

Networking and Operating System

Prepared by: Bidhan Pokhrel
Date: 16-Apr-2020

Contents

1. Task A	1
1.1 Introduction.....	1
1.1 Script.....	1
1.2 Testing.....	15
1.2.1 Test 1.....	15
1.2.2 Test 2	16
1.2.3 Test 3	17
1.2.4 Test 4	18
1.2.5 Test 5	19
1.2.6 Test 6	20
1.2.7 Test 7	21
1.2.8 Test 8	21
1.2.9 Test 9	22
1.2.10 Test 10.....	23
1.2.11 Test 11.....	24
1.2.12 Test 12.....	25
1.2.13 Test 13.....	26
1.2 Conclusion	27
2. Task B	28
2.1 Introduction.....	28
2.2 Efficiency.....	31
2.3 Implementation & application on different areas	32
2.4 Physical memory management in network & data storage	33
2.5 Memory Management in operating system.....	34
2.6 Virtual Memory concept with uses and development	35
2.7 Conclusion	37
2.8 References	38

List of Figures

Fig 1.....	15
Fig 2.....	16
Fig3.....	17
Fig 4.....	18
Fig 5.....	19
Fig 6.....	20
Fig 7.....	21
Fig 8.....	21
Fig 9.....	22
Fig10.....	23
Fig 11.....	24
Fig 12.....	25
Fig 13.....	26

1. Task A

1.1 Introduction

The main goal of this task is to work on UNIX shell-based script programming.

Ubuntu release 18.10 was used as an alternative to Unix framework. We used a virtual box to enable the Ubuntu unity desktop to run another operating system as an application on our Window PC. Windows and Ubuntu both do the same kind of function on a PC, but we just can't run all at once. However,our machine can be programmed to run "dual-boot," On disk (a new partition) one should make space for Windows to install Windows apart from Ubuntu. We can choose between running Ubuntu or windows at boot-time.

1.1 Script

```
#!/bin/bash
```

```
function login {
```

```
    clear
```

```
    echo "Fill your login details"
```

```
    echo "-----"
```

```
    #for validation of program name input
```

```
    validProgramName=false
```

```
    #while program name is invalid
```

```
    while [ $validProgramName == false ]
```

```
    do
```

```
        #ask program name
```

```
    echo -n "Enter Program name: "

    readprogramName

    if [[ -z $programName ]]

    then

        echo "-----"

        echo "Program name is empty"

        echo "Please enter your program name"

        echo "-----"

    else

        validProgramName=true

    fi

done

validUserName=false

#while user name is invalid

while [ $validUserName == false ]

do

    #ask user name

    echo -n "Enter user name: "

    read username

    if [[ -z $username ]]

    then
```

```
        echo "-----"

        echo "user name is empty"

        echo "Please enter your user name"

        echo "-----"

    else

        validUserName=true

    fi

done

validId=false

#until id is invalid

while [ $validId == false ]

do

    #ask id

    echo -n "Enter ID: "

    read id

    if [[ -z $id ]]

    then

        echo "-----"

        echo "ID is empty"

        echo "Please enter your id"
```

```
        echo "-----"

    else

        validId=true

    fi

done

}

function password {

    #hold for 0.5 seconds

    sleep 0.5

    #clear the screen

    clear

    echo "5 chances to guess secret key"

    echo "-----"

    num_of_login=1

    login_left=4

    while [ $num_of_login -lt 6 ]

    do

        #ask secret key

        echo "Enter secret key: "

        read -s key
```

```
#if secret key is correct

if [ "$key" = "bdn" ]

then

    echo "-----"

    #logged in successful

    echo "You are logged in successfully"

    echo "-----"

    break

else

    #try again

    echo "-----"

    echo "Please try again"

    echo "You have $login_left chances left"

    echo "-----"

    ((login_left--))

fi

((num_of_login++))

done

#if number of login left equals to 0

if [ $num_of_login -eq 6 ]

then
```



```
#inform user cannot login more

echo "-----"

echo "You cannot log in more"

echo "-----"

#exit the system

exit 0

fi

}

function welcome {

    sleep 0.5

    clear

    #welcome username displaying its id and date of execution

    echo "-----"

    echo "Welcome to the World Cup Prediction Board"

    echo

    echo "          ID: $id"

    echo "          Username:$username"

    echo

    echo "Date of execution: $(date)"

    echo "-----"
```

```
}
```

```
functionbest_country {  
    while [ true ]  
    do  
        #display menu  
        echo "-----"  
        echo "|   Code       |   Country   |"  
        echo "-----"  
        echo "|   AUS        |   Australia |"  
        echo "|   BAN        |   Bangladesh|"  
        echo "|   NEP        |   Nepal    |"  
        echo "|   IND        |   India    |"  
        echo "|   ENG        |   England  |"  
        echo "-----"  
        #ask user best cricket team  
        echo -n "Enter code of best cricket team: "  
        read code  
        case $code in  
            BAN)  
                echo "Correct"
```

```
echo "
```

Bangladesh is a country in Southern Asia and is located on the Bay of Bengal bordered by India on all sides except for a small border with Burma. Bangladesh has flat plains, and most of the country is situated on deltas of large rivers flowing from the Himalayas."

```
break
```

```
::
```

```
AUS | NEP | IND | ENG)
```

```
echo "Wrong!!!"
```

```
echo "Guess again"
```

```
echo
```

```
::
```

```
*)
```

```
echo "Enter code above table next time"
```

```
::
```

```
esac
```

```
sleep 0.5
```

```
clear
```

```
done
```

```
}
```

```
function player {
```

```
    sleep 0.5
```

```
clear

echo "The names and codes of five star players of Bangladesh are:"

echo "-----"

echo "|   Code       |   Player Name   |"

echo "-----"

echo "|   PK         |   Paras Khadka  |"

echo "|   VK         |   ViratKholi    |"

echo "|   DW         |   David Warner  |"

echo "|   BS         |   Ben Stokes    |"

echo "|   RT         |   Ross Taylor   |"

echo "-----"

playerList=('PK' 'VK' 'DW' 'BS' 'RT')

while :

do

    emptyPlayerFound=false

    playerFound=false

    playerItemCount=0

    read -p "Choose Three Players: " playerInputs[0] playerInputs[1]
playerInputs[2]
```

```
for (( i=0; i<3; i++ ))
do
    for (( j=0; j<5; j++ ))
    do
        if [[ ${playerInputs[$i]} = ${playerList[$j]} ]]

            then

                (( playerItemCount++ ))

            fi
        done
    done

done

if [ $playerItemCount -eq 3 ]
then
    playerFound=true
fi

for (( i=0; i<3; i++ ))
```

```
do

    if [[ -z ${playerInputs[i]} ]]

    then

        emptyPlayerFound=true

        break

    fi

done

if [[ $emptyPlayerFound == true ]]

then

    echo "Please enter three  Players"

elif [[ $playerFound == true ]]

then

    selectOnePlayer

    break

else

    echo "Player Not Found"

fi
```

```
done

}

functionselectOnePlayer {

    echo "Choose one of the players code: "

    selectplayer_selected in ${playerInputs[0]} ${playerInputs[1]} ${playerInputs[2]}

    do

        case $player_selected in

            ${playerInputs[0]}|${playerInputs[1]}|${playerInputs[2]})

                chooseFile

                break

                ;;

            *)

                echo "-----"

                echo "| Please enter 1 or 2 or 3. |"

                echo "-----"

                ;;

        esac

    done

}
```

```
function chooseFile {  
  
    #extract content of specified file  
  
    playerDetails=$(find . -name "$player_selected")  
  
    #check if the file is matched  
  
    if [ -f "$playerDetails" ]  
  
    then  
  
        #display the content of the file  
  
        cat $playerDetails  
  
    else  
  
        echo "Sorry, we are unable to open the specified file."  
  
    fi  
  
    echo ""  
  
}
```

#running the program

login

password

welcome

while [true]

do

best_country


```
player

#ask user to repeat above steps

echo -n "Do you want to execute program again?(enter yes or no) "

read choice

if [[ "$choice" = "YES" || "$choice" = "yes" ]]

then

    echo "Executing the program again"

    sleep 0.5

    clear

else

    echo "-----"

    echo "|   Thanks for using my program   |"

    echo "-----"

    break

fi

done
```

1.2 Testing

1.2.1 Test 1

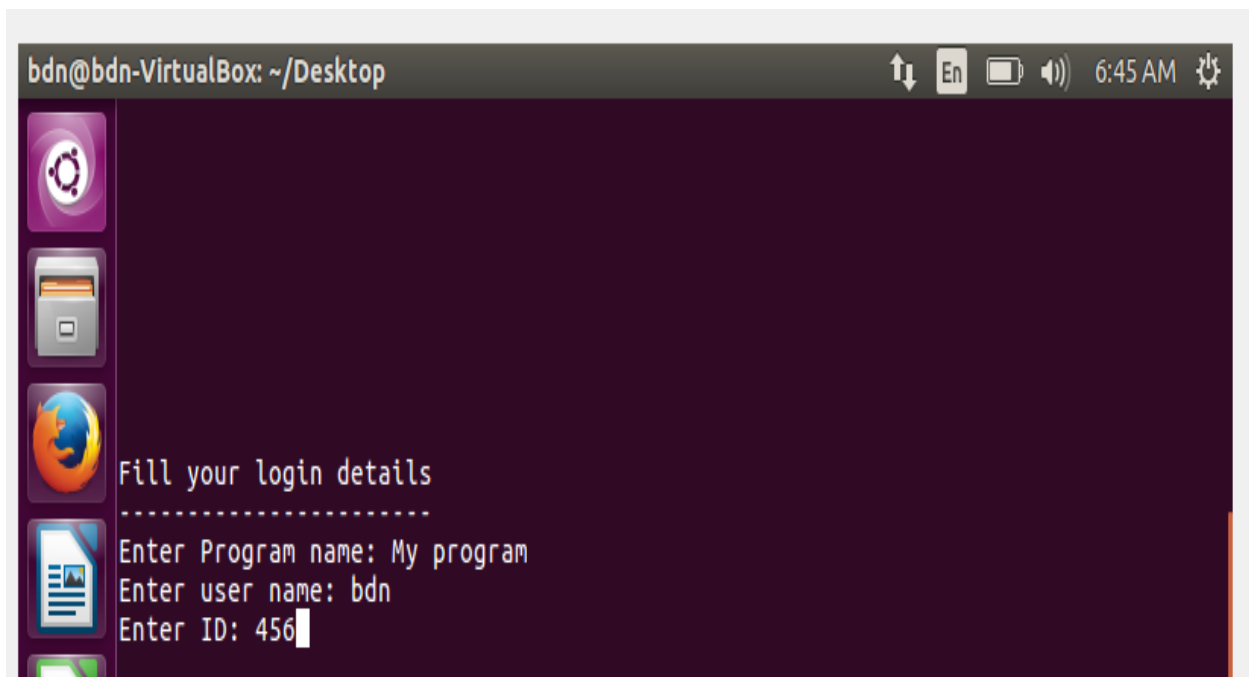


FIG:1

Objective	A user starts the program typing the program name and first his/her first name and then the ID number as two parameters of the program
Result	Objective Fulfilled

1.2.2 Test 2

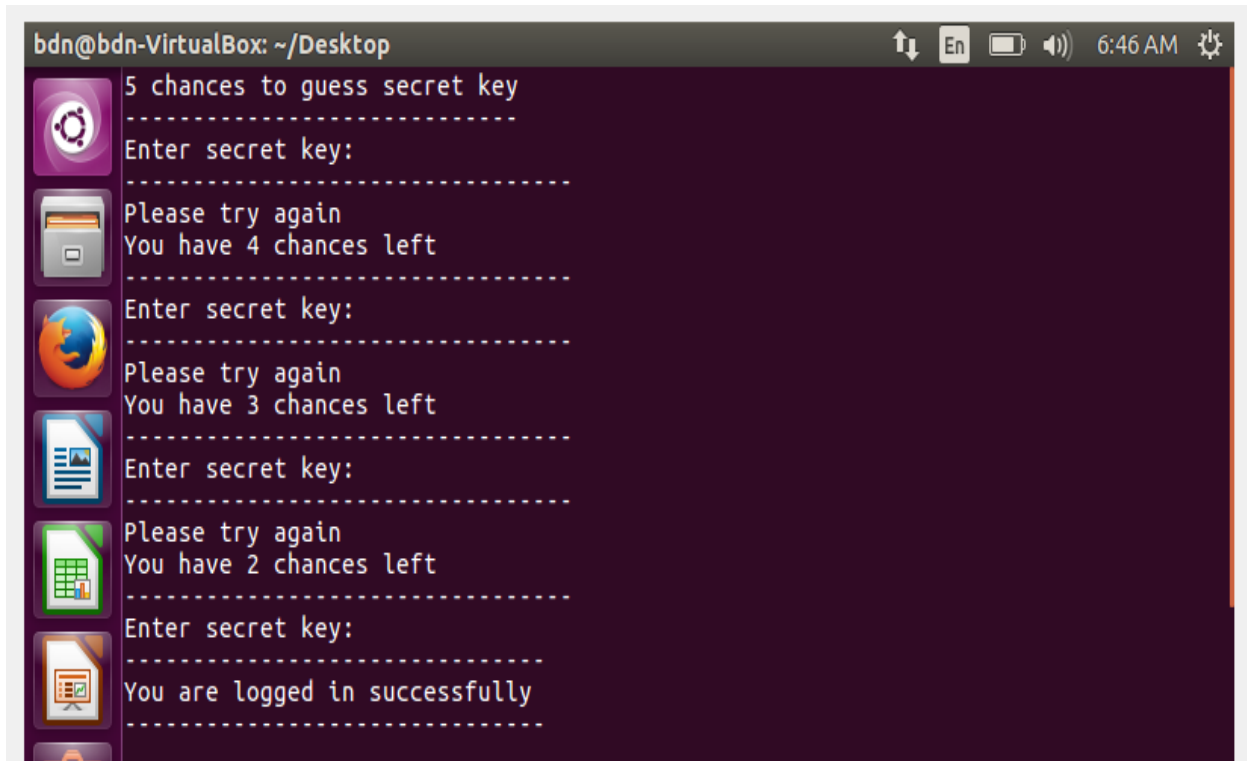
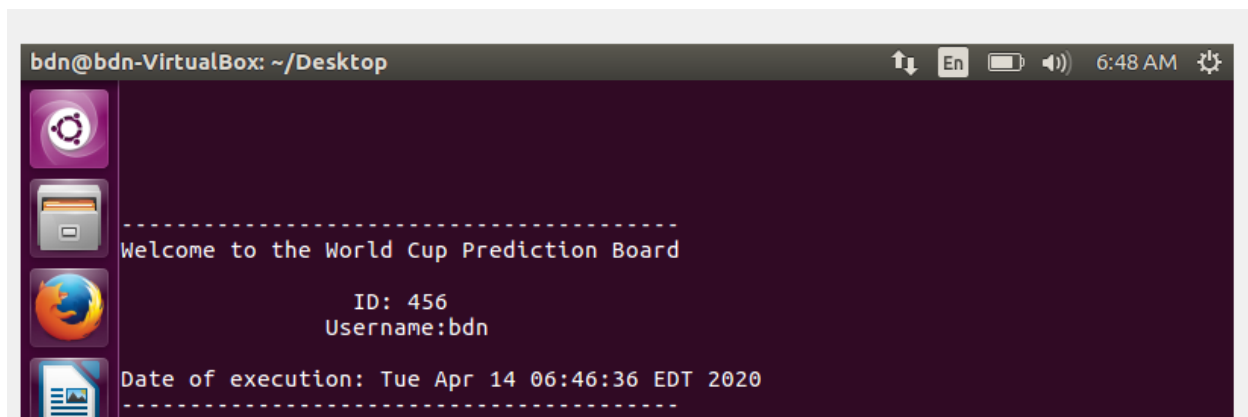


FIG: 2

Objective	Program shall ask the secret key to run the program; a user should program this secret key in advance. If the user enters correct secret key it should move to next step (3), else it should prompt to enter correct key for five times and then exit the program
Result	Objective Fulfilled

1.2.3 Test 3



```
bdn@bdn-VirtualBox: ~/Desktop
-----
Welcome to the World Cup Prediction Board
      ID: 456
Username: bdn
Date of execution: Tue Apr 14 06:46:36 EDT 2020
-----
```

FIG: 3

Objective	The program welcomes the user somehow mentioning his/her ID number first, then his/her name and then displays the date and the time of the execution.
Result	Objective Fulfilled

1.2.4 Test 4

```

bdn@bdn-VirtualBox: ~/Desktop
-----
Welcome to the World Cup Prediction Board
      ID: 456
  Username: bdn
Date of execution: Tue Apr 14 06:46:36 EDT 2020
-----
| Code | Country |
|-----|-----|
| AUS  | Australia |
| BAN  | Bangladesh |
| NEP  | Nepal |
| IND  | India |
| ENG  | England |
|-----|-----|
Enter code of best cricket team: █

```

FIG: 4

Objective	The program displays five codes and names of the five countries team that has played Cricket (one code and name per line) and asks the user to guess which the one is best cricket team. The codes for Australia, Bangladesh, Nepal, India and England are AUS, BAN, NEP, IND and ENG respectively.
Result	Objective Fulfilled

1.2.5 Test 5

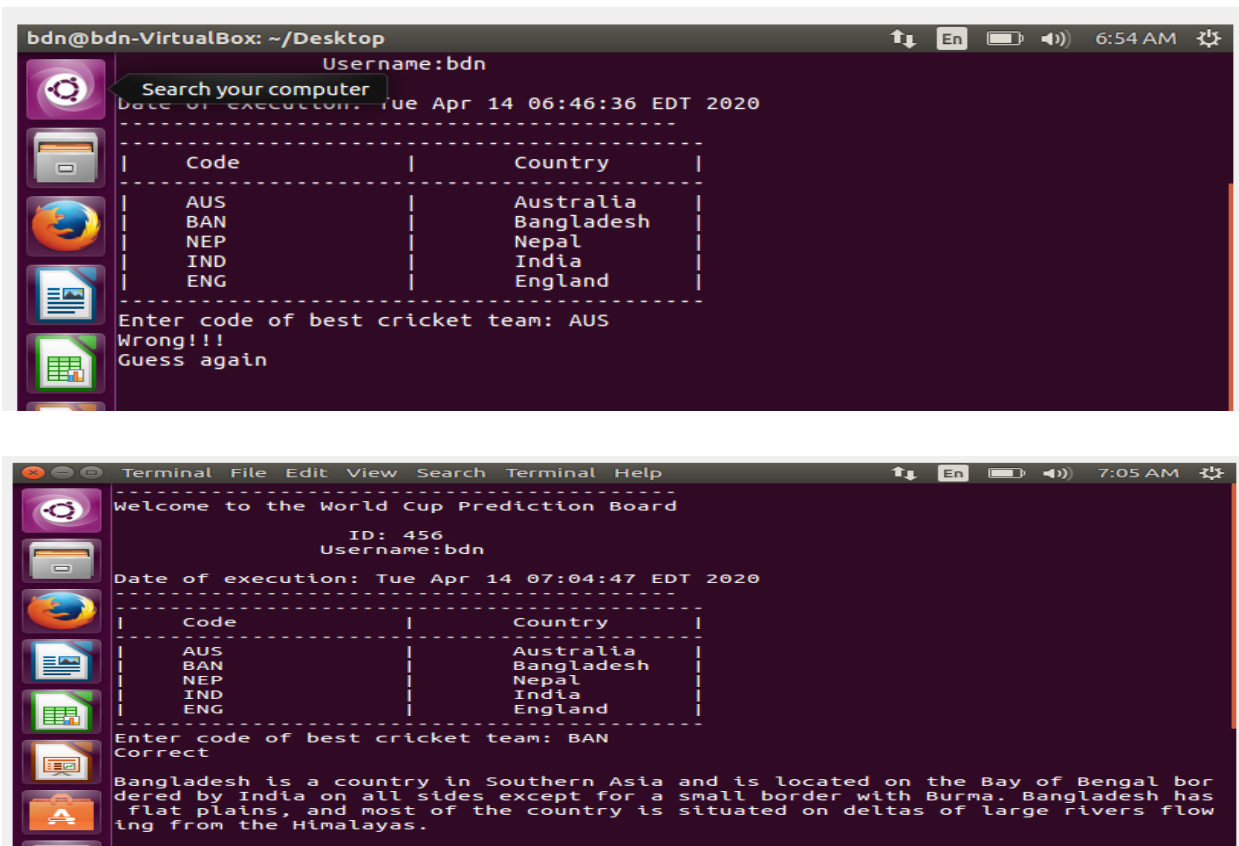


FIG:5

Objective	After the user typed the country code, if the option is correct, the program informs about it and gives one two sentences explaining about some of the descriptions/ characteristics of the chosen team, otherwise the program warns about a mistake, and the user will guess until the correct option is chosen.
-----------	---

Result	Objective Fulfilled
--------	---------------------

1.2.6 Test6



FIG: 6

Objective	The program displays Five players of the Cricket team. The codes for five-star players at Cricket are Paras Khadka, ViratKohli, David Warner, Ben Stokes and Ross Taylor are PK, VK, DW, BS, RT respectively.
Result	Objective Fulfilled.

1.2.7 Test 7

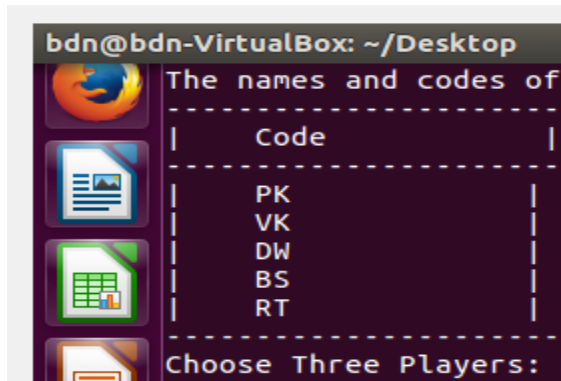


FIG: 7

Objective	program invites to type three codes out of these five.
Result	Objective Fulfilled

1.2.8 Test 8

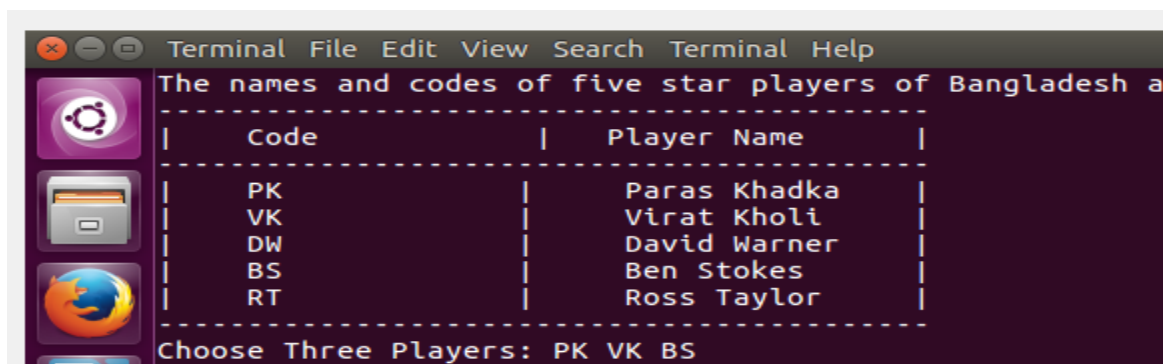


FIG: 8

Objective	The user types these codes separating
-----------	---------------------------------------

	them by spaces.
Result	Objective Fulfilled.

1.2.9 Test 9

```

bdn@bdn-VirtualBox: ~/Desktop
The names and codes of five star players of Bangladesh are:
-----
|   Code   |   Player Name   |
-----
|   PK   |   Paras Khadka   |
|   VK   |   Virat Kholi    |
|   DW   |   David Warner   |
|   BS   |   Ben Stokes     |
|   RT   |   Ross Taylor    |
-----
Choose Three Players: PK VK BS
Choose one of the players code:
1) PK
2) VK
3) BS
#? 

```

FIG: 9

Objective	The program displays a menu with these three codes and asks to choose one of them
Result	Objective Fulfilled

1.2.10 Test 10

```

bdn@bdn-VirtualBox: ~/Desktop
The names and codes of five star players of Bangladesh are:
-----
|   Code       |   Player Name   |
-----
|   PK         |   Paras Khadka  |
|   VK         |   Virat Kohli   |
|   DW         |   David Warner  |
|   BS         |   Ben Stokes    |
|   RT         |   Ross Taylor   |
-----
Choose Three Players: PK VK BS
Choose one of the players code:
1) PK
2) VK
3) BS
#? DW
-----
| Please enter 1 or 2 or 3. |
-----
#? 5
-----
| Please enter 1 or 2 or 3. |
-----
#? 

```

FIG: 10

Objective	The user types the number of a chosen code in the list; the user will be able to type numbers until it is one of these three codes
Result	Objective fulfilled

1.2.11 Test 11

```

TerminalB File~/EditkView Search Terminal Help
The names and codes of five star players of Bangladesh are:
-----
|   Code   |   Player Name   |
-----
|   PK   |   Paras Khadka   |
|   VK   |   Virat Kholi    |
|   DW   |   David Warner   |
|   BS   |   Ben Stokes     |
|   RT   |   Ross Taylor    |
-----
Choose Three Players: PK VK DW
Choose one of the players code:
1) PK
2) VK
3) DW
#? 3
Sorry, we are unable to open the specified file.
Do you want to execute program again?(enter yes or no) █

```

```

bdn@bdn-VirtualBox: ~/Desktop
The names and codes of five star players of Bangladesh are:
-----
|   Code   |   Player Name   |
-----
|   PK   |   Paras Khadka   |
|   VK   |   Virat Kholi    |
|   DW   |   David Warner   |
|   BS   |   Ben Stokes     |
|   RT   |   Ross Taylor    |
-----
Choose Three Players: PK VK BS
Choose one of the players code:
1) PK
2) VK
3) BS
#? DW
-----
| Please enter 1 or 2 or 3. |
-----
#? 5
-----
| Please enter 1 or 2 or 3. |
-----
#? 1
Paras Khadka is known as one of the best players Nepal has ever produced and is
current captain of the national side. He has also played at the Under-15 and Und
er-17 levels. A middle-order batsman who also bowls medium pace, he represented
Nepal in the Under-19 World Cups of 2004, 2006 and 2008 while also being a part
of the senior side. In the Plate semi-final against South Africa in 2006, Khadka
was responsible for taking his side through to the finals. He was also the high
est wicket-taker for Nepal in that tournament.

```

FIG: 11

Objective	The program displays the code with a comment on the same line and then, starting from the next line, displays a 3-4 lines description of that player. The description is taken from a file, which is
-----------	--

	stored in directory. If the file with that name is absent or not readable, the program informs the user about that and returns him/her to Step 4
Result	Objective Fulfilled.

1.2.12 Test 12

```

bdn@bdn-VirtualBox: ~/Desktop
|      PK      |      Paras Khadka      |
|      VK      |      Virat Kholi      |
|      DW      |      David Warner      |
|      BS      |      Ben Stokes      |
|      RT      |      Ross Taylor      |
|-----|
Choose Three Players: PK VK BS
Choose one of the players code:
1) PK
2) VK
3) BS
#? DW
|-----|
| Please enter 1 or 2 or 3. |
|-----|
#? 5
|-----|
| Please enter 1 or 2 or 3. |
|-----|
#? 1
Paras Khadka is known as one of the best players Nepal
current captain of the national side. He has also playe
er-17 levels. A middle-order batsman who also bowls med
Nepal in the Under-19 World Cups of 2004, 2006 and 2008
of the senior side. In the Plate semi-final against Sou
was responsible for taking his side through to the fin
est wicket-taker for Nepal in that tournament.
Do you want to execute program again?(enter yes or no)
|-----|

```

FIG: 12

Objective	The program asks whether the eight previous steps described above, (4) to (11), should be repeated
Result	Objective Fulfilled

1.2.13 Test 13

```

Terminal File Edit View Search Terminal Help
|  PK      |  Paras Khadka  |
|  VK      |  Virat Kohli   |
|  DW      |  David Warner  |
|  BS      |  Ben Stokes    |
|  RT      |  Ross Taylor   |
|-----|
Choose Three Players: PK VK BS
Choose one of the players code:
1) PK
2) VK
3) BS
#? DW
|-----|
| Please enter 1 or 2 or 3. |
|-----|
#? 5
|-----|
| Please enter 1 or 2 or 3. |
|-----|
#? 1
Paras Khadka is known as one of the best players Nepal has ever produced and is
current captain of the national side. He has also played at the Under-15 and Und
er-17 levels. A middle-order batsman who also bowls medium pace, he represented
Nepal in the Under-19 World Cups of 2004, 2006 and 2008 while also being a part
of the senior side. In the Plate semi-final against South Africa in 2006, Khadka
was responsible for taking his side through to the finals. He was also the high
est wicket-taker for Nepal in that tournament.
Do you want to execute program again?(enter yes or no) no
|-----|
| Thanks for using my program |
|-----|
bdn@bdn-VirtualBox:~/Desktop$

```

FIG: 13

Objective	If user says 'yes', they are repeated; if the user says 'no', the program terminates.
Result	Objective Fulfilled

1.2 Conclusion

The goal is to learn about the bash programming. Bash is not only an excellent command line shell but, a scripting language in itself. Shell scripting allows us to use the shell's abilities and to automate a lot of tasks that would otherwise require a lot of commands.

The script file is started with entering the Program name, user id and code on the opening screen followed by the security key which is not displayed for security. After the login is successful, the user is requested to make a guess from the list of the countries. After a successful guess, the user was again requested to make three choices from the players list. The user then can choose a player from the previous choices and the players details were obtained.

This coursework was interesting and the challenges that arose were fun to deal with. It wouldn't have been possible without the guidance from the module leader.

2. Task B

2.1 Introduction

As the tasks given in the course 2 part ii of the Network and Operating System, As the Network & Operating System Course is based on the Kali Linux Operating System, there is a necessary essay on memory management. So, memory management definition is the functionality of an operating system that manages primary memory and moves processes back and forth between main memory and disk during execution. It keeps track of each and every memory location, regardless of whether it can be assigned to a certain process. It can decide which process will get memory at certain time. It even tracks whenever some memory gets freed or unallocated & correspondingly it update the status. (tutorialspoint, 2020).

As memory management has different types of memory, it has the certain field called Process Address Space. It is define as the set of logical addresses that a process references in its code. Address Space has its three types of memory addresses. They are given below:

1. Symbolic addresses:

This addresses used in source code. The variable names, constants & instruction labels are basic elements of symbolic address.

2. Relative addresses:

At time of compilation, a compiler converts symbolic addresses in to relative addresses.

3. Physical addresses:

The loader generates these addresses at time when a program is loaded into main memory. It can be referred as logical address space.

Physical addresses space has its type of different memory. They are given below:

1) Static vs Dynamic Loading:

The choice between Static or Dynamic must be made at the time of the creation of the computer programme. When you need to load your software dynamically, then at the time of compilation, the full programme will be followed & connected without leaving any external programme. The linker combines the object programme with other required object modules into an absolute programme, which often includes logical addresses.

When you are writing a dynamically loaded programme, then your compiler will have to compile the programme & for all the modules that you chose to dynamically include, only the references will be given and the rest of the work will be done at the time of execution.

2) Static vs Dynamic Linking:

As explained above, by using a static connection the linker combines all the other modules required by the system into a single executable system to prevent any runtime errors and dependency.

3) The linker integrates all other modules available in a single program for the static connection to avoid errors and dependencies in the runtime. The linker uses this description.

4) Swapping:

Swapping is a mechanism in which a process can be swapped temporarily out of main memory to secondary storage & make that memory available to other memories. But some time later, system swaps back the process from secondary storage to main memory.

From this performance, it usually affected by swapping process but it helps in running multiple & big processes in parallel & that's the reason Swapping is also known as technique for memory compaction.

5) Memory Allocation:

From the memory allocation, main memory has its two partitions:

- Low memory: Operating systems resides in this memory.
- High memory: User processes are held in the high memory.

As operating system uses the following memory allocation:

- Single-partition allocation:
 - The relocation registry framework is used from this form of allocation to separate user processes from each other and to separate them from changing code and data on operational systems. The relocation register shall contain the smallest physical address value while the boundary register shall contain the number of logical addresses. Any logical address must be less than the register cap.
 - Multiple-partition allocation:
- 6) The main memory type given is divided in a number of fixed partitions, while only one process should be provided for each partition. If a partition is free, a process from the input queue is selected and the partition is loaded. The partition must be usable for another process when the process is done.
- 7) Fragmentation:
- The free memory space is divided into small bits as processes are loaded out and removed from memory. This also occurs that processes can not be allocated to their memory blocks due to their limited size and memory blocks. This is called fragmentation.
- It has two types:
- External fragmentation:

The total capacity of the memory is adequate to fulfil a request or to exist in a process in the network.
 - Internal fragmentation:

The compressed memory block is larger. A part of memory has not been used, because many processes cannot use it.
- 8) Paging:
- 9) More memory than physically mounted on the device can be used by the computer. This extra memory is called virtual memory and is a hard field that is

set up to simulate the RAM of the machine. Throughout the implementation of virtual memory, Paging technology plays an significant role.

10)Paging is a technique of memory management in which the space in a process address is divided into block(size is 2, between 512 and 8192 bytes) of the same size. A quantity of pages tests the size of the operation.

11)Segmentation:

Segmentation is a memory management technique in which each job is divided into multiple size modules, one per section containing similar functional parts. The program's various logical address space is given in each chapter.

Once a procedure is performed, the correct segmentation is loaded into a non-contiguous memory, but each segment must be put in a contiguous memory block.

Segmenting memory management operates much like paging, except here segments are variable in length because they are of fixed size, as in paging pages.

2.2Efficiency

Memory management is very effective in networking and operating system applications, because it includes memory processes in its user machine. We can find memory up to 32gigs during these days. The programmes are much larger with these sizes, & the computer system requires memory to operate optimally. The memory management function is for the existing processes to always have memory accessible. (Technobyte, 2018).

As well as the advantages of memory management are given below (Brainly, 2020):

- There is no fragmentation in internal memory.
- If segments are small & it should not be joined into one page it may be saved into its memory.
- It is less overhead.
- It is responsible for managing the computers primary memory.
- It keeps track of each as well as every memory location.

- The average segment is greater than average page size.
- From terms of segment table only one as possible entry per actual segment as opposed to per page in virtual memory.

2.3 Implementation & application on different areas

If users want to have private, infinitely big, unvolatile room. Non-volatile means the data can still be stored when the computer unintentionally is off. This is too expensive for us, however, if current technologies do not allow us.

The operating system has something called the memory manager. It's the job of the memory manager to efficiently manage memory which includes:

- Keeping track of which parts of memory is in use.
- Allocate memory to processes.
- Free up memory after processes have used it.
- In order to prevent the machine from getting stuck, it's the task of the OS and the memory manager to handle this memory between applications. When the possibility happens, it goes through an impasse, the OS knows how to tackle the impasse too. (Technobyte, 2018).

Memory management has its different uses on its fields are given below (Guru99, 2020):

1. Process management
2. File management
3. Device management
4. I/O System management
5. Secondary-Storage management
6. Security
7. Command interpretation
8. Networking
9. Job accounting

10. Communication management

2.4 Physical memory management in network & data storage

Disk Block is the system which maintains free disc block tracks when files are generated to allocate space. It also reuse the space freed from removing the files, so it is necessary to manage free space. A free space list that records disc blocks not allocated to a file or directorie is maintained by the programme. The disc block has the form of device memory management. It's: bitmap, list connected, grouping, counting, etc. (geeksforgeeks, 2020).

The physical address of the device RAM is the memory location and a series of numbers. The memory control unit (MMU) is the core hardware component of a computer system which translates the space for virtual logical addresses into physical addresses. The MMU is a physical hardware unit and is often referred to as a page memory management unit. (PMMU).

A virtual address is a translation of the mechanism through which the MMU translates a virtual address to a physical address, using the Page Directory Pointer (PDPT) to convert an address type to another address type.

The process is directly related to the allocation of a table list, matching and management of a specific address type. A Cache Function called the Lookaside Buffer Translation (TLB) is often used to accelerate Virtual Address Translation and forms part of the virtual address for the translation of physical addresses.

And its type of memory management are given below:

Static vs Dynamic Loading:

- The programmes & data can be charged to memory in various ways and the static and dynamic loading of two main methods in system memory.
- Static Loading :
The code is loaded into memory, before it is executed. Used in structured programming language including c.

- Dynamic Loading:

The code is stored in the memory of the operating system as needed. Found in object-oriented languages like Java.

The machine stores must be handled in addition to memory management. Memory, which is essential to the OS, is a central component of all modern computing systems. The memory is continuously recycled, removed, segmented, used and reused through continued virtualization & resource use optimisation. Memory management strategies can minimise memory management errors that can lead to instability and failures of the system & application.

2.5 Memory Management in operating system

With the use of the memory management system for managing the main memory and the secondary storage for the application and software. There are some of the reason that memory management is needed in the operating system are given below:

1. Relocation:
2. If a multi-programming framework is commonly used, many application processes run in the background. We can't know in advance which other programmes will be in the main memory and when our processes are going to be executed.
3. To overcome this, the memory manager is responsible for the planning, planning and assigning and releasing the memory accordingly, which makes the execution of processes effective and smooth.

4. Sharing:

When multiple processes run in the main memory, it is required to have a protection mechanism that must allow several processes to access the same portion of the main memory. By allowing each process the access to the same memory or the identical copy of a program rather than having a copy for each program has an advantage of efficient memory allocation. Memory management

should allow the controlled access for sharing memory without compromising the protection in the OS.

5. Protection:

When multiple processes are run, one process may be written in the address space of another process. That is why any other process applications should be protected against unwanted interference. In this situation, the memory manager protects the address space of each process. Keep the relocation algorithm in mind, too. The protection aspect and the transfer aspect of the memory manager work in the synchronisation and speed of the OS.

6. Physical Organization:

The memory structure consists of volatile main memory and secondary non-volatile memory. Applications that are stored in the secondary memory, which is the user computer's hard drive. But when you run the application, the application processes are transferred to the main memory, the RAM of the system. To maintain the flow of these transfers from main memory to secondary memory with ease, proper memory management is required.

7. Logical Organization:

Memory is a linear storage structure that consists of certain parts (data) that can be modified along with those that can not be. Memory management should allow the allocation, use and access of memory to user programmes in a systematic way that does not cause chaos by modifying a file that was not intended to be accessed by the user. It supports a basic module that provides the necessary protection and sharing of files that users need.

2.6 Virtual Memory concept with uses and development

Virtual memory is a memory management capability of an operating system (OS) that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory (RAM) to disk storage. Virtual address space has increased using active memory in RAM and inactive

memory in the hard disk drives (HDDs) to form contiguous addresses that hold both the application and its data processes.

Virtual memory was developed at a time when physical memory and installed RAM were expensive. Computers have a finite RAM limitation, so that memory can run out, especially when multiple programmes run at the same time. A system that uses virtual memory can use a section of the hard drive to emulate the RAM. With virtual memory, the system can load larger programmes or multiple programmes running at the same time, allowing each programme to operate as if it has infinite memory and without having to purchase more RAM.

Among the primary benefits of virtual memory is its ability to handle twice as many addresses as the main memory. You can use software to consume more memory by using the HDD as a temporary storage device for data processes while MMUs convert virtual memory addresses to physical addresses via the CPU. Programs use virtual addresses to store instructions and data; when a programme is running, virtual addresses are converted to actual memory addresses. Virtual memory also frees applications to manage shared memory and saves users files from main memory that need to add memory modules when RAM space is running out. (TechTarget, 2020).

For the uses of the virtual memory in the operating system, with the uses of the virtual memory in the operating system. The advantages of virtual memory are given below (Guru99, 2020):

- It helps to gain speed when only a particular segment of the program is required for the execution of the program.
- It is very helpful in implementing a multiprogramming environment.
- It allows you to run more applications at once.
- Code should be read from disk whenever required.
- The code can be placed anywhere in physical memory without requiring relocation.
- It helps you to fit many large programs into smaller programs.

- More processes should be maintained in the main memory, which increases the effective use of CPU.
- Each page is stored on a disk until it is required after that, it will be removed.
- It allows more applications to be run at the same time.
- There is no specific limit on the degree of multiprogramming.
- Large programs should be written, as virtual address space available is more compared to physical memory.
- It helps you to fit many large programs into smaller programs.

2.7 Conclusion

From the above description of the memory management that is used on the operating system. By learning how to manage data process memory in a versatile way and managing data resources. With the use of memory management, the operating system and networking have divided the amount of data processes in the main memory and the performance of the operating system. With the memory management system, there is a small amount of programme that slows down the operating system. It can be used in a variety of fields such as job accounting, security, file management, etc. From the above description of the memory management that is used by the operating system. With the use of memory management, the operating system and networking have divided the number of data processes in the main memory and the performance of the operating system. There is a certain amount of small programme that slows down the operating system with the memory management system.

2.8References

TechTarget, G. S., 2020. *TechTarget*. [Online]

Available at: <https://searchstorage.techtarget.com/definition/virtual-memory>

[Accessed 29 March 2020].

Brainly, g .A ., 2020. *brainly. in*. [Online]

Available at :<https://brainly.in/question/6315728>

[Accessed 29 March 2020].

https://www.tutorialspoint.com/operating_system/os_memory_management.htm

[Accessed 28 March 2020].

Guru99, K. R., 2020. *guru99*. [Online]

Available at: <https://www.guru99.com/operating-system-tutorial.html>

[Accessed 29 March 2020].

Guru99, K. R., 2020. *Guru99*. [Online]

Available at: <https://www.guru99.com/virtual-memory-in-operating-system.html>

[Accessed 29 March 2020].

geeksforgeeks, S. J., 2020. *geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/free-space-management-in-operating-system/>

[Accessed 29 March 2020].

Technobyte, T. P., 2018. *technobyte*. [Online]

Available at: <https://www.technobyte.org/memory-management-os-simple-explanation/>

[Accessed 29 March 2020]