Module Code & Module Title

CS5001NA Networks and Operating System

Assessment Weightage & Type

40% Individual Coursework

Year and Semester

2019-20 Autumn

Student Name: Bibhu Manandhar

London Met ID: 18029955

College ID: NP01CP4A180301

Assignment Due Date: 29th May, 2020

Assignment Submission Date: 22th May, 2020

Word Count (TASK B): 2195

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

Table of Contents

Table of Figures

Table of Tables

Task A

Introduction

This is an individual coursework which is part 2 of coursework 2 given to us in week 20. In this part of the coursework we were given the task to write a shell script that demonstrated our ability to organize interactive communication of its potential user with the UNIX environment in a friendly manner. The script was implemented regarding the scenario provided in the coursework. All the scripts were also done using Debian system which currently use the Linux Kernel or the FreeBSD kernel.

UNIX is an operating system which is a suite of programs that makes computer work while it is also a stable, multi-user, multi-tasking system for servers, desktops and laptops. Similar to Microsoft Windows, UNIX systems also have a graphical user interface which offer a trouble-free environment. Sun Solaris, GNU/Linux, and MacOS X are the most popular variation of UNIX. The UNIX working framework is comprised of three sections, the kernel, shell and the programs. The hub of the OS is the kernel of UNIX whereas the shell acts as an interface between the user and the kernel. (M.Stonebank, 2000)

Aims and Objectives

The main aim for Task A of this coursework is to develop a script that implements interaction with the UNIX environment in a user-friendly manner while also performing simple input/output operations. Whereas the objective for this part of the coursework is to develop the script which is a small program in the environment of a well-known UNIX shell.

The aims and objectives or these simplicity and friendliness can be acquired by the following points:

- Constructing an interactive request by using UNIX commands.
- Checking user input for errors.
- Diagnosing these errors with clear messages.
- Testing the developed script.
- Finalizing all the progress in a report.

Script

This includes the code of the program

```
#!/bin/bash
#For Username and UserID
value='^[+-]?[0-9]+$'
if [ $# != 2 ] #Checking for user input of Username and UserID or not.
then
        echo -e "\nPlease Enter Your First Name and then ID Number."
        exit

elif ! [[ $2 =~ $value ]] #UserID can only be numeric value so checking if
numeric value or not.
then
        echo -e "\nOnly numeric value for ID Number. Please Try again."
        exit
fi

#Secret key to run the program
secretkey="bibhu"
count=4
echo -e ""
echo -e "Enter the secret key:\c"

read key
until [ $secretkey == $key ] #Looping five times until the secret key is
not matched, and after that exit the program.
do
        if [ $count = 0 ]
        then

                echo -e "Incorrect Password. No chances left."
                exit
        else
                echo -e "Invalid Key. $count chances remaining.\c"
                read key
                count=`expr $count - 1` #Decreases count everytime wrong key is
entered.
        fi
done

#Welcoming the user when secret key is matched
echo -e "---------WELCOME-----------"
echo -e "---------UserID:$2---------"
echo -e "---------Name:$1-----------"
echo -e "***Program started on $(date).***"


#Guessing the best cricket team to played Cricket


BestTeam() #Function for user to guess the best cricket team
{
echo -e ""
echo -e "Guess the best cricket team:\c"
read best
case $best in
NEP | nep) #executes when the users guesses the right team.
        echo -e "The best team is Nepal."
        echo -e "Its our country and we must support them."
```

```
;;

*) #for wrong guesses
      echo -e "Wrong team guessed. Guess again."
      BestTeam #Calls the function again and runs until the right team is
guessed.
;;
esac
}

#Five players from Cricket

ChoosePlayer() #Function for choosing player from five.
{
echo -e "\t\t\t--------------------------"
echo -e "\t\t\t| Players      \t| Codes |"
echo -e "\t\t\t| Paras Khdaka \t| PK    |"
echo -e "\t\t\t| Virat Kholi  \t| VK    |"
echo -e "\t\t\t| David Warner \t| DW    |"
echo -e "\t\t\t| Ben Stokes   \t| BS    |"
echo -e "\t\t\t| Ross Taylor  \t| RT    |"
echo -e "\t\t\t--------------------------"

#Three players to choose among five

echo -e "Choose three players with thier corresponding code with a space
seperating them."
read p1 p2 p3 #Reads the three players choosen by the user.
players=($p1 $p2 $p3)
if [[ ${#players[@]} -eq 3 ]] #Condition if three players are choosen or
not.
then
      echo "Three players has been choosen."
else #Keeps on looping until three players are choosen.
      echo "Please choose three players."
      ChoosePlayer #Function keeps on looping until three players are
choosen.

fi
play=($p1 $p2 $p3) #Checking if the users chooses the player code available
and not other than existing codes.
for p in ${play[@]};
do
      if [ $p == PK ] || [ $p == VK ] || [ $p == DW ] || [ $p == BS ] || [
$p == RT ]
      then
            echo
      else
            echo -e "Invalid Selection. Please select the code from the
list."
            ChoosePlayer #Keeps looping until user selection is valid.
            break
      fi
done

if [ $p1 == $p2 ] || [ $p1 == $p3 ] || [ $p2 == $p3 ] #Checking if the same
player is choosen more than once.
then
      echo
      echo
"__________________________________________________________________________
_________"
      echo "Same player has been choosen more than once. Please choose
three different players."
      echo "---------------------------------------------------------------
--------------------"
      ChoosePlayer # This function is called so the user can try again if
the same player was entered more than once.
fi

}

#One player to be choosen among three.
ChooseOne() #Function for choosing one player among the three choosen
players.
{
echo -e ""
echo -e "\t\t\t The three choosen players are:"
echo -e "\t\t\t ------------------------------"
echo -e "\t\t\t| PlayerCode |"
echo -e "\t\t\t--------------"
echo -e "\t\t\t| 1.$p1      |"
echo -e "\t\t\t| 2.$p2      |"
echo -e "\t\t\t| 3.$p3      |"
echo -e "\t\t\t--------------"

PS3="Choose a number among the three numbers shown above:"
select players in $p1 $p2 $p3
do
      case $players in  #Case for each number choosen.
            $p1)
            if [ -f $p1 ];
            then
                  cat $p1
                  details
            break
            else # Restarting when a non readable file is choosen.
                  echo -e "File not readable. Please choose again."
                  ChoosePlayer
                  ChooseOne
                  break
            fi
            ;;
            $p2)
                  if [ -f $p2 ];
                  then
                        cat $p2
                  details
                  break
                  else
                        echo -e "File not readable. Please choose again."
                        ChoosePlayer
                        ChooseOne
                        break
                  fi
                  ;;
            $p3)
                  if [ -f $p3 ];
                  then
                        cat $p3
```

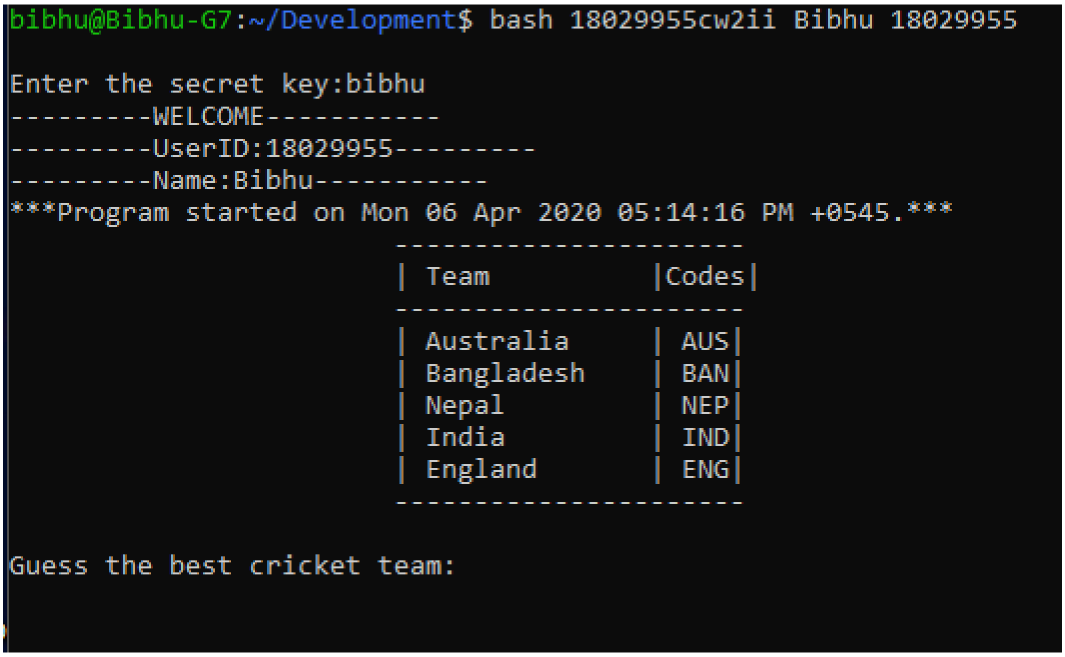
```
                                details
                                break
                        else
                                echo -e "FIle not readable. Please choosen again."
                                ChoosePlayer
                                ChooseOne
                                break
                        fi
                        ;;
                        *) # When the user inputs number except the existing
numbers.
                        echo "Invalid Choice"
                        ChooseOne
                        break
                        ;;
                esac
done
}
details()
{
echo -e "Press Y/y to know about other playes or any other key to
continue:\c"
read detail
case $detail in
Y|y)
        ChooseOne
;;
*)
;;
esac

}

program() #Program sequence
{

#Cricket Teams
echo -e "\t\t\t----------------------"
echo -e "\t\t\t| Team       \t|Codes|"
echo -e "\t\t\t----------------------"
echo -e "\t\t\t| Australia   \t| AUS|"
echo -e "\t\t\t| Bangladesh  \t| BAN|"
echo -e "\t\t\t| Nepal       \t| NEP|"
echo -e "\t\t\t| India       \t| IND|"
echo -e "\t\t\t| England     \t| ENG|"
echo -e "\t\t\t----------------------"

BestTeam #Calls the function BestTeam

ChoosePlayer #Calls the function ChoosePlayer

ChooseOne #Calls the function ChooseOne

}

repeat="yes"
while [ $repeat = "yes" ] || [ $repeat = "YES" ] || [ $repeat = "y" ] || [
$repeat = "Y" ]
do #Re-running the process if the user wants to.
        program #Callig the loop function.
```

```
	echo -e "Do u want to try again? Press Yes to continue and other to
exit:\c"
	read repeat #Reading the user chouce.
done
echo -e "Thank you."
```

Testing

Test No. 1	When valid inputs are given.
Input	Unix
Expected Output	Welcomes the user, mentioning the ID number first, then username and then displays the date and time of execution.
Actual Output	User was welcomed with their UserID, name and, date and time of execution.

Table 1: Testing no. 1

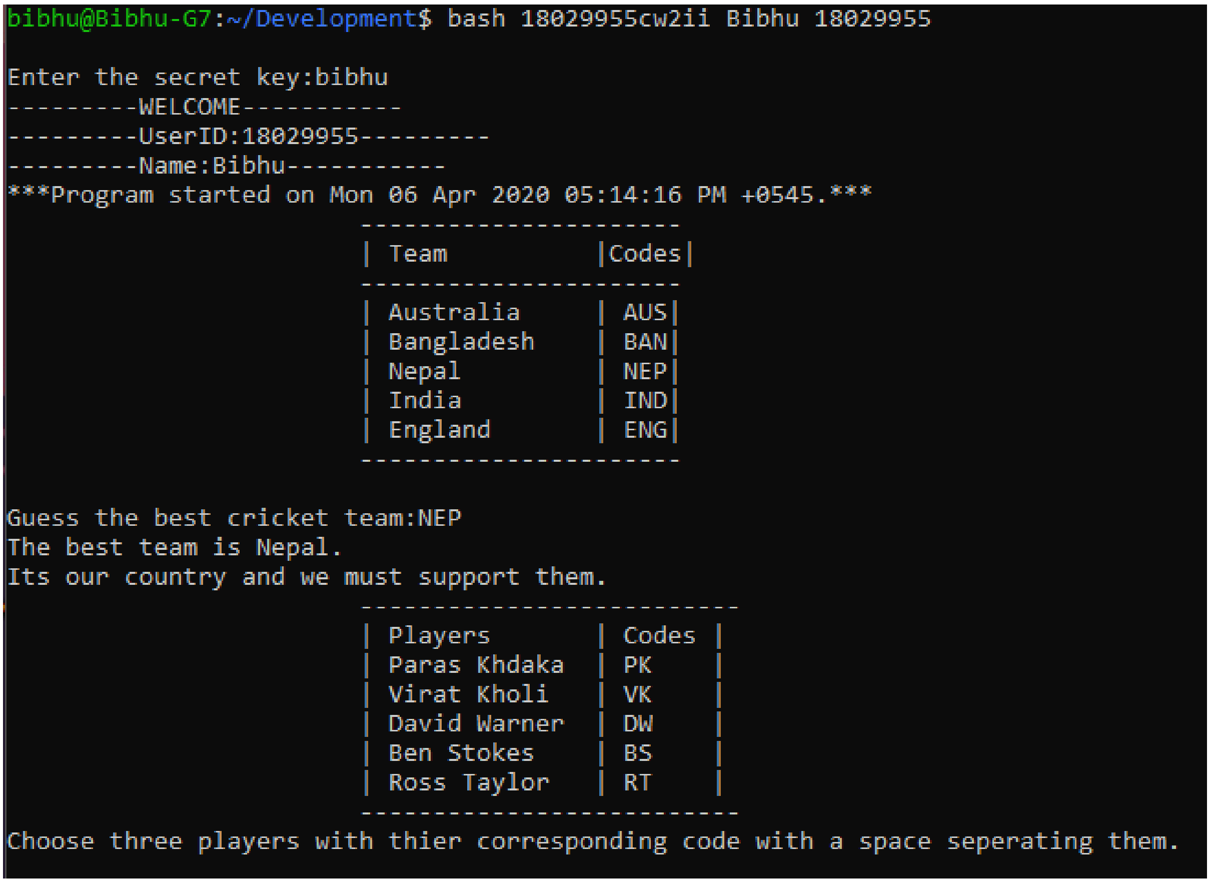


```
bibhu@Bibhu-G7:~/Development$ bash 18029955cw2ii Bibhu 18029955

Enter the secret key:bibhu
---------WELCOME-----------
---------UserID:18029955---------
---------Name:Bibhu-----------
***Program started on Mon 06 Apr 2020 05:14:16 PM +0545.***
                        ----------------------
                        | Team           |Codes|
                        ----------------------
                        | Australia      | AUS|
                        | Bangladesh     | BAN|
                        | Nepal          | NEP|
                        | India          | IND|
                        | England        | ENG|
                        ----------------------

Guess the best cricket team:
```

Figure 1: Test 1

Test No. 2	When the correct best cricket team is chosen.
Input	NEP
Expected Output	The system informs about the correct option and a description about the country is displayed. Then List of players with their codes are displayed.
Actual Output	The system does inform about the correct guess and the players were shown.

Table 2: Testing no. 2

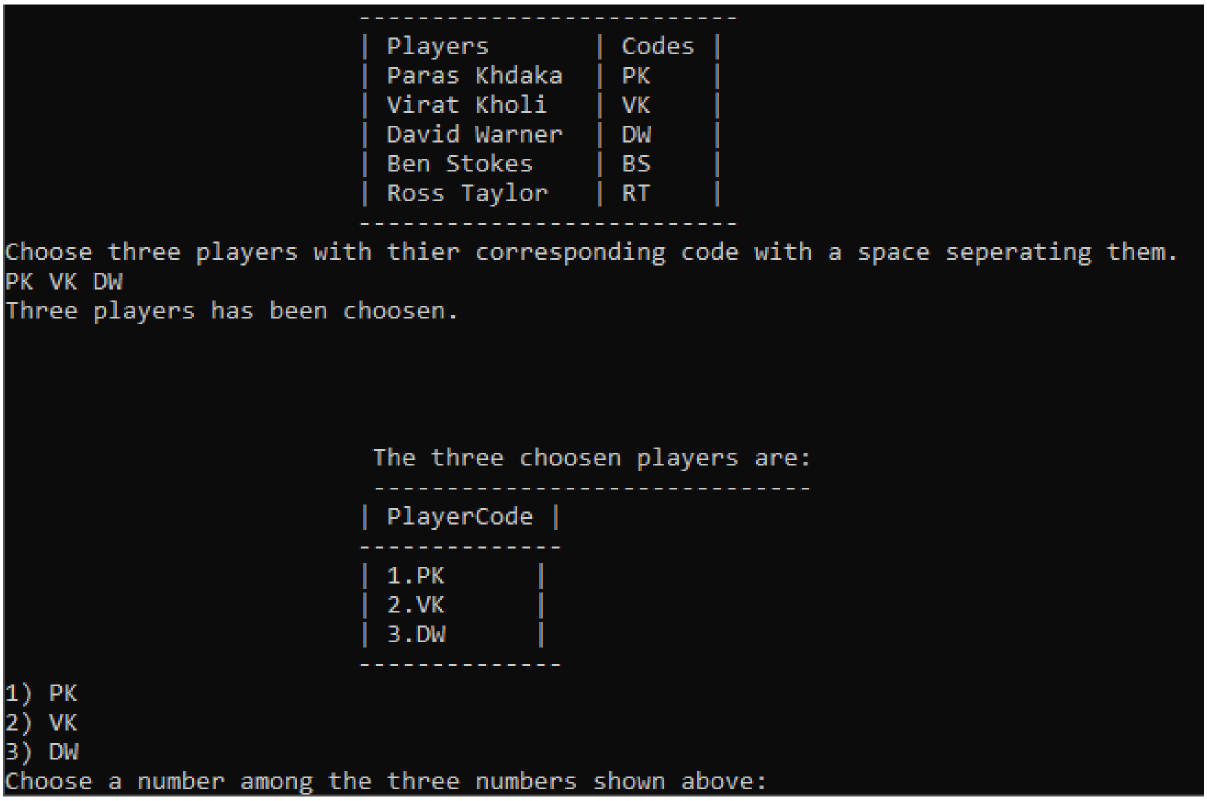


```
bibhu@Bibhu-G7:~/Development$ bash 18029955cw2ii Bibhu 18029955

Enter the secret key:bibhu
---------WELCOME-----------
---------UserID:18029955---------
---------Name:Bibhu-----------
***Program started on Mon 06 Apr 2020 05:14:16 PM +0545.***
                        ----------------------
                        | Team           |Codes|
                        ----------------------
                        | Australia      | AUS|
                        | Bangladesh     | BAN|
                        | Nepal          | NEP|
                        | India          | IND|
                        | England        | ENG|
                        ----------------------

Guess the best cricket team:NEP
The best team is Nepal.
Its our country and we must support them.
                        --------------------------
                        | Players        | Codes |
                        | Paras Khdaka   | PK    |
                        | Virat Kholi    | VK    |
                        | David Warner   | DW    |
                        | Ben Stokes     | BS    |
                        | Ross Taylor    | RT    |
                        --------------------------
Choose three players with thier corresponding code with a space seperating them.
```

Figure 2: Test 2

Test No. 3	Choosing three player codes and displaying them with a menu box.
Input	PK VK DW
Expected Output	The chosen players were displayed and a menu box is shown to choose a number among the chosen players.
Actual Output	The three players that were chosen were displayed and menu box is shown.

Table 3: Testing no. 3

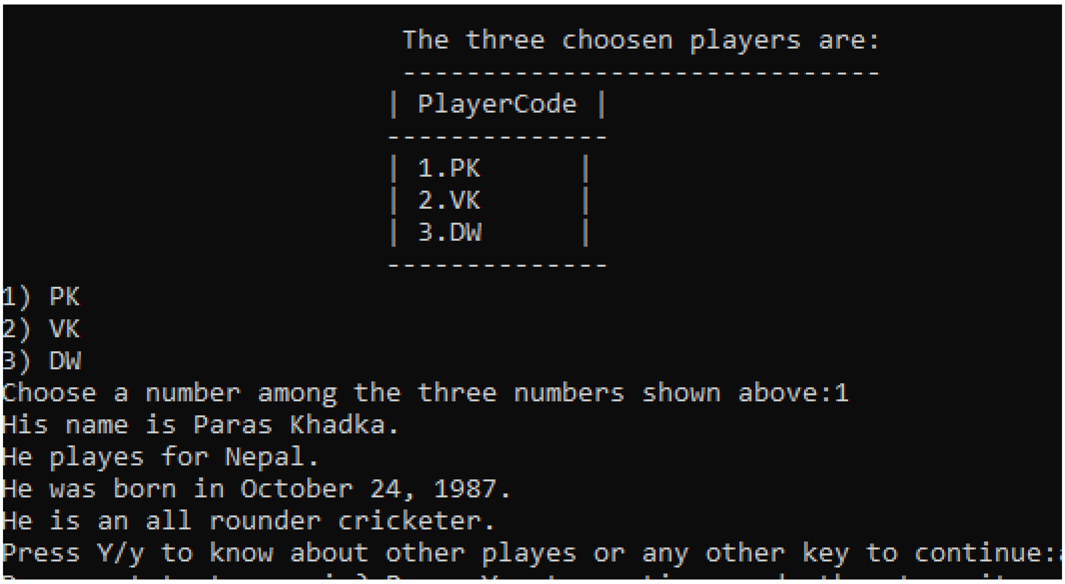


```
                         -------------------------
                        | Players       | Codes |
                        | Paras Khdaka  | PK    |
                        | Virat Kholi   | VK    |
                        | David Warner  | DW    |
                        | Ben Stokes    | BS    |
                        | Ross Taylor   | RT    |
                         -------------------------
Choose three players with thier corresponding code with a space seperating them.
PK VK DW
Three players has been choosen.



                         The three choosen players are:
                         ------------------------------
                        | PlayerCode |
                         -------------
                        | 1.PK       |
                        | 2.VK       |
                        | 3.DW       |
                         -------------
1) PK
2) VK
3) DW
Choose a number among the three numbers shown above:
```

Figure 3: Test 3

Test No. 4	Details of the selected player among three if there is a file of it and asks the user to repeat or not.
Input	1
Expected Output	A description about the player is shown and an option for repeating the steps is displayed.
Actual Output	Description about the selected player is shown and an option message is displayed.

Table 4: Testing no. 4

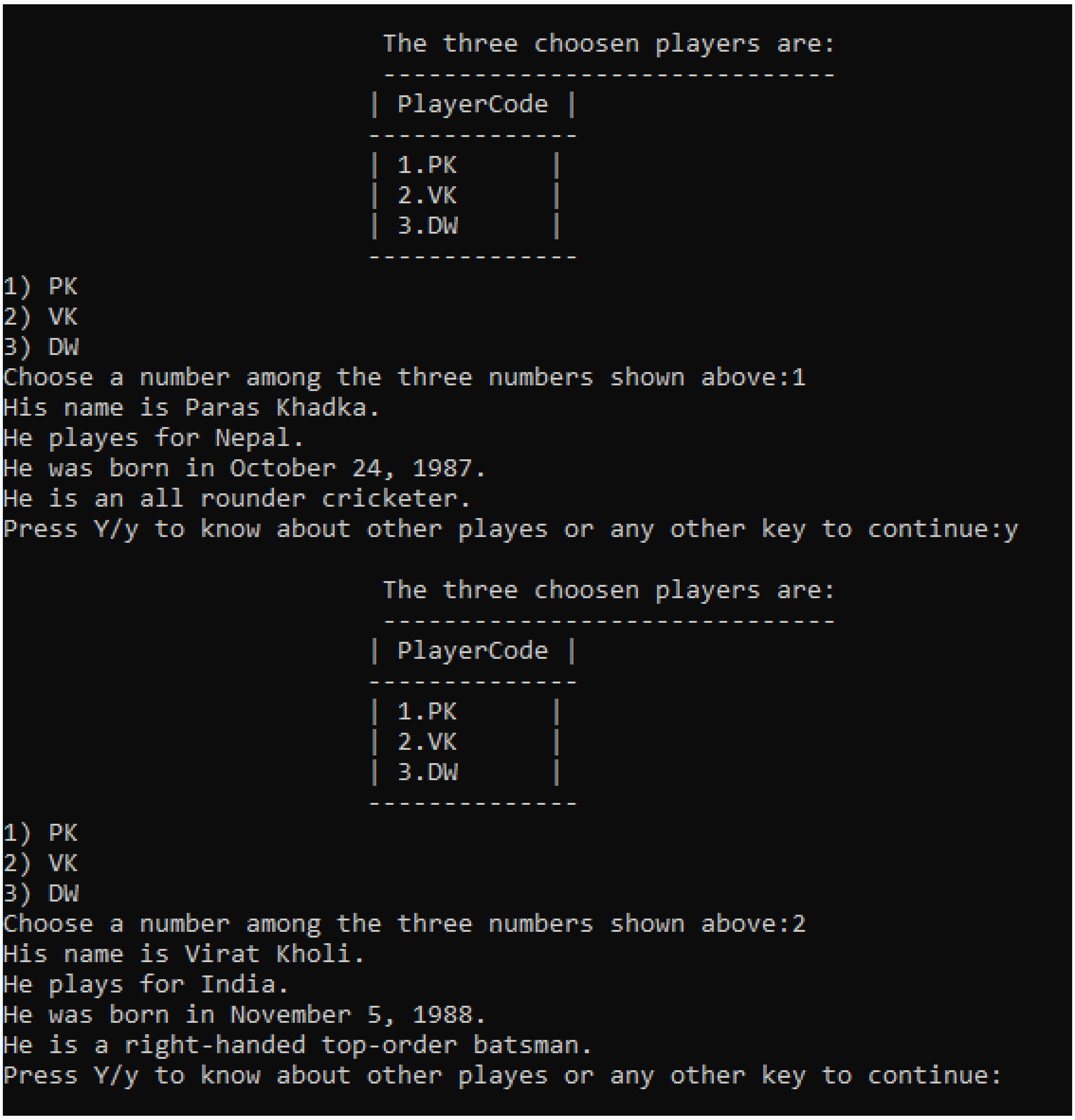


```
                         The three choosen players are:
                         ------------------------------
                        | PlayerCode |
                        -------------
                        | 1.PK      |
                        | 2.VK      |
                        | 3.DW      |
                        -------------
1) PK
2) VK
3) DW
Choose a number among the three numbers shown above:1
His name is Paras Khadka.
He playes for Nepal.
He was born in October 24, 1987.
He is an all rounder cricketer.
Press Y/y to know about other playes or any other key to continue:
```

Figure 4: Test 4

Test No. 5	Checking when Y/y is pressed the user can view the other player’s description.
Input	Y
Expected Output	The program runs the process where the user can see the description of other players among the already chosen three.
Actual Output	The program does repeat the process again and displays the player description.

Table 5: Testing no. 5

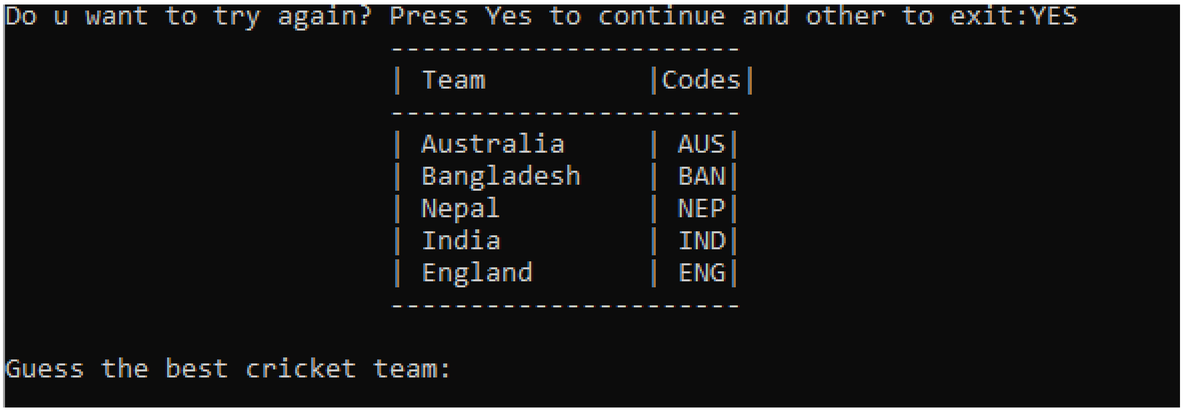


```
                          The three choosen players are:
                          ------------------------------
                         | PlayerCode |
                         --------------
                         | 1.PK      |
                         | 2.VK      |
                         | 3.DW      |
                         --------------
1) PK
2) VK
3) DW
Choose a number among the three numbers shown above:1
His name is Paras Khadka.
He playes for Nepal.
He was born in October 24, 1987.
He is an all rounder cricketer.
Press Y/y to know about other playes or any other key to continue:y

                          The three choosen players are:
                          ------------------------------
                         | PlayerCode |
                         --------------
                         | 1.PK      |
                         | 2.VK      |
                         | 3.DW      |
                         --------------
1) PK
2) VK
3) DW
Choose a number among the three numbers shown above:2
His name is Virat Kholi.
He plays for India.
He was born in November 5, 1988.
He is a right-handed top-order batsman.
Press Y/y to know about other playes or any other key to continue:
```

Figure 5: Test 5

Test No. 6	Checking if the program goes back to the process of guessing the best cricket team when asked to the user to be repeated.
Input	YES
Expected Output	The program runs to the process where the user have to guess the best cricket team.
Actual Output	The program does go to the process of guessing the best cricket team and so on.

Table 6: Testing no. 6

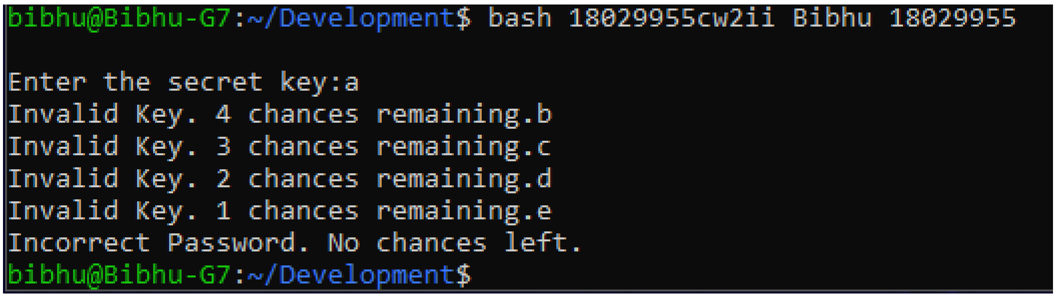


```
Do u want to try again? Press Yes to continue and other to exit:YES
                        ----------------------
                        | Team            |Codes|
                        ----------------------
                        | Australia       | AUS|
                        | Bangladesh      | BAN|
                        | Nepal           | NEP|
                        | India           | IND|
                        | England         | ENG|
                        ----------------------

Guess the best cricket team:
```

Figure 6: Test 6

Test No. 7	Testing when the secret key is incorrect.
Input	Incorrect secret keys
Expected Output	The program should prompt to enter correct key for five times and then exit the program.
Actual Output	A message is displayed that incorrect key was entered with the remaining chances and after all the chance terminates the program.

Table 7: Testing no. 7

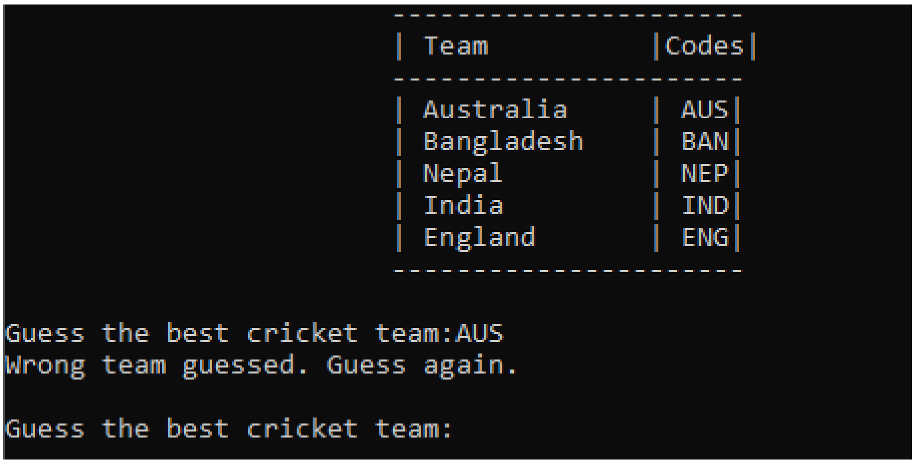


```
bibhu@Bibhu-G7:~/Development$ bash 18029955cw2ii Bibhu 18029955

Enter the secret key:a
Invalid Key. 4 chances remaining.b
Invalid Key. 3 chances remaining.c
Invalid Key. 2 chances remaining.d
Invalid Key. 1 chances remaining.e
Incorrect Password. No chances left.
bibhu@Bibhu-G7:~/Development$
```

Figure 7: Test 7

Test No. 8	Testing when the best cricket team is guessed incorrect.
Input	Incorrect cricket team code.
Expected Output	The program warns about the mistake and user can keep on guessing until the correct option is chosen.
Actual Output	A message is displayed that wrong team is guessed and asks to guess again until the right team is guessed.

Table 8: Testing no. 8

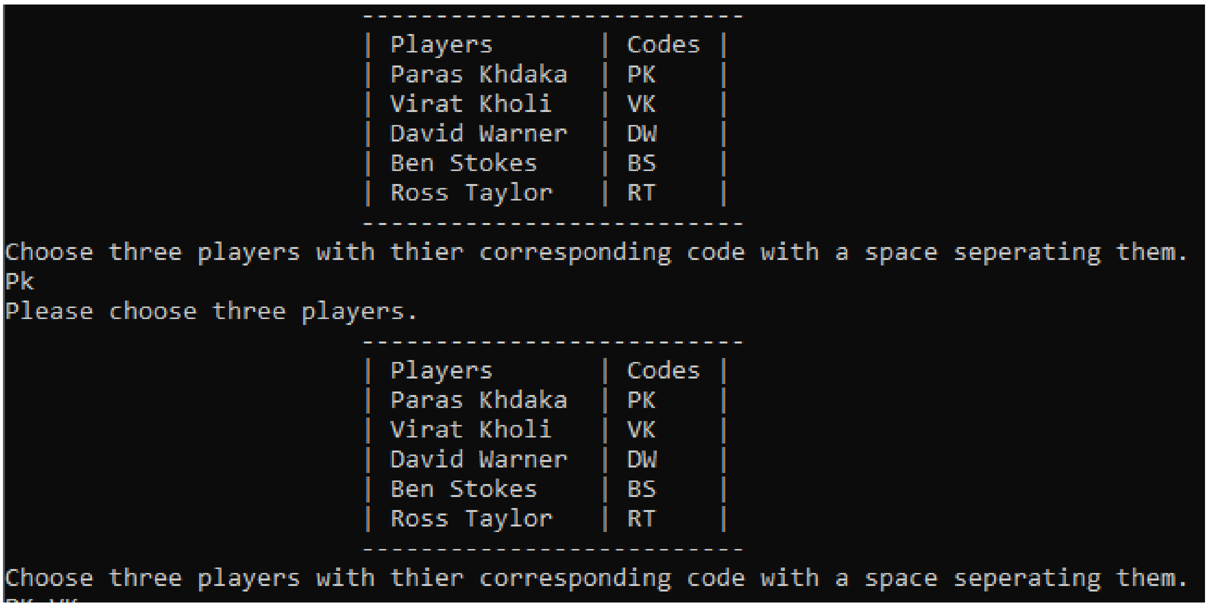


```
                        ----------------------
                        | Team           |Codes|
                        ----------------------
                        | Australia      | AUS|
                        | Bangladesh     | BAN|
                        | Nepal          | NEP|
                        | India          | IND|
                        | England        | ENG|
                        ----------------------

Guess the best cricket team:AUS
Wrong team guessed. Guess again.

Guess the best cricket team:
```

Figure 8: Test 8

Test No. 9	Testing when not all three players are chosen from five.
Input	Only one player is chosen from five,
Expected Output	The program displays a message that three players must be chosen and repeats the process.
Actual Output	A message is displayed to choose three players and repeats the process again.

Table 9: Testing no. 9

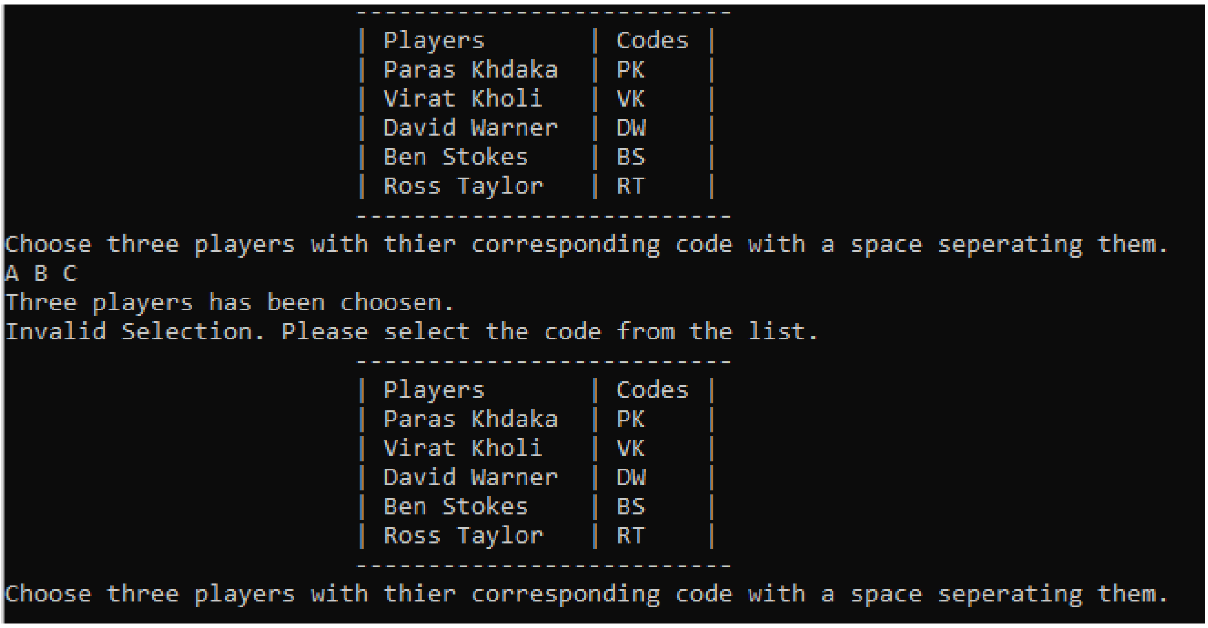


```
                        -------------------------
                        | Players        | Codes |
                        | Paras Khdaka   | PK    |
                        | Virat Kholi    | VK    |
                        | David Warner   | DW    |
                        | Ben Stokes     | BS    |
                        | Ross Taylor    | RT    |
                        -------------------------
Choose three players with thier corresponding code with a space seperating them.
Pk
Please choose three players.
                        -------------------------
                        | Players        | Codes |
                        | Paras Khdaka   | PK    |
                        | Virat Kholi    | VK    |
                        | David Warner   | DW    |
                        | Ben Stokes     | BS    |
                        | Ross Taylor    | RT    |
                        -------------------------
Choose three players with thier corresponding code with a space seperating them.
```

Figure 9: Test 9

Test No. 10	Testing when the players which are not in the list are selected.
Input	Incorrect player codes.
Expected Output	The program displays a message indicating invalid selection and asks user to select from the list.
Actual Output	A message is displayed that the selection is invalid and asks to select the code from the list by displaying the list again.

Table 10: Testing no. 10

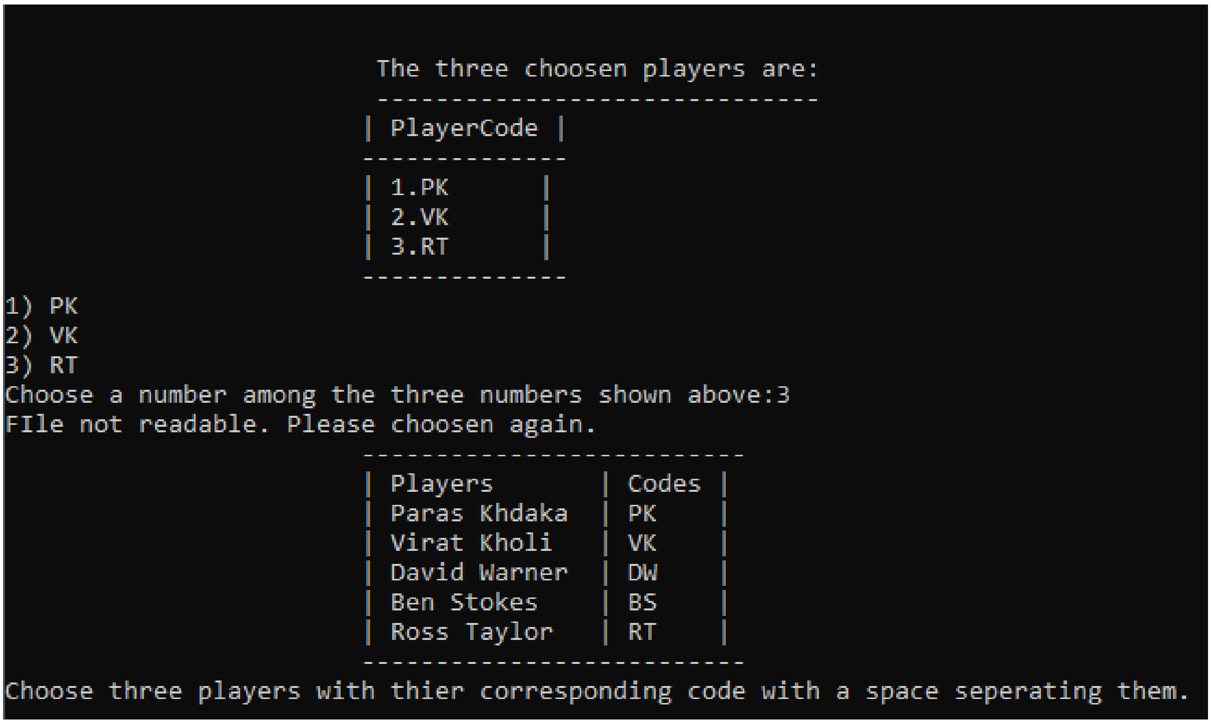


```
                                  --------------------------
                                  | Players       | Codes |
                                  | Paras Khdaka  | PK    |
                                  | Virat Kholi   | VK    |
                                  | David Warner  | DW    |
                                  | Ben Stokes    | BS    |
                                  | Ross Taylor   | RT    |
                                  --------------------------
Choose three players with thier corresponding code with a space seperating them.
A B C
Three players has been choosen.
Invalid Selection. Please select the code from the list.
                                  --------------------------
                                  | Players       | Codes |
                                  | Paras Khdaka  | PK    |
                                  | Virat Kholi   | VK    |
                                  | David Warner  | DW    |
                                  | Ben Stokes    | BS    |
                                  | Ross Taylor   | RT    |
                                  --------------------------
Choose three players with thier corresponding code with a space seperating them.
```

Figure 10: Test 10

Test No. 11	Testing when the player is chosen with no description among the three.
Input	Player code with no file is entered.
Expected Output	The program displays a message indication file is not readable and goes to process to choose three player again from five.
Actual Output	A message is displayed that the selected player file is not readable and goes to the menu of selecting three players from five.

Table 11: Testing no. 11

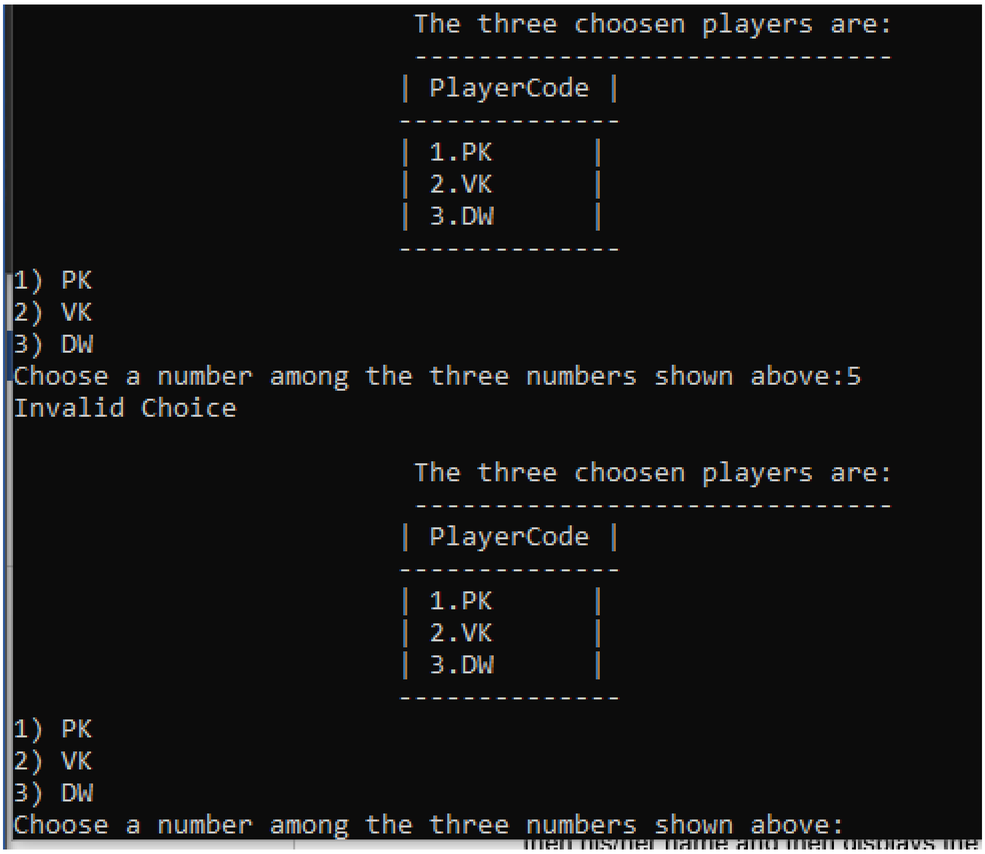


```
                         The three choosen players are:
                         ------------------------------
                        | PlayerCode |
                        --------------
                        | 1.PK      |
                        | 2.VK      |
                        | 3.RT      |
                        --------------
1) PK
2) VK
3) RT
Choose a number among the three numbers shown above:3
FIle not readable. Please choosen again.
                        --------------------------
                        | Players        | Codes |
                        | Paras Khdaka   | PK    |
                        | Virat Kholi    | VK    |
                        | David Warner   | DW    |
                        | Ben Stokes     | BS    |
                        | Ross Taylor    | RT    |
                        --------------------------
Choose three players with thier corresponding code with a space seperating them.
```

Figure 11: Test 11

Test No. 12	Testing when the number of the player that is not on the list is chosen among the three.
Input	5
Expected Output	The program displays an error message and ask user to choose again from the list.
Actual Output	A message is displayed indicating invalid choice and the menu to select the number among the three is displayed.

Table 12: Testing no. 12

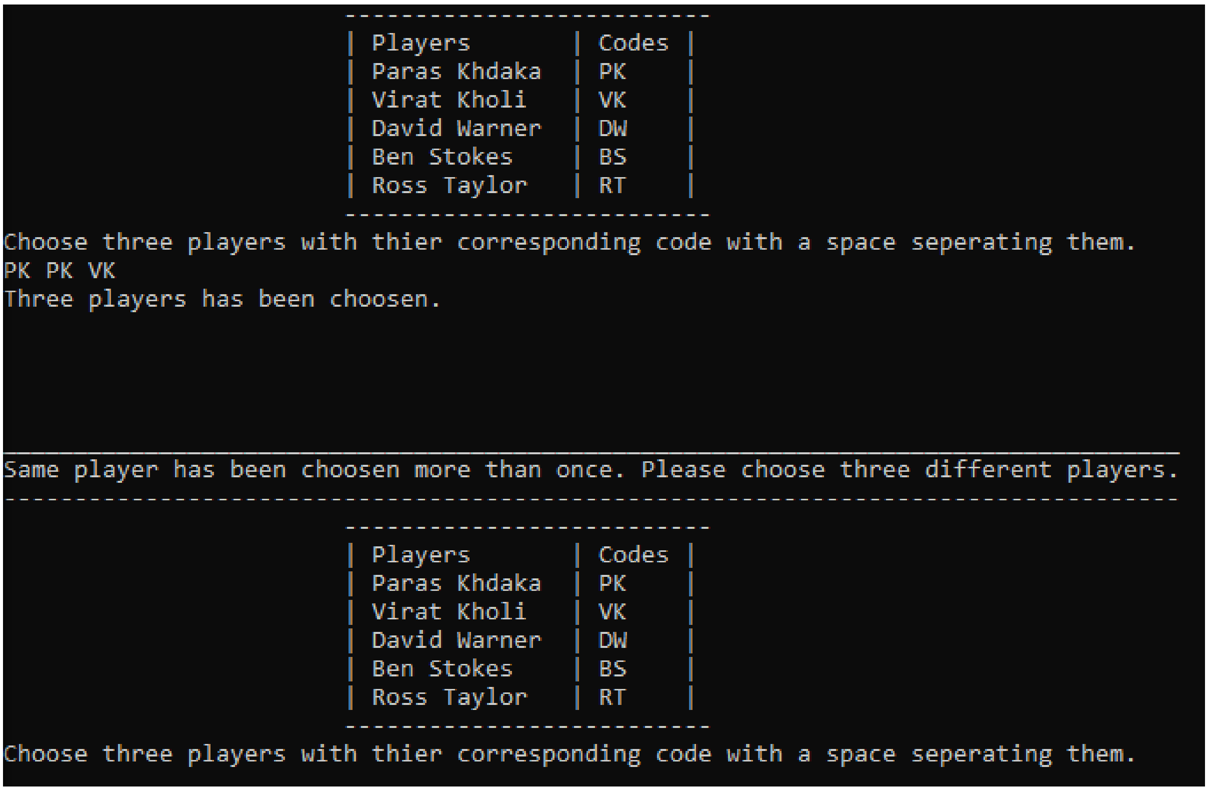


```
                          The three choosen players are:
                          ------------------------------
                         | PlayerCode |
                         --------------
                         | 1.PK       |
                         | 2.VK       |
                         | 3.DW       |
                         --------------
1) PK
2) VK
3) DW
Choose a number among the three numbers shown above:5
Invalid Choice

                          The three choosen players are:
                          ------------------------------
                         | PlayerCode |
                         --------------
                         | 1.PK       |
                         | 2.VK       |
                         | 3.DW       |
                         --------------
1) PK
2) VK
3) DW
Choose a number among the three numbers shown above:
```

Figure 12: Test 12

Test No. 13	Testing when the players are chosen more than once.
Input	PK PK VK
Expected Output	The program displays the message that the same player has been chosen more than once and acknowledges to pick three different players and runs the process again.
Actual Output	A message is displayed that same player has been chosen more than once and asks the user to choose three different players and starts the process again.

Table 13: Testing no. 13

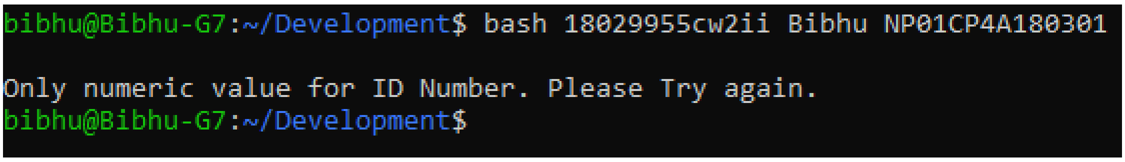


```
                                ---------------------------
                                | Players        | Codes |
                                | Paras Khdaka   | PK    |
                                | Virat Kholi    | VK    |
                                | David Warner   | DW    |
                                | Ben Stokes     | BS    |
                                | Ross Taylor    | RT    |
                                ---------------------------
Choose three players with thier corresponding code with a space seperating them.
PK PK VK
Three players has been choosen.




_______________________________________________________________________________________
Same player has been choosen more than once. Please choose three different players.
---------------------------------------------------------------------------------------
                                ---------------------------
                                | Players        | Codes |
                                | Paras Khdaka   | PK    |
                                | Virat Kholi    | VK    |
                                | David Warner   | DW    |
                                | Ben Stokes     | BS    |
                                | Ross Taylor    | RT    |
                                ---------------------------
Choose three players with thier corresponding code with a space seperating them.
```

Figure 13: Test 13

Test No. 14	Testing if the User ID is a String value.
Input	String User ID is entered.
Expected Output	The program displays an error message.
Actual Output	A message is displayed that only numeric value is taken for User ID.

Table 14: Testing no. 14

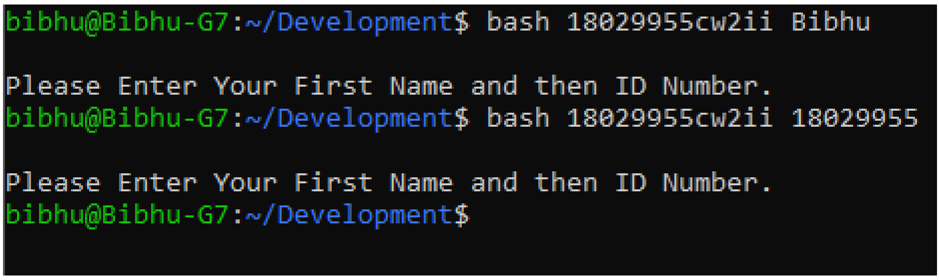


```
bibhu@Bibhu-G7:~/Development$ bash 18029955cw2ii Bibhu NP01CP4A180301

Only numeric value for ID Number. Please Try again.
bibhu@Bibhu-G7:~/Development$
```

Figure 14: Test 14

Test No. 15	Testing when only one parameter either username or userid is passed.
Input	Only one parameter is passed on.
Expected Output	The program indicates to enter username and then userid.
Actual Output	A message is displayed asking the user to enter first name and user id.

Table 15: Testing no. 15

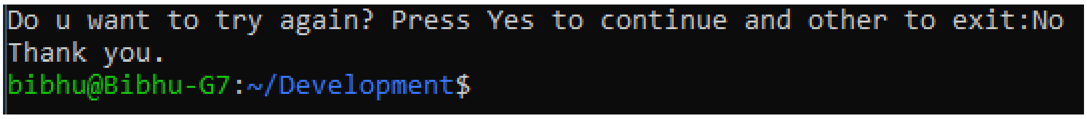


```
bibhu@Bibhu-G7:~/Development$ bash 18029955cw2ii Bibhu

Please Enter Your First Name and then ID Number.
bibhu@Bibhu-G7:~/Development$ bash 18029955cw2ii 18029955

Please Enter Your First Name and then ID Number.
bibhu@Bibhu-G7:~/Development$
```

Figure 15: Test 15

Test No. 16	Testing if the program ends when the user does not want to try again.
Input	No
Expected Output	The program ends with a message.
Actual Output	The program does end with a message.

Table 16: Testing no. 16



```
Do u want to try again? Press Yes to continue and other to exit:No
Thank you.
bibhu@Bibhu-G7:~/Development$
```

Figure 16: Test 16

Contents of three files: (TEXTS)

- PK

```
His name is Paras Khadka.
He plays for Nepal.
He was born in October 24, 1987.
He is an all rounder cricketer.
```

- VK

```
His name is Virat Kholi.
He plays for India.
He was born in November 5, 1988.
He is a right-handed top-order batsman.
```

- DW

```
His name is David Warner.
He plays for Australia.
He was born in October 27, 1986.
He is a left handed opening batsman.
```

Conclusion

In conclusion, the program required by this course work 2 was developed as to be implemented by the scenario. All the 11 steps was implemented with proper research and with working conditions. The program also uses all the control structures as required for this coursework. Whereas all the possible test cases was also carried out and presented in the tests section. The code was also developed in such a way where it is not long winded or repetitive, well commented and used with meaningful variable names, provides clear error messages and also defines and call UNIX functions. The program also works for all functions presented in the scenario for both valid and invalid inputs. The script was developed with the bash shell using the Debian system.

Task B

Introduction

In Task B of our coursework we were required to write a technical report about Memory Management which included terms such as the history of memory management, categories of memory management, memory controllers, types of memory allocation and description of paging and segmentation.

As we only know the basic things about the topic, it was essential to do a lot of research for the completion of task B. Research was done in the form of searching the web, going through different books on the topic, journals, different textbooks and outputs of organizations. A lot of effort and time was given for the completion of this particular task.

Aims and Objectives

The main aim for Task B of this coursework is to write a technical report in which the topic to be searched was on Memory Management. The main focus was on 'locality of reference' in Memory Management. Other topics regarding Memory Management was also to be researched such as Memory Allocation, Segmentation and Paging.

The objectives for achieving the above mentioned aims can be acquired by the following points:

- Research was done in the form of searching mostly the web, going through college materials, going through different online books on the topic and outputs of organizations.
- A lot of effort and time was given for the completion of this particular task.
- A lot of work was put into organization of material and quality of documentation for this task.

Background:

Memory Management

Memory management is the method of managing and organizing computer memory, assigning parts to various running programs called blocks to maximize overall system performance. Memory management occurs in hardware, OS and in applications and programs.

RAM, chips, memory caches, and flash-based SSDs which physically store data are involved in memory management for hardware. In the OS, memory management includes allocating different memory blocks and constant reallocation to individual programs as per changes to user demands. Whereas, managing the program memory incorporates two related functions, called allocation and recycling. (Rouse, 2012)

Memory Hierarchy Design

Memory Hierarchy is an improvement in computer system design to arrange the memory in such a way as to reduce the access time. The Memory Hierarchy was built based on the behaviour of a program known as reference locality. The figure below best describes the different levels of memory hierarchy:

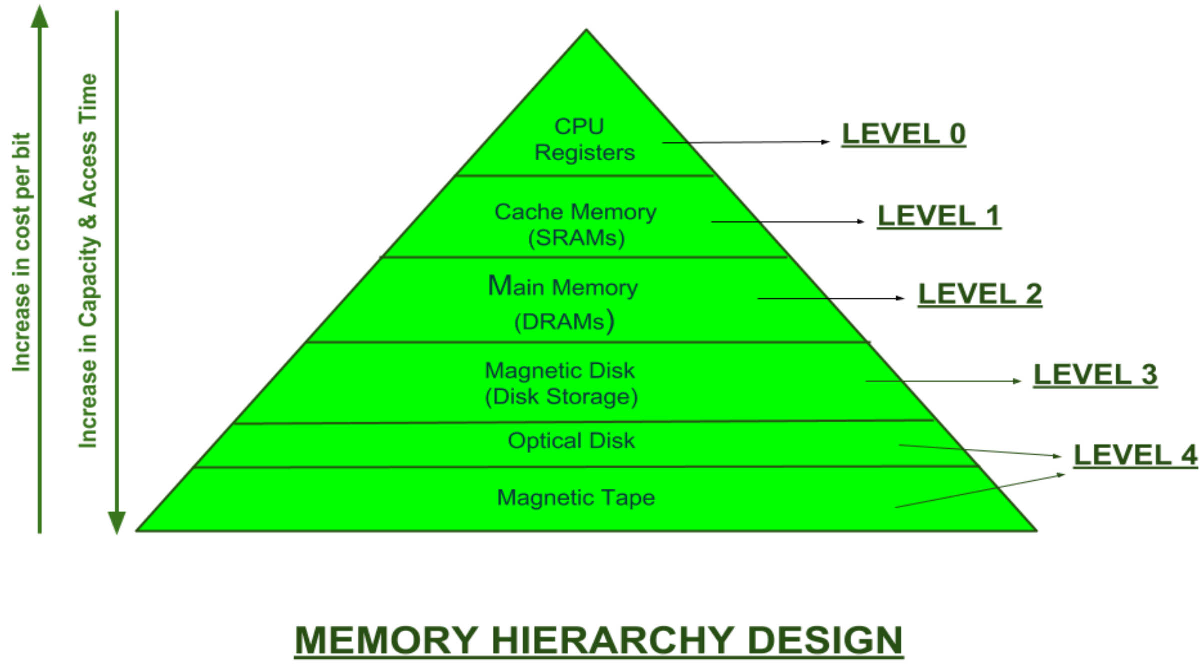


Figure 17: Memory Hierarchy Design (GeeksforGeeks, 2020)

This memory Hierarchy Design is divided into 2 main types which are External Memory or Secondary Memory that are peripheral storage devices that are accessible via I/O module to the processor for example Primary Memory which is directly accessible by the processor comprising of Main Memory, Cache Memory and CPU. (GeeksforGeeks, 2020)

Locality of reference

The phenomenon where a computer program appears to access the same set of memory locations over a given time span is referred to as Locality of reference. In other words, Locality of reference refers to the Computer programs ability to access instructions whose addresses are close to each other. The property of the locality of reference is seen primarily by loops and subroutine calls in the system. (Singh, 2018)

Caches are the faster memory that is designed to cope with the Processor-Memory gap in the data read operation which means the time difference in the data read operation on the CPU register and in the main memory. Caches are used as a temporary staging area for a subset of data and instructions stored in a comparatively sluggish main memory. Only data which are mostly used by the processor during program execution is stored in cache because the cache size is small. Caching this frequently used CPU data removes the need to repeatedly bring the data out of the slower main memory which takes hundreds of CPU cycles. (Tutorialspoint.dev, 2019)

The principle of storing the useful data centres around a central property of computer programs is known as locality. Efficient locality programs continue to access the same set of data items from the upper memory hierarchy levels such as cache over and over again, and therefore run faster. There are two types of locality which are:

i) Temporal locality

Temporal locality means that it will be appropriate to obtain current data or instructions in the near future. So we can store the data or instruction in the cache memory so we can stop looking for the same data again in main memory. Once CPU accesses the current main memory location for reading needed data or instructions, it will also be stored in the cache memory based on the assumption that in the near future the same data or instructions will be needed. This location is known as temporal locality. If any data are referenced, then it is highly likely to be referenced again in the near future.

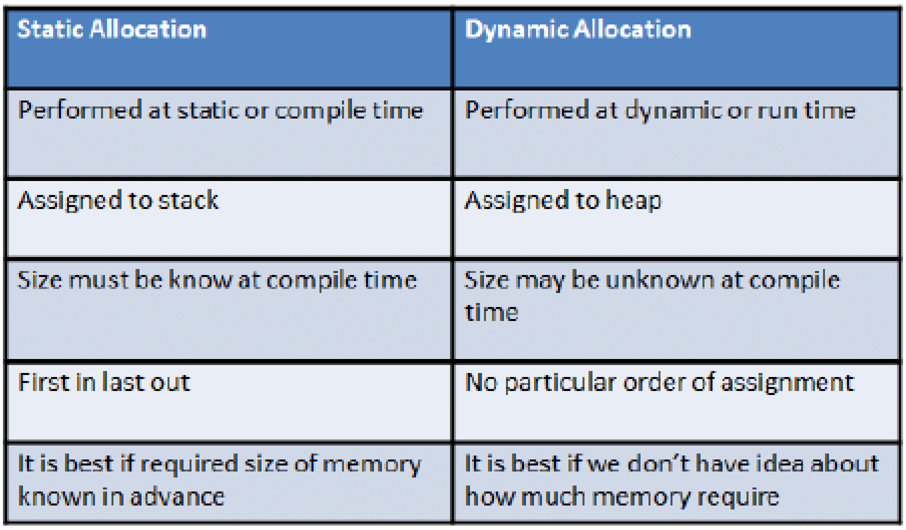
ii) Spatial locality

Spatial locality means instructions or data near the current position of the memory being stored in the near future may be required. That is slightly different from the temporal locality. Here we are thinking about nearly located memory locations when we were thinking about the actual location of information that was being stored in temporal locality. (Singh, 2018)

Memory Allocation

Memory allocation is mainly a hardware device operation but is handled by software and operating system applications. Simply, assigning space to a process in memory is called memory allocation. In physical and virtual memory management the method of allocating memory is very close. When executed, the programs and facilities are allocated with a common memory according to their requirements. The memory is released and transferred to another program or combined within the primary memory until the program has finished its operation or is idle (Techopedia Inc, 2020). There are two core types of Memory allocation which are:

- Static Memory Allocation: The program is allocated memory at compile time.
- Dynamic Memory Allocation: The programs are allocated with memory at run time



Static Allocation	Dynamic Allocation
Performed at static or compile time	Performed at dynamic or run time
Assigned to stack	Assigned to heap
Size must be know at compile time	Size may be unknown at compile time
First in last out	No particular order of assignment
It is best if required size of memory known in advance	It is best if we don't have idea about how much memory require

Figure 18: Static Vs Dynamic (Irabashetti, 2014)

Static Memory Allocation

When the compiler compiles the program and produces object files, the static memory allocation is performed, the linker merges all these object files and creates a single executable file, and the loader loads this single executable file into main memory for execution. In static memory allocation, the size of the data that the process needs must be known before the process initiates execution. Execution process is faster as static method of allocating memory does not require any operation of allocating memory during process execution. As all the memory allocation operation needed for the process is performed before process execution has begun. And, this leads to a procedure being executed more efficiently and also efficient when compared to the dynamic memory allocation. (T, 2019)

Dynamic Memory Allocation

Dynamic memory allocation is done while the program is being run. Here, as they are to be used for the first time when the program is running the memory is allocated to the program entities. The actual size of the necessary data is determined at runtime, so the exact memory space is allocated to the program, thus minimizing memory waste. Dynamic memory allocation gives flexibility in program execution. As it can determine which amount of memory space the program should need. If the program is big enough then on the various parts of the program a dynamic memory allocation is performed which is to be used at present. This reduces memory wastage and increases device efficiency. It is up to the programmer to design the program in the way that takes advantage of the dynamic memory allocation method as it does not require special operating system support. So, dynamic memory allocation is slower than static memory allocation but is more flexible. (T, 2019)

The operating system will maintain a list of each memory location indicating which are free and which are busy for both fixed and dynamic memory allocation schemas. The free partitions will then be distributed as new jobs come into the network. Those partitions can be distributed in four ways:

- *First – Fit*

This system preserves free / busy list of jobs sorted by location of the memory, low-ordered to high-ordered memory. First job claims in this process the first usable memory with capacity greater than or equal to the size of it. The operating system does not check for the right partition but instead allocates the job to the nearest accessible memory partition with adequate capacity. It is fast in both execution and processing as the processor allocates the nearest memory partition available to the task. The huge disadvantage of this memory allocation is that it wastes a lot of memory as the processor ignores whether or not the size of the partition allocated to the job is very big as compared with the size of the task and as of this a lot of memory is lost and several jobs would not get memory space, and will have to wait for another job to be done. (Singhal, 2020)

- *Best – Fit*

This approach holds the list free / busy by scale-from smallest to largest. In this process, the operating system first scans the entire memory according to the size of the specified job and allocates it to the memory's closest-fitting free partition, allowing efficient memory usage. The jobs are in order here, from the smallest job to the largest job. It is the best method for saving memory from being wasted as it is memory efficient as the OS allocates minimal available memory space to the job. The main disadvantage of this method is a slow moving operation since checking the entire memory for each job makes OS run very slow. (Singhal, 2020)

- *Worst – Fit*

In this method the algorithm looks for free-space in memory in which the necessary information can be processed. The algorithm selects and stores the largest available free space in which the information can be stored (i.e., this is greater than the information that needs to be stored). It is in directly opposition to the best fit algorithm that scans the memory in almost the same way as before. This method works best when allocations are of medium sizes. The main disadvantage of this method is external fragmentation and tends to break large free blocks such that large partitions cannot be allocated. (Amjad, 2018)

- *Next – Fit*

Next fit is a modified version of first fit. It begins as first fit to find a free partition. When called next time it starts searching from where it left off, not from the beginning. The strategy uses a roving pointer which travels down the memory chain to try a next fit. It helps in preventing the use of memory from the free block chain's head (beginning). Next-fit attempts to tackle the problem of first-fit by starting the search for free portions of parts not from the beginning of the memory but from where it finishes last. Next fit is a very fast searching algorithm and is also comparatively faster than First Fit and Best Fit Memory Management Algorithms. (Gupta, 2020)

Description of Paging and Segmentation

Paging is a storage mechanism used in Operating Systems to retrieve processes in the form of pages from the secondary storage into the main memory. The key concept behind paging is to break up each process in the form of pages which also includes the main memory to be divided into the form of frames. One page of the cycle needs to be stored in one of the memory frames. The pages may be placed at various memory locations but it is always the goal to locate the contiguous frames or gaps. Process pages are only brought into the main memory when necessary otherwise they reside in the secondary storage. Different operating systems describe various frame sizes. Considering that in paging the pages are mapped to the frames with equal size, the page size has to be the same as the size of the frame. (JavaTpoint, 2018)

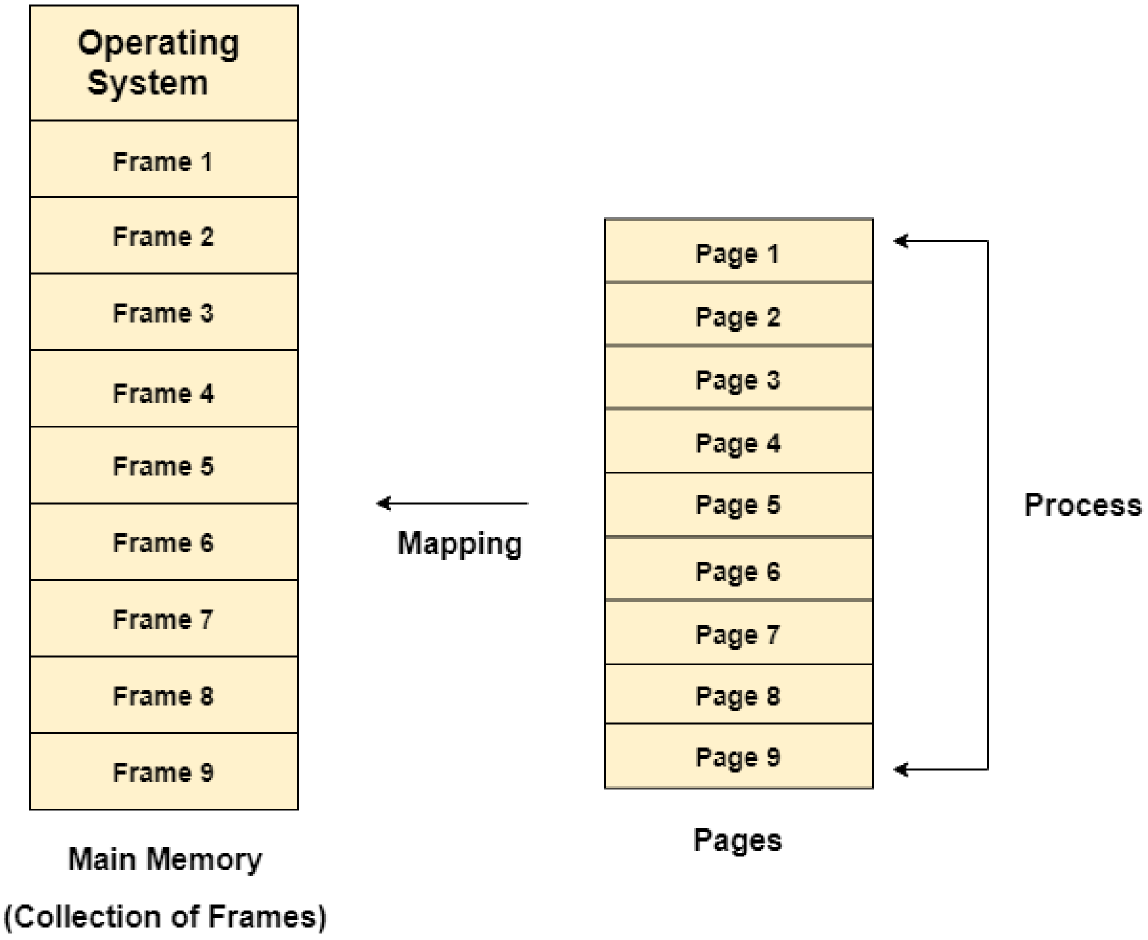


Figure 19: Paging (JavaTpoint, 2018)

Segmentation is a memory management method in Operating Systems in which the memory is separated into sections of the variable size. That component is known as a segment that can be allocated to a process. In a table called as a segment table, the data about each segment are stored. Segment table is stored in one or many segments. Base and Limit are the two information that contains in a segment table. Better have segmentation than paging as it separates the process into segments in which each segment contain same type of functions as the main function and can be included in one section, so it can include library functions.

CPU produces a logical address with two parts Segment Number and Offset. The Segment Number is mapped to the table of segment. The limit is compared to the offset for the respective segment. Unless the offset is less than the limit, the address is correct otherwise, if the address is invalid, it will cause an error. Whereas in the case of valid address, to get the physical address of the actual word in the main memory, the base address of the segment is applied to the offset. (JavaTpoint, 2018)

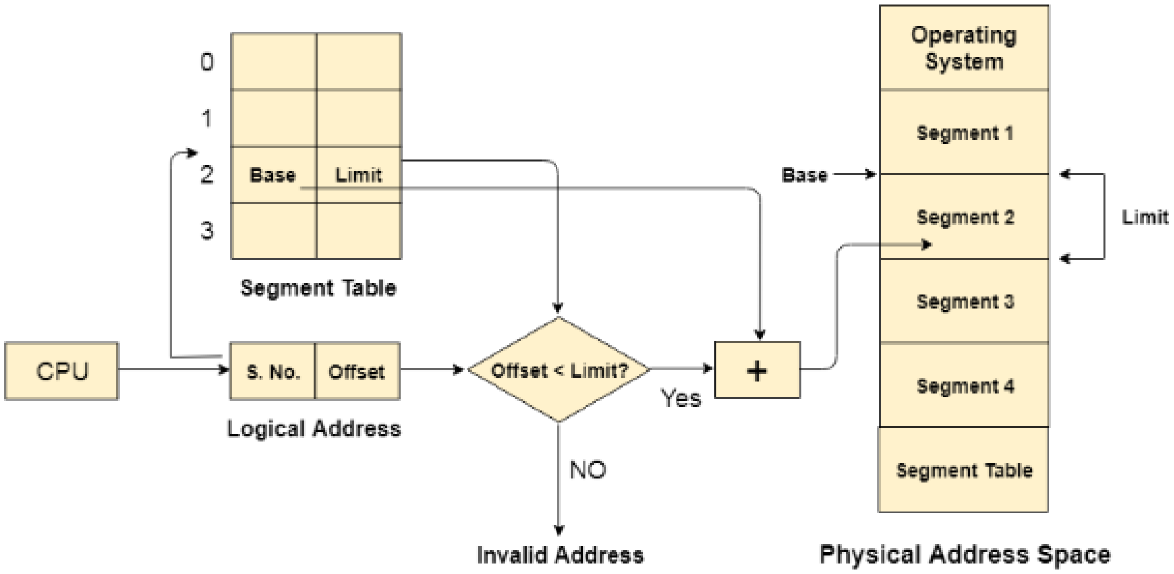


Figure 20: Segmentation (JavaTpoint, 2018)

Conclusion

In conclusion, memory management manages and organizes computer memory to maximize overall system performance. Whereas Locality of reference in memory management refers to the Computer programs ability to access instructions whose addresses are close to each other. Assigning space to a process in memory is called memory allocation and there are two types of them which are Static and dynamic memory allocation each having their own use and rules. The key concept behind paging is to break up each process in the form of pages whereas break up each process into sections of the variable size in segmentation.

References

Techopedia Inc, 2020. *What is Memory Allocation? - Definition from Techopedia.* [Online]
Available at: https://www.techopedia.com/definition/27492/memory-allocation
[Accessed 4 April 2020].

Amjad, S., 2018. *First fit , Best fit, Worst fit.* [Online]
Available at: https://www.slideshare.net/ShahzebAmjad/first-fit-best-fit-worst-fit
[Accessed 5 April 2020].

GeeksforGeeks, 2020. *Memory Hierarchy Design and its Characteristics - GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/memory-hierarchy-design-and-its-characteristics/
[Accessed 3 April 2020].

Gupta, A., 2020. *Program for Next Fit algorithm in Memory Management - GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/program-for-next-fit-algorithm-in-memory-management/
[Accessed 4 April 2020].

Irabashetti, P., 2014. *Difference between static and dynamic allocation | Download Scientific Diagram.* [Online]
Available at: https://www.researchgate.net/figure/Difference-between-static-and-dynamic-allocation_fig2_265166374
[Accessed 4 April 2020].

JavaTpoint, 2018. *OS Paging with Example - javatpoint.* [Online]
Available at: https://www.javatpoint.com/os-paging-with-example
[Accessed 5 April 2020].

JavaTpoint, 2018. *OS Segmentation - javatpoint.* [Online]
Available at: https://www.javatpoint.com/os-segmentation
[Accessed 5 April 2020].

M.Stonebank, 2000. *UNIX Tutorial - Introduction.* [Online]
Available at: http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html
[Accessed 27 March 2020].

Rouse, M., 2012. *What is memory management? - Definition from WhatIs.com.* [Online]
Available at: https://whatis.techtarget.com/definition/memory-management
[Accessed 3 Apri 2020].

Rouse, M., 2018. *What is virtual memory? - Definition from WhatIs.com.* [Online]
Available at: https://searchstorage.techtarget.com/definition/virtual-memory
[Accessed 6 April 2020].

Singhal, T., 2020. *Best-Fit Allocation in Operating System - GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/best-fit-allocation-in-operating-system/?ref=rp
[Accessed 5 April 2020].

Singhal, T., 2020. *First-Fit Allocation in Operating Systems - GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/first-fit-allocation-in-operating-systems/
[Accessed 5 April 2020].

Singh, B., 2018. *Locality of Reference and Cache Operation in Cache Memory - GeeksforGeeks.* [Online]
Available at: https://www.geeksforgeeks.org/locality-of-reference-and-cache-operation-in-cache-memory/
[Accessed 4 April 2020].

T, N., 2019. *What is Memory Allocation in Operating System? Static and Dynamic Memory Allocation - Binary Terms.* [Online]
Available at: https://binaryterms.com/static-and-dynamic-memory-allocation.html
[Accessed 4 April 2020].

Tutorialspoint.dev, 2019. *Computer Organization | Locality and Cache friendly code - Tutorialspoint.dev.* [Online]
Available at: https://tutorialspoint.dev/computer-science/computer-organization-and-architecture/computer-organization-locality-and-cache-friendly-code
[Accessed 4 April 20202].

Vgevge, 2020. *Computer Memory Controllers: How They Work.* [Online]
Available at:
https://www.streetdirectory.com/travel_guide/124468/hardware/computer_memory_controllers_how_they_work.html
[Accessed Apirl 6 2020].

Appendix

Memory Controller

The memory controller chip's main function is to read, write, and update the Random Access Memory by transmitting the current over the entire device. If there was no device to refresh the RAM, the application could lose all the data that it has stored. The existence of a memory controller is of absolute significance in this context.

The RAM is read and written by the Memory controller via multiplexers and demultiplexers. For the data, they select the right row, column, and memory location. Those controllers operate by bus width. A system's bus width is determined by the number of parallel lanes communicating with the RAM's memory cell.

Controllers for the machine memory have grown into a variety of categories. There are memory with double data rate, dual channel memory, and memory buffered to the maximum. The Double data rate memory controller is used to control the RAM in the form of an integrated circuit that is used in the computer. Here, the information is transmitted in the highs and lows rhythm of the computer's memory clock. They are a little more detailed than a single rate controller, since they allow the data to be transmitted almost twice.

Dual channel memory controllers, on the other hand, are used in the systems where RAM is split into two separate buses. This allows two controllers to operate in parallel directions. Fully buffered memory means putting a buffer memory device on each memory module which uses a serial connection to the data instead of a parallel link. Multiple memory devices can be attached to a single board with the aid of this computer memory controller. (Vgevge, 2020)

Virtual Memory

Virtual memory is an operating system (OS) memory management technology that uses hardware and software to allow a device to compensate for physical memory limitations by moving data from random access memory (RAM) to disk storage for a temporary time. Using active memory in RAM and inactive memory in hard disk drives (HDDs), virtual address space is increased to form contiguous addresses that contain both the program and its data.

Virtual memory was created at a time when the cost of physical memory — the RAM mounted — was high. Computers have a finite amount of RAM, so it can run out of memory, particularly when multiple programs run concurrently. A program that uses virtual memory to simulate RAM uses a portion of the hard disk. A device can load larger programs or several programs running at the same time with virtual memory, allowing it to function as if it had unlimited memory and without needing to buy more RAMs.

The OS splits memory into page files when copying virtual memory into physical memory, or switches files with a set number of addresses. Page is stored on a disk and the OS copies it from the disk to the main memory when the page is required, and translates the virtual addresses into real addresses.

A computer's memory management unit (MMU) handles memory operations, including managing virtual memory. In most computers, the MMU hardware is integrated into the CPU. There are two ways in which virtual memory is handled: paged and segmented. (Rouse, 2018)