignments were done satisfying all the requirements as provided. The code and the contents of the files created for doing the operations are also been presented above. The extensive testing and the positive result provide the required evidence of the script being legit and showcase its features. The testing was done to reflect the real-time usage of the script and the problems it might face during any execution and it demonstrates how the program handles the invalid inputs or other user triggered loops and actions.

Task B

1. Introduction

I/O devices stand for Input and output devices which are essential for any computer system and human to communicate in physical form. It provides a physical interface in which we can interact with the computer by basic but crucial hardware. Basic computer peripherals such as mouse, keyboard, monitor, printer, etc. are all examples of IO devices. Therefore, to manage the I/O devices the UNIX utilizes a kernel which directly interacts with the devices and simplifies the process for the various multi-user terminals connected directly or via a network. UNIX refers to a family of Operating Systems which satisfy the POSIX standard and its license is governed by The Open Group company. There are many variants of UNIX but the most notable and popular is Mac OS. UNIX was created to be used by multiple users to handle multiple tasks running simultaneously in a sophisticated network of computers and peripherals which is the basis of any Network Operating System *(see appendix)*.

1.1. Aims

The aim of this report is to understand the UNIX Operating System and Network Operating System, why UNIX is used as a Network Operating System. Understand the categories of I/O Devices, Device Drivers and Device Controllers and how I/O System gets managed in UNIX.

1.2. Objectives

- Going through book, journals, industry white papers, websites, and textbooks to gather information.
- Recording information about the UNIX operating system.
- Understanding the use of UNIX as a Network Operating System.
- Searching about the categories of I/O Devices, Device Drivers and Device Controllers, management of I/O System such as I/O subsystem.
- Researching on the UNIX kernel and how it manages the I/O.

2. Body

Input and Output devices are the essential part of any Computer system which ensures the communication between various parts of the operation and provides a platform for the user to interact with a computer system. To handle all the I/O devices the UNIX Operating System deploys a monolithic kernel to interact in hardware level. So, to have a deeper understanding of the I/O operations in UNIX various topics are discussed below:

2.1. Categories of I/O Devices

There are various criteria by which I/O Devices can be categorized but the most comprehensive and comfortable categorization done according to the UNIX OS will be by the mode of operation of the device. According to this categorization, there will be various categories of I/O devices according to their communication operation. The most common and easy categorization of devices is by the type of data transported which are Character Devices and Block Devices which are directly connected to the System. But there are also Network Devices which are indirectly connected to a system via a network like printers, terminals, and other devices. However, according to the UNIX System, the devices are categorized into:

2.1.1. Character Devices

A device which sends data via a stream (String of data) which cannot be addressed or randomly accessed like mouse and keyboard (The kernel development community, 2019). The serial Ports, parallel ports are also a fine example of character devices which are used to interact with the computer system using a serial stream of data.

2.1.2. Block Devices

A device which sends bulk data in a fixed size but can be randomly accessible while communicating like Storage devices (Johnson, 1996). The block device also supports the layering of Memory mapped I/O. It can be beneficial on accessing the file base locating specific files instead of reading a whole file.

2.2. Device Drivers and Device Controllers

For an I/O device to work fluidly with a computer system, it requires a set of hardware and software components crafted accordingly to a specific device. The hardware-centric part which ensures the functionality of the Devices are named the Device Controllers while the software files which enables the communication of the OS to the Device Controllers are called Device Drivers.

2.2.1. Device Drivers

Device Drivers are a set of files that translates, converts the generic requests to the form which device controllers can understand to ensures the communication between the hardware and the Operating System. It usually operates a specific device attached to a computer by providing a suitable platform in the form of an abstracted software for optimal usage of the device without unnecessary complexity. The device driver allows the transfer of data between the devices (Computer Hope, 2019). The UNIX device drivers are the collection of files usually written in C to support the standard C function-calling mechanism and can be divided into four types: Character, Block, Terminal and Streams. The Character, Terminal and Streams device driver is for the various Character Devices as discussed before (Pajari, 1992).

2.2.2. Device Controller

Device Controllers are the pre-existing Software part of any device which is stored in the hardware itself instead of the OS. The main purpose of the device controllers is to abstract the coding and decoding of the signals generated and received by the device while communicating. The device Controller stores all the temporary data on a local buffer is connected via a suitable plug and socket interface to the CPU by the common bus. Device Controllers also have a corresponding device driver (Lithmee, 2018).

(The difference between Device Drivers and Device Controller are further discussed in the appendix.)

2.3. I/O subsystem

I/O subsystem manages the communication of various devices in a computer system, similarly, the management of I/O subsystem in UNIX is handled by the kernel which is responsible for all the data transfer in that process (Geeks for Geeks, 2019). I/O subsystem provides various features related to the I/O devices. The kernel of the OS is solely responsible for handling all the I/O operations directly as it is the only layer of the OS to have direct communication with the hardware. The operations handled by the kernel includes the scheduling, buffering, caching, spooling (device reservation), protection, error-handling, and other miscellaneous features.

2.3.1. I/O scheduling:

To improve the overall performance of a system it must deploy a certain algorithm which manages the scheduling the operations to execute the processes to make the processes more efficient. Scheduling implements the idea to execute the processes in a certain manner which maximizes the efficiency of the system.

2.3.2. Buffering, caching and spooling

Buffering and caching are very similar techniques, but buffering holds the only one copy of the given information while caching holds the duplicate copy of existing information for faster access. Spooling is also like buffering as it holds the output of a device in a queue in case of multiple instances of tasks to be executed as soon as possible (Silberschatz, et al., 2008).

2.3.3. Error handling and protection

The I/O subsystem also encounters various problems which it must protect from and it also prevents misuse of the normal functions of the I/O (GeeksforGeeks, 2019).

(These topics are further discussed in the appendix)

2.4. DMA

DMA (Direct Memory Access) is a capability built into some of the computer bus architectures which enables direct access to the main memory of the computer to the various hardware subsystems. DMA is managed by a special chip named DMA Controller (DMAC) which will ease the process of memory operation by decreasing the cycles which pass through the CPU. This is a vital I/O feature for data transfer from the various device directly to the Memory (Sexton, 1996). DMA can simplify the transfer of large quantities of data without being redirected to the CPU, so the DMA controllers are a standard feature in the modern Computer Systems.

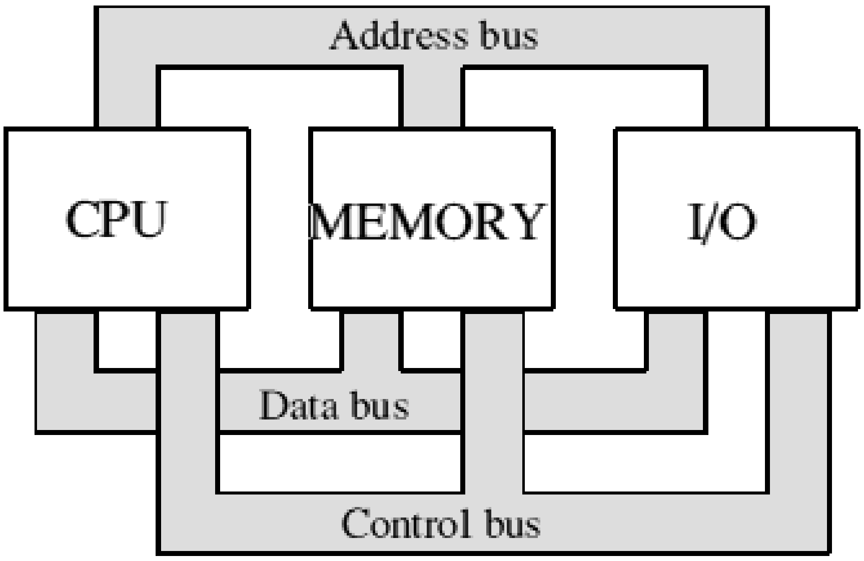


Figure 15: Bus connecting vital parts of a Computer System (Sexton, 1996)

3. Conclusion

As the UNIX was mostly designed for big co-operations and used for large scale computers, which limited its domain due to high margin, thus an alternative Operating System was in high demand which had all the advantages of the Unix Operating System while being easily accessible it was also a really great Network Operating system as it provided all the basic functionality and was designed to run on a network with multiple devices and users. Subsequently, to make the UNIX-like method more accessible, Linus Torvalds created a separate kernel named Linux kernel which was open source, unlike the UNIX. Therefore, by making the Kernel provided by Linus the GNU community started to develop applications to provide UNIX like functionalities. Together with the Linux Kernel and GNU applications created a fully functional open-source Operating System which provided most of the functionality of the UNIX operating system while being easily accessible.

I/O is a vital part of a computer system. It ensures the communication between different devices and users. There are various categories of I/O devices but the most common is the character Devices and Block devices categorized by the type of information communicated. The UNIX has four levels for abstraction and the kernel is responsible for all the communication between the hardware and the applications. The kernel is also responsible for managing the I/O operations. The monolithic kernel is widely used in the UNIX and UNIX like Operating Systems. The I/O device consists of blocked and character devices and requires device drivers which can easily communicate with the other components. The I/O subsystems are also handled by the kernel which includes I/O scheduling, Caching, Buffering, Spooling, Error Handling, and I/O security. DMA is a useful feature which is useful for transferring large amounts of data, so a lot of computer system nowadays comes with a DMA Controller to ease the I/O operations.

References

Cesati, M. & Bovet, D. P., 2002. *Understanding the Linux Kernel.* 2nd ed. Sebastopol: O'Reilly Media.

Computer Hope, 2019. *Device Drivers.* [Online]
Available at: https://www.computerhope.com/jargon/d/driver.htm
[Accessed 6 April 2019].

Dummies: A Wiley Brand, 2019. *Network Operating System Features – Support And File Sharing.* [Online]
Available at: https://www.dummies.com/programming/networking/network-operating-system-features-support-and-file-sharing/
[Accessed 4 April 2019].

Estes, P., 2018. *Linux vs. Unix: What's the difference?.* [Online]
Available at: https://opensource.com/article/18/5/differences-between-linux-and-unix
[Accessed 3 Feburary 2019].

Geeks for Geeks, 2019. *Operating System | Kernel I/O Subsystem (I/O System).* [Online]
Available at: https://www.geeksforgeeks.org/operating-system-kernel-i-o-subsystem-i-o-system/
[Accessed 4 April 2019].

Giometti, R., 2017. *GNU/Linux Rapid Embedded Programming.* 1st ed. Birmingham: Packt Publishing Ltd.

Indiana University, 2019. *About Unix.* [Online]
Available at: https://kb.iu.edu/d/agat
[Accessed 29 March 2019].

Johnson, M. K., 1996. *Block Device Drivers.* [Online]
Available at: http://www.tldp.org/LDP/khg/HyperNews/get/devices/block.html
[Accessed 8 April 2019].

Lewine, D., 1991. *POSIX Programmers Guide.* Reprint ed. California: O'Reilly Media, Inc..

Lithium, 2018. *What is the Difference Between Device Driver and Device Controller.* [Online]
Available at: http://pediaa.com/what-is-the-difference-between-device-driver-and-device-controller/
[Accessed 6 April 2019].

McHoes, A. M. & Flynn, I. M., 2011. *Understanding Operating Systems.* 6th ed. Boston: Cengage Learning.

Pajari, G., 1992. *Writing UNIX Device Drivers.* First ed. Boston: Addison-Wesley Pub. Co..

Rouse, M., 2005. *POSIX (Portable Operating System Interface).* [Online]
Available at: https://whatis.techtarget.com/definition/POSIX-Portable-Operating-System-Interface
[Accessed 3 February 2019].

Rowe, L. & Birman, K., 1982. A Local Network Based on the UNIX Operating System. *IEEE Transactions on Software Engineering ,* 8(2), pp. 137-246.

Shotts Jr., W. E., 2019. *What Is "The Shell"?.* [Online]
Available at: http://linuxcommand.org/lc3_lts0010.php
[Accessed 3 February 2019].

Stevens, W., 2003. *UNIX Network Programming.* 3rd ed. Boston: Addison Wesley.

TekSlate, 2019. *Unix System Architecture And Its Explanation.* [Online]
Available at: https://tekslate.com/unix-system-architecture-explination
[Accessed 8 April 2019].

The Geek Diary, 2019. *UNIX / Linux : What Is a Shell? What are different Shells?.* [Online]
Available at: https://www.thegeekdiary.com/unix-linux-what-is-a-shell-what-are-different-shells/
[Accessed 4 April 2019].

The kernel development community, 2019. *Character device drivers.* [Online]
Available at: https://linux-kernel-labs.github.io/master/labs/device_drivers.html
[Accessed 8 April 2019].

The Open Group, 2013. *Certification.* [Online]
Available at: https://www.opengroup.org/openbrand/
[Accessed 2 April 2019].

Winkelman, D. R., 2013. *What is a Network?.* [Online]
Available at: https://fcit.usf.edu/network/chap1/chap1.htm
[Accessed 8 April 2019].

Appendix

Password for the script file: noway

UNIX and UNIX-like Operating Systems

UNIX refers to a family of Operating Systems which satisfy the POSIX standard and whose license is governed by The Open Group company. Originally created by researchers at AT&T Bell Labs using a high-level programming language(C), it was highly popular as it was easier to understand, portable and modular in nature, and provided multi-user platform (Indiana University, 2019). The basic principles of the UNIX OS are derived from previous Multix (Estes, 2018) which was created to serve as an OS for large scale systems. Now various flavors of the UNIX OS are provided by vendors, the most popular being the Mac OS, a single user UNIX OS provided by Apple Inc. POSIX (Portable Operating System Interface) is an IEEE standard designed to facilitate application portability to facilitate C programming interfaces (Rouse, 2005). This will define how the files get managed, processes get handled, the functionality of the Terminal in the Operating System (Lewine, 1991). The open Group extends the POSIX certification and maintains the overall standard for the UNIX trademark (The Open Group, 2013).

As the UNIX was mostly designed for big co-operations and used for large scale computers, which limited its domain due to high margin, so an alternative Operating System was in high demand which had all the advantages of the Unix Operating System while being easily accessible. Thus, to counter this problem Linus Torvalds created a separate kernel named Linux kernel which was open source, unlike the UNIX. Therefore, by making the Kernel provided by Linus the GNU community started to develop applications to provide UNIX like functionalities. Together with the Linux Kernel and GNU applications created a fully functional open-source Operating System which provided most of the functionality of the UNIX operating system while being easily accessible.

GNU/Linux or simply Linux is a set of Operating system mostly Open Source which uses Linux kernel developed by Linus Torvalds and various component and software developed by the GNU community started by Richard Stallman (Giometti, 2017). Most of the popular Linux distributions like Ubuntu are POSIX certified OS (Lewine, 1991) which has many UNIX-like features and supports a similar command line Interface native to UNIX. Linux Operating systems are generally POSIX certified operating system which has many shared features with the UNIX operating system.

Network Operating System

A network operating system (NOS) is simply a type of Operating System which is designed to support any type of computer, peripherals or terminal connected via a network. A network operating system enables the sharing of printers, files, interfaces, and applications. It also used to easily manage operations and entities in a network (Dummies: A Wiley Brand, 2019). UNIX is popular multitasking, multi-user computer operating system was first intended to be used in the Bell Labs. Since it was modular and hosted various terminal to access the main computer system through the network it was a good network operating system with all the capabilities (Rowe & Birman, 1982). There are various examples of Network OS, but the most mainstream NOS are VINES, LAN Manager, Windows NT, Windows Server, Linux Distros and many more. These Operating Systems have some essential features which make the use of the device in a network more fluid and robust (McHoes & Flynn, 2011).

Features of Network Operating System:

- Basic Operating System to support basic software
- Gives protocol support for communication
- Provides Security and manages the services over a network
- Provides data, program, and device sharing features
- Various network and internetworking-based services

Utilization of the Network Operating System:

- For the administration of multiple devices
- System Maintenance
- Managing shared features
- Security management over the network

Communication in UNIX

The UNIX abstracts the communication of the Users and the Hardware by four different layers for simplicity. These are the Kernel, Shell, and applications. The Shell and Application are the interfaces by which a user can interact with the computer while the kernel is the part which communicates directly with the Devices (TekSlate, 2019).

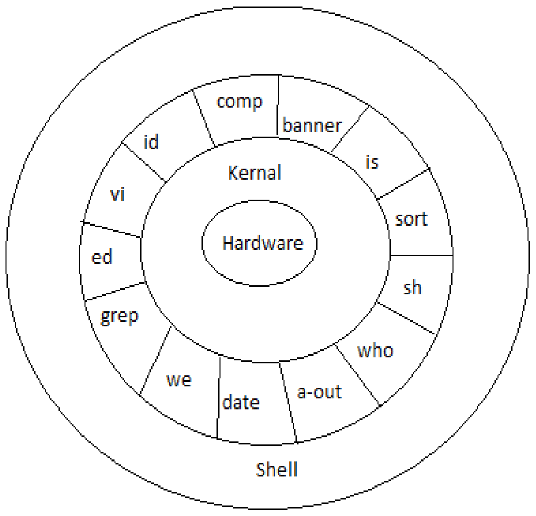


Figure 16: Communication in UNIX (TekSlate, 2019)

<u>Kernel</u>

Kernal interacts directly with the hardware, provides processor scheduling, memory management, and I/O management. It also provides interprocess communication (Cesati & Bovet, 2002). The kernel handles the I/O Scheduling, Buffering, cashing, Spooling, Error Handling, and I/O protection. There are various categories of kernels: monolithic kernels, exokernels, hybrid kernels, and microkernels. Monolithic kernels are the most common in most traditional UNIX-like Operating Systems, it has the features to modules at runtime, extending the kernels abilities as desired.

<u>Shell and Application</u>

A shell is a program bridge the platform between the user and the kernel of the operating system (The Geek Diary, 2019). The shell is customizable and provides the desired user environment depending on the prerequisites. There are various types of shell the most notable being the Bourne Shell in the UNIX. The Korn Shell, C Shell, and the Bourne Again Shell also provide a similar or better user environment to the user according to the scenario. Both the Shell and the Application can interact with the user directly in the UNIX OS but requires the kernel which then translates the commands and information for this layer.

Differences between Device Controller and Device Driver:

Device Controller:

- Part of the computer System usually a hardware part.
- Translates the signal between the device and The CPU.
- Converts the Serial stream of data to blocks.
- Performs Error correction where required while converting.

Device Driver

- Program or files to support a device.
- Feeds Digital information to the Controller.
- A translator between the Hardware component and the upper-level applications.
- Simply a set of instructions for supporting a device.

3.1.1. I/O Scheduling

The UNIX Operating System handles the I/O operations by scheduling them in various orders so that various resources of a Computer System are managed by the various users. I this improve the overall efficiency and tasks can be prioritized according to their importance. Buffering and Caching can also enable various flexible options for scheduling. Various methods for the scheduling have been proposed for maximizing the efficiency in different scenarios but the most popular schedulers currently deployed are Completely Fair Queuing (CFQ), Deadline, NOOP, and Anticipatory. The algorithms used are usually Round Robin Scheduling or FCFS (First Come First Service) (Geeks for Geeks, 2019).

3.1.2. Buffering, Caching and their differences

Buffering and caching are very similar techniques, but buffering holds the only one copy of the given information while caching holds the duplicate copy of existing information for faster access.

Buffering stores the transferring data to smooth out the impedance mismatch between the devices. There are various buffering strategies such as Single buffering, Double Buffering and Circular Buffering which are deployed to various I/O operations. Buffering can greatly enhance the I/O operation by cutting down the waiting time by matching the speed between the sender and receiver.

Caching enhances the transfer speed by providing a fast memory holding platform for various I/O operations by storing a copy of original data to a faster location. This decreases the loading time and is very essential in I/O operations (Silberschatz, et al., 2008).