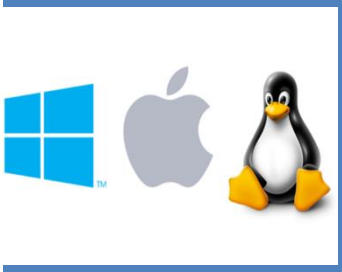


207SE OPERATING SYSTEMS, SECURITY AND NETWORKS SUBMISSION

Portfolio 1

Student ID: 8251628

Course: Computer Hardware and Software Engineering

Contents Page

Lab Activity 1 – Operating Systems Tasks and Programming

a. Comparison between the Harvard Architecture and Von Neumann Architecture

Embedded system architecture relates to how input and output devices, memory buses, controllers are integrated into operating systems and applications as seen in **Figure 1** below. Most electronic devices do have forms of embedded processors. The main types of system architecture are the Harvard architecture and Von Neumann architecture.

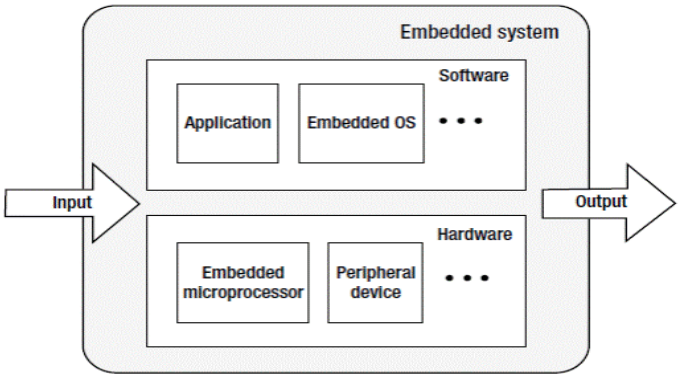


Figure 1: Typical Architecture of An Embedded System (n.d.)

The Von Neumann architecture also referred to as the Princeton architecture (Vega et al. 2017), is easier to understand when compared to the Harvard architecture model which is seen in figure 3.0. This model uses a singular path to access memory that holds both data and instruction sets. This model is used in personal computers and workstations (Olivka 2020). The advantage of this model is that the implementation and development are simplified and faster as the control unit derives data and instruction from the same memory (Von Neumann Architecture | History & Use | Computer Science (n.d.) On the other hand, instructions can only be carried out sequentially as parallel implementation is not feasible.

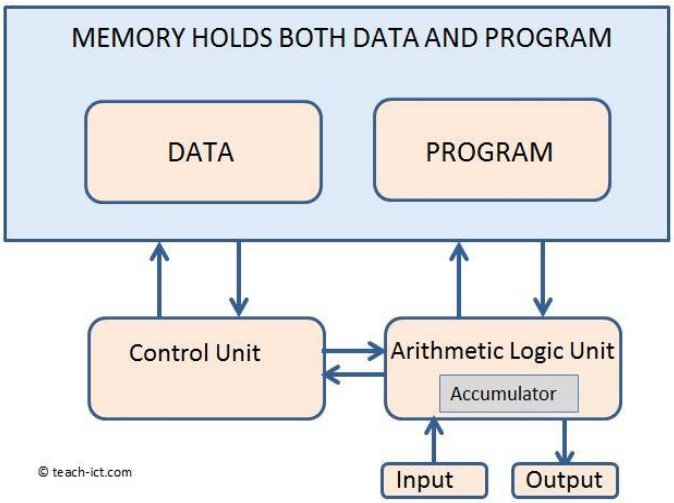


Figure 2: Teach ICT- Neumann Layout (n.d.)

Furthermore, the Harvard architectural setup seen in Figure 3, has separate data and instruction set memory blocks. There are also independent buses that access these two areas of memory. Harvard Architecture these days are mainly used in microprocessors and signal processing. The benefit of the Harvard model is that computers utilizing this model can run and programs and access data independently and simultaneously, this model is more complicated, but it solves the Von Neumann bottleneck problem which arises from accessing data and instruction set. The problem with this model arises from its complexity and cost.

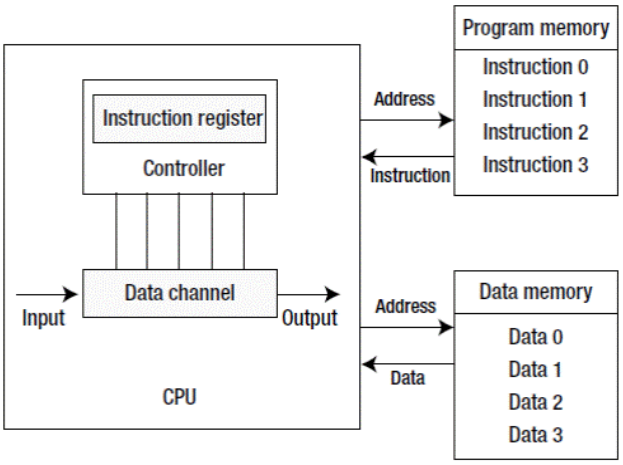


Figure 3: Harvard Layout (n.d.)

Lastly, the Harvard layout is a modern approach in terms of computer architecture thus enabling development to be more flexible in comparison with the Von Neumann approach. Thus, the layout also aids prevents program corruption which the Von Neumann approach was liable

b. Programming activity

```
#include <iostream>
#include <string>
#include <sstream>
#include <vector>

using namespace std;



    int main()
    {
        string userInput;
        while (true)
        {
            cout << "Enter the Baxter Bot instructions: " << endl;
            std::getline(std::cin, userInput);

            if (cin.fail()) // has a previous extraction failed?
            {
               cin.clear(); // put us back in 'normal' operation mode
               cin.ignore(32767, '\n'); // and remove the bad input
               cout << "Whoa bad data, try again\n";
            }
            else if (cin.good())
            {
                stringstream currentstring(userInput); //Initalise Input
String//
                int count = -1;
                string instruction[10];
                //Repeatedly put instruction in string array
                while (currentstring.good())
                {
                    count = count + 1;
                    currentstring >> instruction[count];
                }

                for (int i = 0; i < instruction->size(); i++) //This loop
enables me use the break statement.
                {
                    //Object Action Time//
                    if ((instruction[0] == "orange" || instruction[0] ==
"apple" || instruction[0] == "car" || instruction[0] == "bus" ||
instruction[0] == "diamond")
                        &&
                        (instruction[1] == "recognize" || instruction[1] ==
"eat" || instruction[1] == "see" || instruction[1] == "lift" ||
instruction[1] == "drop" || instruction[1] == "fetch")
                        &&
                        (instruction[2] == "1second" || instruction[2] ==
"2seconds" || instruction[2] == "5seconds" || instruction[2] ==
"unlimited"))
                    {
                        cout << "Instruction is okay" << endl; // Output
can be understood by Baxter robot//
                        break;
```

```
                    }

                    //Object size action//
                    else if ((instruction[0] == "orange" || instruction[0]
== "apple" || instruction[0] == "car" || instruction[0] == "bus" ||
instruction[0] == "diamond")
                            &&
                            (instruction[1] == "small" || instruction[1] ==
"big" || instruction[1] == "little" || instruction[1] == "massive")
                            &&
                            (instruction[2] == "recognize" || instruction[2] ==
"eat" || instruction[2] == "see" || instruction[2] == "lift" ||
instruction[2] == "drop" || instruction[2] == "fetch"))
                    {
                        cout << "Instruction is okay " << endl;  // Output
can be understood by Baxter robot//
                        break;
                    }


                    //Move time//
                    else  if ((instruction[0] == "left" || instruction[0]
== "right" || instruction[0] == "forwards" || instruction[0] == "backwards"
|| instruction[0] == "stop")
                            &&
                            (instruction[1] == "1second" || instruction[1] ==
"2seconds" || instruction[1] == "5seconds" || instruction[1] ==
"unlimited"))
                    {
                        cout << "Instruction is okay" << endl; // Output
can be understood by Baxter robot//
                        break;
                    }

                    //Move time Move Time//
                    else if ((instruction[0] == "left" || instruction[0] ==
"right" || instruction[0] == "forwards" || instruction[0] == "backwards"
|| instruction[0] == "stop")
                            &&
                            (instruction[1] == "1second" || instruction[1] ==
"2seconds" || instruction[1] == "5seconds" || instruction[1] ==
"unlimited")
                            &&
                            (instruction[2] == "left" || instruction[2] ==
"right" || instruction[2] == "forwards" || instruction[2] == "backwards"
|| instruction[2] == "stop")
                            &&
                            (instruction[3] == "1second" || instruction[3] ==
"2seconds" || instruction[3] == "5seconds" || instruction[3] ==
"unlimited"))
                    {
                        cout << "Instruction is okay" << endl; // Output
can be understood by Baxter robot//
                        break;
                    }
```

```
                    //Location <Action> <Object>//
                    else if ((instruction[0] == "door" || instruction[0] ==
"kitchen" || instruction[0] == "table")
                            &&
                            (instruction[1] == "recognize" || instruction[1] ==
"eat"  ||  instruction[1]  ==  "see"  ||  instruction[1]  ==  "lift"  ||
instruction[1] == "drop" || instruction[1] == "fetch")
                            &&
                            (instruction[2] == "orange" || instruction[2] ==
"apple"  ||  instruction[2]  ==  "car"  ||  instruction[2]  ==  "bus"  ||
instruction[2] == "diamond"))
                    {
                        cout << "Instruction is okay" << endl; // Output
can be understood by Baxter robot//
                        break;

                    }
                    //None of the conditions are True//
                    else
                    {
                        cout << "Instruction is Incorrect " << endl; //
Output cannot be understood by Baxter robot//
                        break;
                    }
                }

                cout << "Would you like to run again?  Y or N:" << endl;
                cin >> userInput;
                if (userInput == "yes" || userInput == "Y" || userInput ==
"YES" || userInput == "y")
                {
                   cin.clear(); // put us back in 'normal' operation mode
                   cin.ignore(32767, '\n'); // and remove the invalid input
                }
                else
                {
                    cout << "Exiting....." << endl;
                    break;
                }
            }
        }
        return 0;
    }
```

Outcome from code

```
akangd@hvs-its-lnx01:~$ g++ -o parser Lab_one.cc
akangd@hvs-its-lnx01:~$ ./parser
Enter the Baxter Bot instructions:
aaple
Instruction is Incorrect
Would you like to run again?  Y or N:
Y
Enter the Baxter Bot instructions:
apple small eat
Instruction is okay
Would you like to run again?  Y or N:
N
Exiting.....
akangd@hvs-its-lnx01:~$
```

Figure 1.0 Code Output

Proof of Compilation Video:

Advance Task: Adv1

Lab Activity 2 – Linux Command Line (Commands and outcomes from a series of small tasks that require use of a number of Linux commands)

Tasks – Files

a. Create a directory in your area of the os-207SE server or your installation of Linux. The directory with a name made up of you second name followed by 207SE and the year (mine would be ELSHAW207SE2020). Make the directory read/write/executable only for you, read/write for your groups and read only for others.

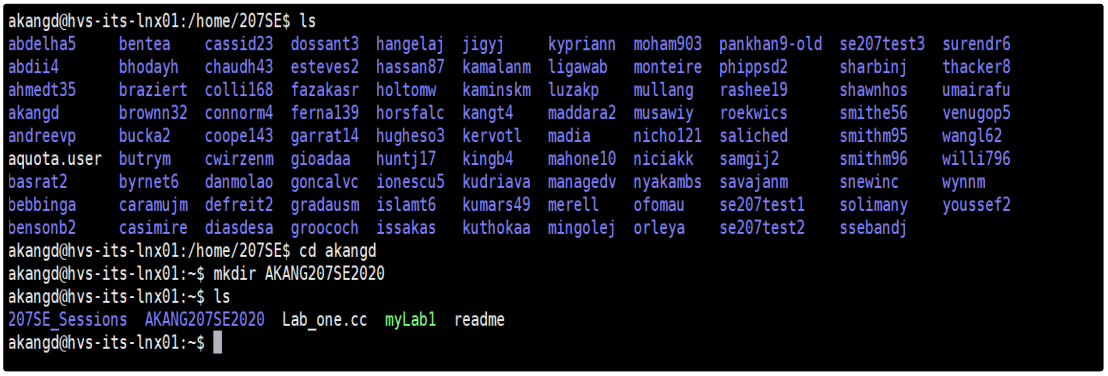


Figure 2.0 Creating a Directory

b. Show evidence of this using the appropriate version of the **ls command**.

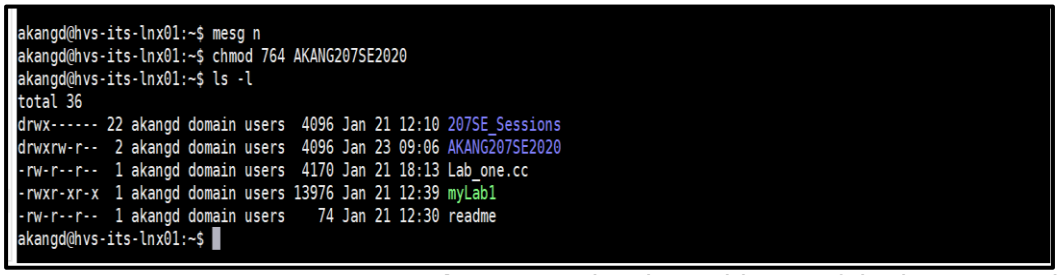


Figure 2.1 Showing evidence of the ls command

c. Download the script http://www.centerkey.com/tree/tree.sh to your home directory using wget and make the file executable.

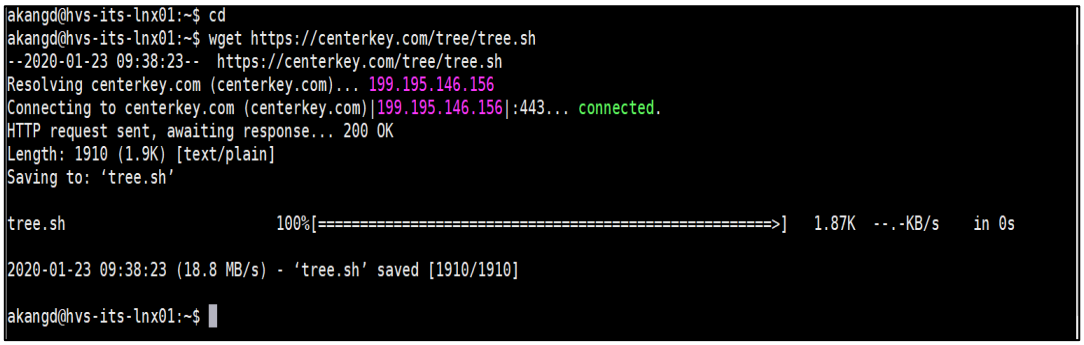


Figure 2.2 Downloading Script using wget

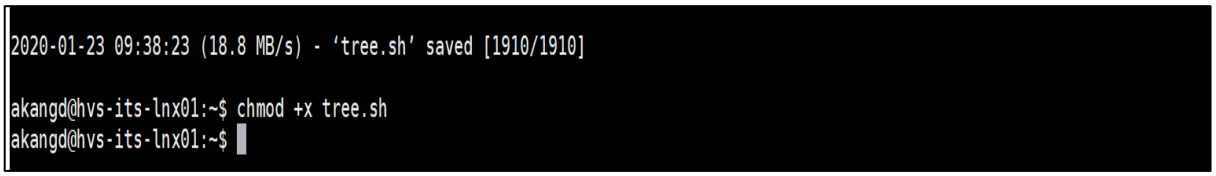


Figure 2.3 Making file executable

The **wget** command word is derived from **World Wide Web** and **get**, it is used to retrieve and download content from web servers, and it helps when the user wants to download files

d. Create a directory called **wrongDirectory**. You release it is not what you wanted so delete it.

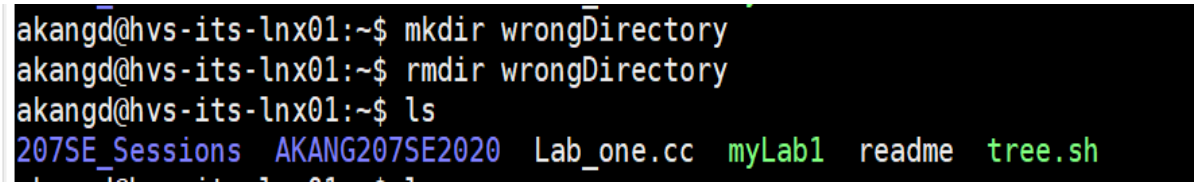


Figure 2.4 Creating and Deleting Directory

e. Create Portfolio1-2020 and Portfolio2-2020 directories in the directory you created in part a.

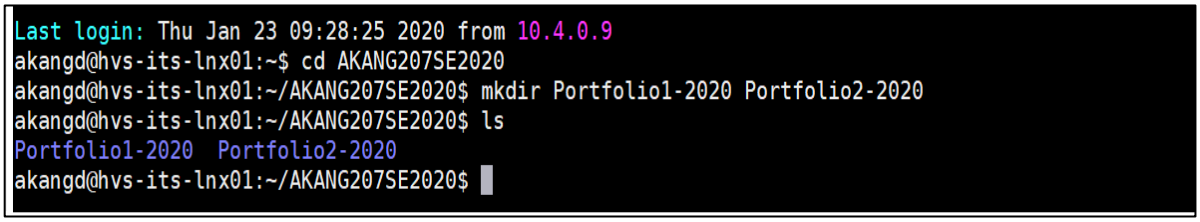


Figure 2.5 Creating Directories

f. Create numbered directories in the Portfolio1-2020 Directory (Lab0-207SE to Lab10-207SE) and in the Porfolio2-2020 Directory (Lab11-207SE to Lab20-207SE).

Create numbered directories for Portfolio1-2020 Directory

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ mkdir -v Lab{0..10}-207SE
mkdir: created directory 'Lab0-207SE'
mkdir: created directory 'Lab1-207SE'
mkdir: created directory 'Lab2-207SE'
mkdir: created directory 'Lab3-207SE'
mkdir: created directory 'Lab4-207SE'
mkdir: created directory 'Lab5-207SE'
mkdir: created directory 'Lab6-207SE'
mkdir: created directory 'Lab7-207SE'
mkdir: created directory 'Lab8-207SE'
mkdir: created directory 'Lab9-207SE'
mkdir: created directory 'Lab10-207SE'
```

Figure 2.6: Numbered Directories output for Portfolio1-2020 Directory

Create numbered directories and output for Portfolio2-2020 Directory

```
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cd Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio2-2020$ mkdir -v Lab{11..20}-207SE
mkdir: created directory 'Lab11-207SE'
mkdir: created directory 'Lab12-207SE'
mkdir: created directory 'Lab13-207SE'
mkdir: created directory 'Lab14-207SE'
mkdir: created directory 'Lab15-207SE'
mkdir: created directory 'Lab16-207SE'
mkdir: created directory 'Lab17-207SE'
mkdir: created directory 'Lab18-207SE'
mkdir: created directory 'Lab19-207SE'
mkdir: created directory 'Lab20-207SE'
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio2-2020$ ls
Lab11-207SE  Lab13-207SE  Lab15-207SE  Lab17-207SE  Lab19-207SE
Lab12-207SE  Lab14-207SE  Lab16-207SE  Lab18-207SE  Lab20-207SE
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio2-2020$
```

Figure 2.7: Numbered Directories output for Portfolio2-2020 Directory

g. In <YourSecondName>207SE2020 directory create a text file called **LastTask.txt** and then using the appropriate Linux command copy this document into Directory Lab0-207SE.

Creating the Text File

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio2-2020$ cd ..
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cat > LastTask.txt
ls
cd..
exit
exit

^C
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$
```

Figure 2.8 Creating text file

Copying the document into Directory Lab0-207SE.

```
akangd@hvs-its-lnx01:~$ cd AKANG207SE2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cp LastTask.txt Portfolio1-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cd Portfolio1-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ ls
Lab0-207SE   Lab1-207SE  Lab3-207SE  Lab5-207SE  Lab7-207SE  Lab9-207SE
Lab10-207SE  Lab2-207SE  Lab4-207SE  Lab6-207SE  Lab8-207SE  LastTask.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ cp LastTask.txt Lab0-207SE
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ cd Lab0-207SE
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab0-207SE$ ls
LastTask.txt
```

Figure 2.9 Copying Document into directory

Evidence of directory structure using tree.sh

```
|--AKANG207SE2020
|----Portfolio1-2020
|------Lab0-207SE
|------Lab10-207SE
|------Lab1-207SE
|------Lab2-207SE
|------Lab3-207SE
|------Lab4-207SE
|------Lab5-207SE
|------Lab6-207SE
|------Lab7-207SE
|------Lab8-207SE
|------Lab9-207SE
|----Portfolio2-2020
|------Lab11-207SE
|------Lab12-207SE
|------Lab13-207SE
|------Lab14-207SE
|------Lab15-207SE
|------Lab16-207SE
|------Lab17-207SE
|------Lab18-207SE
|------Lab19-207SE
|------Lab20-207SE
```

Figure 3.0 Evidence of Directory Structure

Linux Commands – Mixed

a. Using the date command show todays date and the time and date 5 years ago. Using the cal command show the month that you were born. Change this calendar to make Monday the first day of the week.

Command Showing Todays Date and time

```
akangd@hvs-its-lnx01:~$
akangd@hvs-its-lnx01:~$ ls
207SE_Sessions  AKANG207SE2020  Lab_one.cc  myLab1  readme  tree.sh
akangd@hvs-its-lnx01:~$ cd AKANG207SE2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ date
Thu 23 Jan 11:22:06 GMT 2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$
```

Figure 4.0 Command Showing Todays Date and time

Command Showing Date 5 years ago

```
akangd@hvs-its-lnx01:~/AKANG207SE2020$ date --date='5 year ago'
Fri 23 Jan 11:30:00 GMT 2015
akangd@hvs-its-lnx01:~/AKANG207SE2020$
```

Figure 4.1: Command showing date 5 years ago

Command Showing month born

```
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cal 07 1999
     July 1999
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

Figure 4.2: Command showing month born

Command making Monday the first day of the week

```
207SE_Sessions  AKANG207SE2020  Lab_one.cc  myLab1  readme  tree.sh
akangd@hvs-its-lnx01:~$ ncal -M
    January 2020
Mo     6 13 20 27
Tu     7 14 21 28
We  1  8 15 22 29
Th  2  9 16 23 30
Fr  3 10 17 24 31
Sa  4 11 18 25
Su  5 12 19 26
akangd@hvs-its-lnx01:~$
```

Figure 4.3: Command making Monday the first day of the week

b. Move into the lab1-207SE directory and use the appropriate command to show the current directory.

Command used PWD- Print Working Directory

```
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cd Portfolio1-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ ls
Lab0-207SE   Lab1-207SE  Lab3-207SE  Lab5-207SE  Lab7-207SE  Lab9-207SE
Lab10-207SE  Lab2-207SE  Lab4-207SE  Lab6-207SE  Lab8-207SE  LastTask.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ cd Lab1-207SE
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab1-207SE$ ls
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab1-207SE$ pwd
/home/207SE/akangd/AKANG207SE2020/Portfolio1-2020/Lab1-207SE
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab1-207SE$
```

Figure 4.4: Command showing current directory

c. Display the time of a user (ab0487) last login.

```
akangd@hvs-its-lnx01:~$ cd
akangd@hvs-its-lnx01:~$ cd AKANG207SE2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ last ab0487
ab0487   pts/18       10.0.76.58       Thu Jan 23 10:32 - 10:51  (00:18)

wtmp begins Thu Jan  2 04:10:01 2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$
```

Figure 4.5: Command displaying a user's last login time

User's Home Directory and Full Name

```
akangd@hvs-its-lnx01:~/AKANG207SE2020$ finger ab0487
Login: ab0487                           Name: Mark Elshaw
Directory: /home/STAFF/ab0487           Shell: /bin/bash
Last login Thu Jan 23 10:32 (GMT) on pts/18 from 10.0.76.58
No mail.
No Plan.
akangd@hvs-its-lnx01:~/AKANG207SE2020$
```

Figure 4.6: Command displaying a user's home directory and full name

d. Find out how to prevent the effects of talk, write and wall from interrupting you. What command can you use?

Preventing the effects of talk, write and wall

```
akangd@hvs-its-lnx01:~$ cd
akangd@hvs-its-lnx01:~$ pwd
/home/207SE/akangd
akangd@hvs-its-lnx01:~$ mesg n
akangd@hvs-its-lnx01:~$
```

Figure 4.7: Preventing the effects of talk, wall and write

e. Show the command to verify that www.coventry.ac.uk exists and can accept requests.

Finding out if a website exists and can accept requests

```
akangd@hvs-its-lnx01:~/AKANG207SE2020$ if wget -q --method=HEAD https://www.coventry.ac.uk;
>  then
>    echo "This page exists and can accept requests."
>  else
>    echo "This page does not exist and cannot accept requests."
> fi
This page exists and can accept requests.
akangd@hvs-its-lnx01:~/AKANG207SE2020$
```

Figure 4.8: Finding out if a website exists and can accept requests

f. Display your name and favourite programming language on the screen using the echo command.

Display your name and favourite programming language using Echo

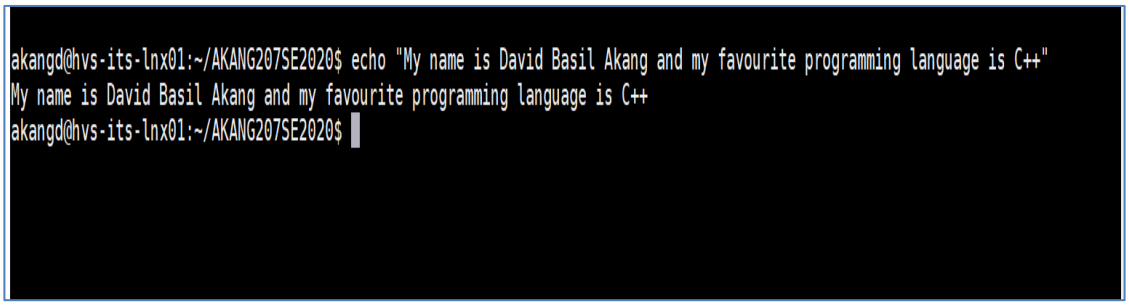


Figure 4.9: Displaying name and favourite programming language

g. Find out how you can display your username on the screen and at least two ways to display who is logged on.

Command showing username and two Commands showing who are logged on

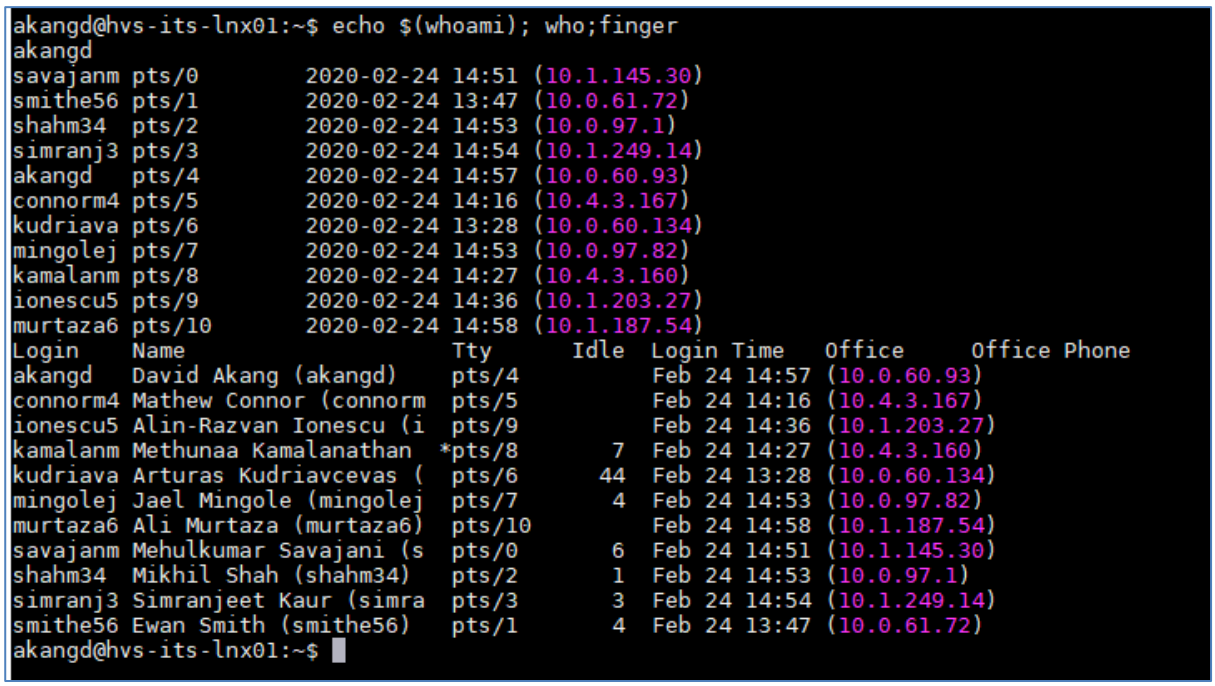


Figure 5.0: Displaying username and two users who are logged on

h. Use two ways to list the processes that are running.

Command showing processes currently running

Top Command

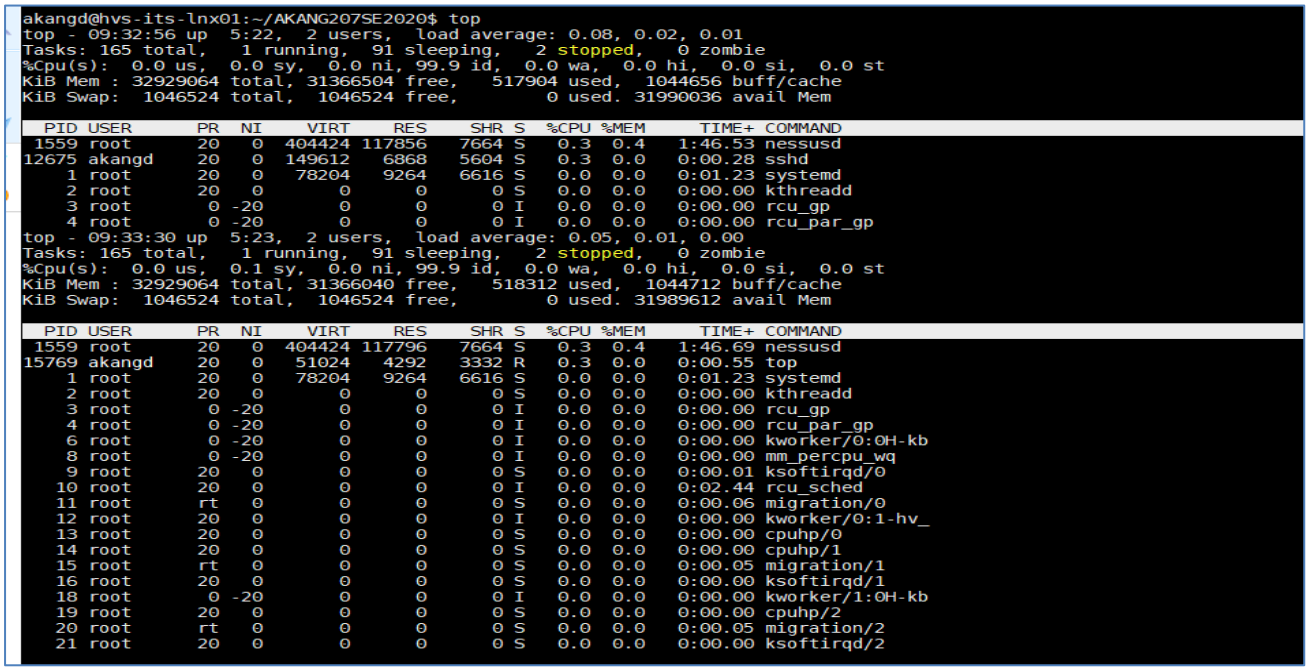


Figure 5.1: Showing processes using the top command

Ps Aux Command

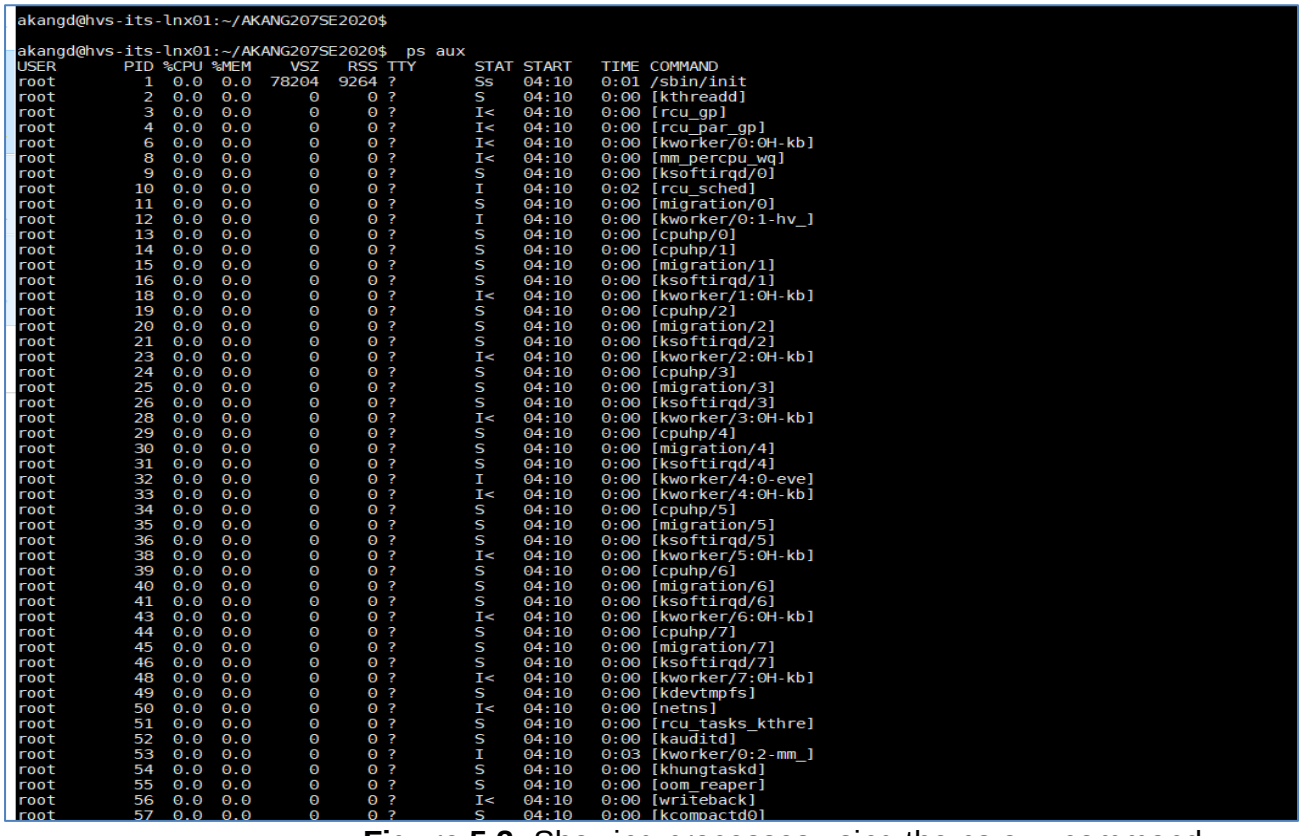


Figure 5.2: Showing processes using the ps aux command

i. What are the differences between the Linux commands copy (cp), rename and move?

Difference between cp, rename and move

The cp command is used for coping files and directories. The 'cp' term stands for copy. An example of the cp command is seen in Figure 5.3 below where LastTask.txt is copied from one directory to another

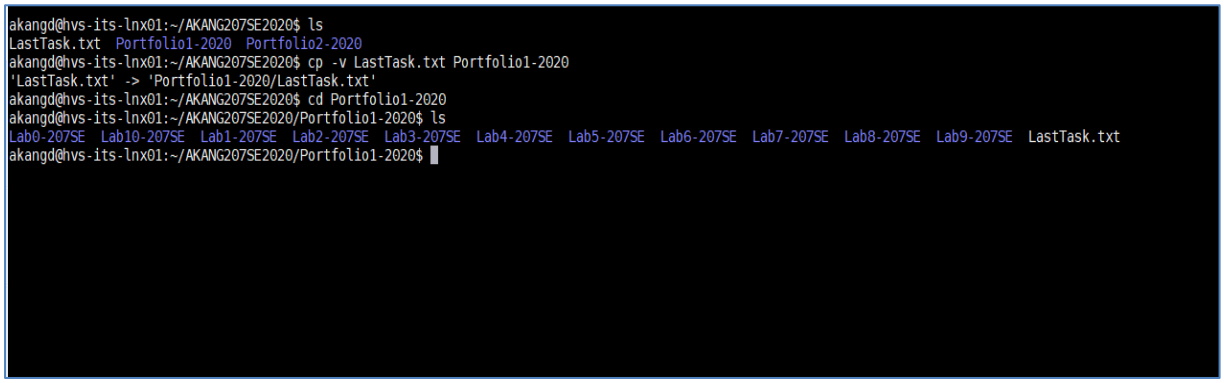


Figure 5.3: Copy Command

The rename command is used for changing the file name for files, the command can either be used to change the file name for files or it can also change the extension as seen in Figure 5.4 and Figure 5.5

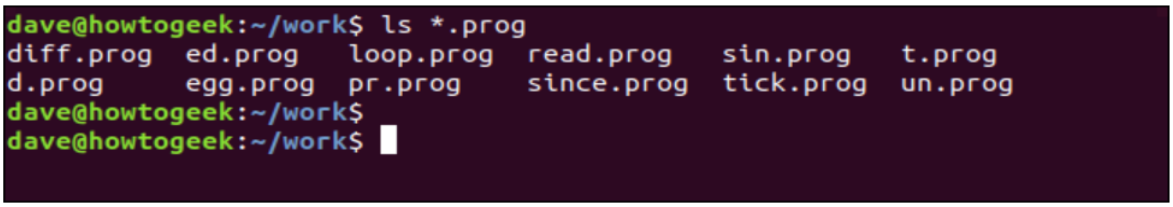


Figure 5.4: Command Listing. Prog Files (Mckay 2019)

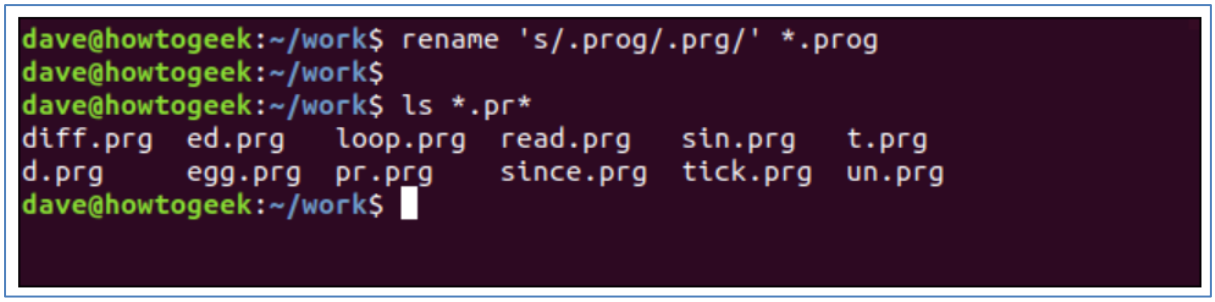


Figure 5.5: Rename Command Changing Extension (Mckay 2019)

The move command is used for moving directories or files from one location to another. It can also be used to move either single files or directories or numerous ones from one location to another as seen in Figure 5.6 below:

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ cd
akangd@hvs-its-lnx01:~$ ls
207SE_Sessions  AKANG207SE2020  Lab_one.cc  myLab1  readme  tree.sh
akangd@hvs-its-lnx01:~$ cd AKANG207SE2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cat > myExample.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  myExample.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ mv myExample.txt Portfolio1-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ ls
LastTask.txt  Portfolio1-2020  Portfolio2-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020$ cd Portfolio1-2020
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$ ls
Lab0-207SE   Lab1-207SE  Lab3-207SE  Lab5-207SE  Lab7-207SE  Lab9-207SE
Lab10-207SE  Lab2-207SE  Lab4-207SE  Lab6-207SE  Lab8-207SE  myExample.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020$
```

Figure 5.6: Move Command

j. With a single command, how would you get systems information such as processes, memory, paging and CPU activity?

Command Showing System Information using vmstat

```
akangd@hvs-its-lnx01:~$ vmstat 1 5
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 30626580 340784 1324868    0    0     4     3   13   78  0  0 99  0  0
 0  0      0 30626068 340784 1324868    0    0     0     0   24  495  0  0 100  0  0
 0  0      0 30626076 340784 1324868    0    0     0     0   25  464  0  0 100  0  0
 0  0      0 30626328 340784 1324868    0    0     0     0   27  468  0  0 100  0  0
 0  0      0 30626580 340784 1324868    0    0     0     4   55  540  0  0 100  0  0
akangd@hvs-its-lnx01:~$
```

Figure 5.7: Vmstat Command

Tasks – Document Manipulation

a. Use **cat** to show the contents of the file.

Using cat to show the contents of the file.

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ cat poem.txt
Where is the clear lad?
Rise quietly like a big pirate.
Why does the shore endure?
Adventure, life, and desolation.
Why does the wind grow?
Oh, faith!
The shark dies like a rainy breeze.
Waves sail!
Desolation, love, and faith.
Love is a stormy wind.
The ship sails like a dead sun.
Never view a ship.
She is a simple woman.
Brought up the old fashion way.
```

Figure 5.8: Cat Command

b. Use an appropriate command to display the CRC checksum and byte count of the file.

Command used cksum

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ cksum poem.txt
264938488 367 poem.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$
```

Figure 5.9: Check sum Command

c. Use **grep** to show only lines not containing the words "she" or "he". Lines contain both “she” and “he” should be shown.

Command used grep

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ grep -v 'she\|he' poem.txt
Rise quietly like a big pirate.
Adventure, life, and desolation.
Oh, faith!
Waves sail!
Desolation, love, and faith.
Love is a stormy wind.
Never view a ship.
```

Figure 6.0: grep Command

d. Use **grep** to show the 5 lines above a line containing the text ‘the’.

Command showing lines containing ‘the’

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ grep -B5 'the' poem.txt
Where is the clear lad?
Rise quietly like a big pirate.
Why does the shore endure?
Adventure, life, and desolation.
Why does the wind grow?
--
Desolation, love, and faith.
Love is a stormy wind.
The ship sails like a dead sun.
Never view a ship.
she is a simple woman.
Brought up the old fashion way.
```

Figure 6.1: Grep command showing lines with the text ‘the’

e. Using Linux commands you should count the lines containing “she” and “he” but not both and display the line numbers that “she” and “he” but not both appear on in the original document.

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ grep -n 'she\|he' poem.txt
1:Where is the clear lad?
3:Why does the shore endure?
5:Why does the wind grow?
7:The shark dies like a rainy breeze.
11:The ship sails like a dead sun.
13:she is a simple woman.
14:Brought up the old fashion way.
```

Figure 6.2: Grep command showing lines with the text ‘he or she’

f. Find a command to list the top 3 lines of the **poem.txt** file and then the bottom line of these top 3.

Top 3 Lines and bottom 3 lines

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ head -3 poem.txt ; tail -3 poem.txt
Where is the clear lad?
Rise quietly like a big pirate.
Why does the shore endure?
Never view a ship.
she is a simple woman.
Brought up the old fashion way.akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$
```

Figure 6.3: command showing top 3 and bottom lines

g. Find a command to split the **poem.txt** file into different files each containing 2 lines.

Splitting poem.txt file into different files each containing 2 lines

```
Brought up the old fashion way.akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ split -l2 poem.txt NewPoemFile
1.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ ls
NewPoemFile1.txtaa  NewPoemFile1.txtac  NewPoemFile1.txtae  NewPoemFile1.txtag
NewPoemFile1.txtab  NewPoemFile1.txtad  NewPoemFile1.txtaf  poem.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ cat NewPoemFile1.txtaa
Where is the clear lad?
Rise quietly like a big pirate.
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$
```

Figure 6.4: command text file into different files

h. Use **sort** and **rev** to reverse the sorted contents of poem.txt and append the output to poem2.txt.

1. Sorting Contents of File

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ sort -R poem.txt > poem2.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ ls
NewPoemFile1.txtaa  NewPoemFile1.txtac  NewPoemFile1.txtae  NewPoemFile1.txtag  poem.txt
NewPoemFile1.txtab  NewPoemFile1.txtad  NewPoemFile1.txtaf  poem2.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ cat poem2.txt
Love is a stormy wind.
Oh, faith!
Never view a ship.
Rise quietly like a big pirate.
Waves sail!
Why does the shore endure?
she is a simple woman.
The ship sails like a dead sun.
Where is the clear lad?
Adventure, life, and desolation.
Desolation, love, and faith.
Brought up the old fashion way.
Why does the wind grow?
The shark dies like a rainy breeze.
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$
```

Figure 6.5: command sorting contents of file.

2. Reversing the Contents of File

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ rev poem2.txt
.dniw ymrots a si evoL
!htiaf ,hO
.pihs a weiv reveN
.etarip gib a ekil ylteiuq esiR
!lias sevaW
?erudne erohs eht seod yhW
.namow elpmis a si ehs
.nus daed a ekil slias pihs ehT
?dal raelc eht si erehW
.noitalosed dna ,efil ,erutnevdA
.htiaf dna ,evol ,noitaloseD
.yaw noihsaf dlo eht pu thguorB
?worg dniw eht seod yhW
.ezeerb yniar a ekil seid krahs ehT
```

Figure 6.6: command reversing contents of file.

i. Use at least two appropriate Linux commands to compare these two files (poem.txt and poem2.txt) and see if they are the same.

Comparing Files

1. Comm Command

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ comm -12 poem.txt poem2.txt
comm: file 2 is not in sorted order
comm: file 1 is not in sorted order
Why does the shore endure?
Why does the wind grow?
The shark dies like a rainy breeze.
```

Figure 6.7: Comm Command

➔ This command compares sorted files line by line

2. Diff Command

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ diff -c poem.txt poem2.txt
*** poem.txt    2020-01-24 14:25:16.912364040 +0000
--- poem2.txt   2020-01-24 15:01:34.923507882 +0000
***************
*** 1,14 ****
! Where is the clear lad?
  Rise quietly like a big pirate.
  Why does the shore endure?
  Adventure, life, and desolation.
  Why does the wind grow?
- Oh, faith!
  The shark dies like a rainy breeze.
- Waves sail!
- Desolation, love, and faith.
- Love is a stormy wind.
- The ship sails like a dead sun.
- Never view a ship.
- she is a simple woman.
- Brought up the old fashion way.
\ No newline at end of file
--- 1,14 ----
! Love is a stormy wind.
! Oh, faith!
! Never view a ship.
  Rise quietly like a big pirate.
+ Waves sail!
  Why does the shore endure?
+ she is a simple woman.
+ The ship sails like a dead sun.
+ Where is the clear lad?
  Adventure, life, and desolation.
+ Desolation, love, and faith.
+ Brought up the old fashion way.
  Why does the wind grow?
  The shark dies like a rainy breeze.
```

Figure 6.8: Diff Command

This command compares two files and prints the lines and the numbers which are different.

j. Use **sort** to sort the content of poem.txt file in a random order and redirect the output to a new file called **poem2.txt**.

Sorting and Redirecting Output

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ sort -R poem.txt > poem2.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ cat poem2.txt
Where is the clear lad?
Why does the shore endure?
Rise quietly like a big pirate.
Adventure, life, and desolation.
Never view a ship.
Desolation, love, and faith.
The shark dies like a rainy breeze.
Oh, faith!
Waves sail!
she is a simple woman.
The ship sails like a dead sun.
Why does the wind grow?
Love is a stormy wind.
Brought up the old fashion way.
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$
```

Figure 6.9: Sorting and redirecting output

k. Sort the **poem.txt** file, remove the duplicates and reverse the sorted contents and append the output to **poem2.txt**.

Sorting, removing duplicates, reversing and outputting to text

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ sort -r poem.txt | uniq > poem2.txt
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ cat poem2.txt
Why does the wind grow?
Why does the shore endure?
Where is the clear lad?
Waves sail!
The ship sails like a dead sun.
The shark dies like a rainy breeze.
she is a simple woman.
Rise quietly like a big pirate.
Oh, faith!
Never view a ship.
Love is a stormy wind.
Desolation, love, and faith.
Brought up the old fashion way.
Adventure, life, and desolation.
```

Figure 7.0: Removing, reversing and appending the output of text file

l. Create an **alias** so rather than having to type the full command for k) you can type **yourSort**.

Creating an Alias

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ alias mySort="sort -r -u poem.txt > poem2.txt"
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab2-207SE$ mySort
```

Figure 7.1: Creating an Alias

Lab Activity 4 Bootloader

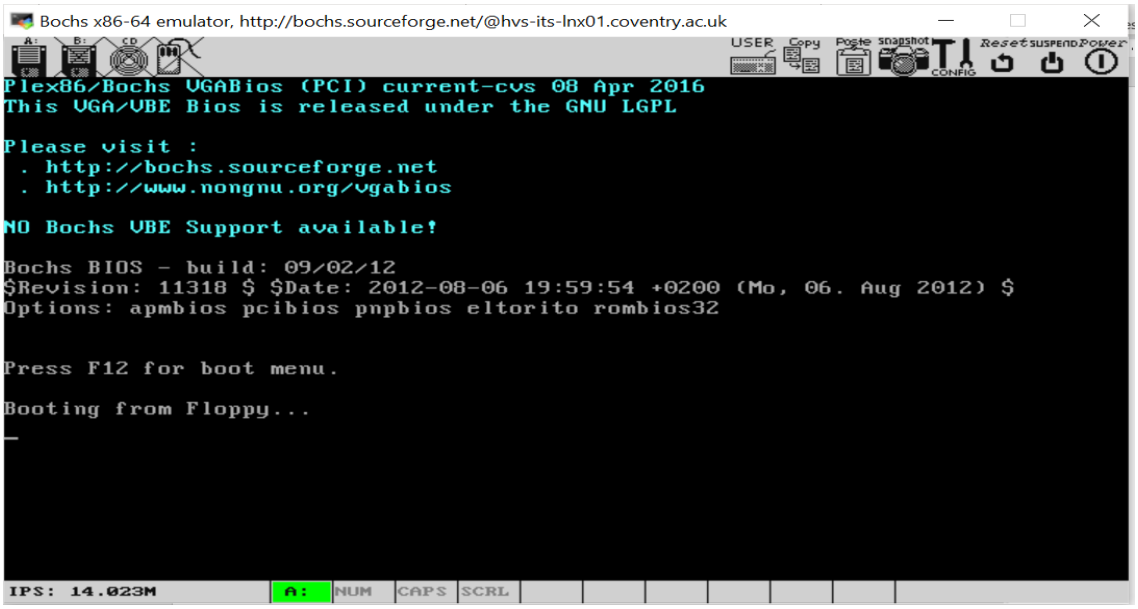
Basic Tasks

a. Brief description of the Lab activity and what you did

The basic activity for this lab is to create a boot loader, a boot loader is a program or set of instructions that is ran which is responsible for starting boot time tasks and processes of an operating system (Techopedia.com, n.d.). For this activity, I displayed my name and course, and thereafter I moved on to the next activity which comprised of displaying a diamond of dots. Thereafter the boot loader is loaded by running bochs using pragma-linux.

Boot pragma Linux with bochs

```
akangd@hvs-its-lnx01:~/207SE_Sessions$ cd Session4
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4$ cd pragmalinux-img/
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ nasm bootloader.asm
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ dd if=bootloader bs=512 of=a.img
1+0 records in
1+0 records out
512 bytes copied, 0.000247203 s, 2.1 MB/s
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ bochs
========================================================================
                       Bochs x86 Emulator 2.6
            Built from SVN snapshot on September 2nd, 2012
========================================================================
00000000000i[     ] LTDL_LIBRARY_PATH not set. using compile time default '/usr/lib/bochs/plugins'
00000000000i[     ] BXSHARE not set. using compile time default '/usr/share/bochs'
00000000000i[     ] lt_dlhandle is 0x5644119b5af0
00000000000i[PLGIN] loaded plugin libbx_unmapped.so
00000000000i[     ] lt_dlhandle is 0x5644119b6890
00000000000i[PLGIN] loaded plugin libbx_biosdev.so
00000000000i[     ] lt_dlhandle is 0x5644119b81a0
00000000000i[PLGIN] loaded plugin libbx_speaker.so
00000000000i[     ] lt_dlhandle is 0x5644119b89f0
```



b. Make a bootloader that displays your name, course, and your favourite operating system

Commented bootloader code to display name, course, and your favourite operating system

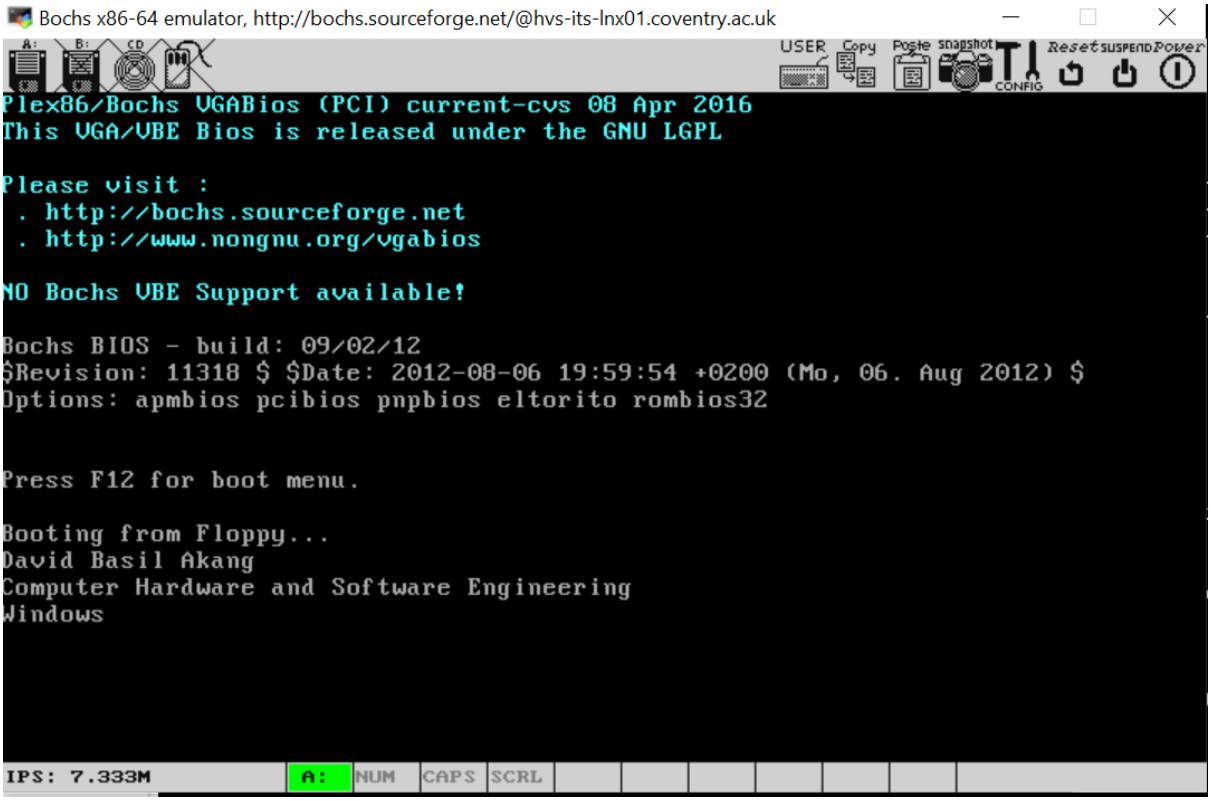
Output from bochs showing name, course, and your favourite operating system

```
[BITS 16]
[ORG 0x7C00]
top:
  ;; Put 0 into ds (data segment)
  ;; Can't do it directly
  mov ax,0x0000
  mov ds,ax
  ;; si is the location relative to the data segment of the
  ;; string/char to display
  mov si, HelloWorld
  call writeString ; function  call to write string
  mov si, courseOutput ; moving the memory address caled courseoutput to si
  call writeString ; function call to write string
  mov si, osOutput ; moving the memory address caled osOutput to si
  call writeString; function call to write string
  jmp $ ; Spin
writeString:
  mov ah,0x0E ; Display a chacter (as before)
  mov bh,0x00
  mov bl,0x07
nextchar:
  Lodsb ; Loads [SI] into AL and increases SI by one
  ;; Effectively "pumps" the string through AL s
  cmp al,0 ; End of the string?
  jz done
  int 0x10 ; BIOS interrupt
  jmp nextchar
done:
  ret
  HelloWorld db 'David Basil Akang',13,10,0 ; Here my name is returned, ASCII character 10 is for line feed and 13 is for carriage return used for
 advancing to the beginning of the next line
  courseOutput db 'Computer Hardware and Software Engineering',13,10,0; Here my course is being returned
  osOutput db 'Windows',13,10,0; Here my operating system is being returned
  times 510-($-$$) db 0
  dw 0xAA55
```

Proof of Compilation

```
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ nasm bootloader.asm
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ dd if=bootloader bs=512 of=a.img
1+0 records in
1+0 records out
512 bytes copied, 0.000228303 s, 2.2 MB/s
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ bochs
========================================================================
                       Bochs x86 Emulator 2.6
            Built from SVN snapshot on September 2nd, 2012
```

Bochs Output



c. Make a bootloader that displays a diamond of dots without using loops as well as the student information.

Commented Bootloader Code:

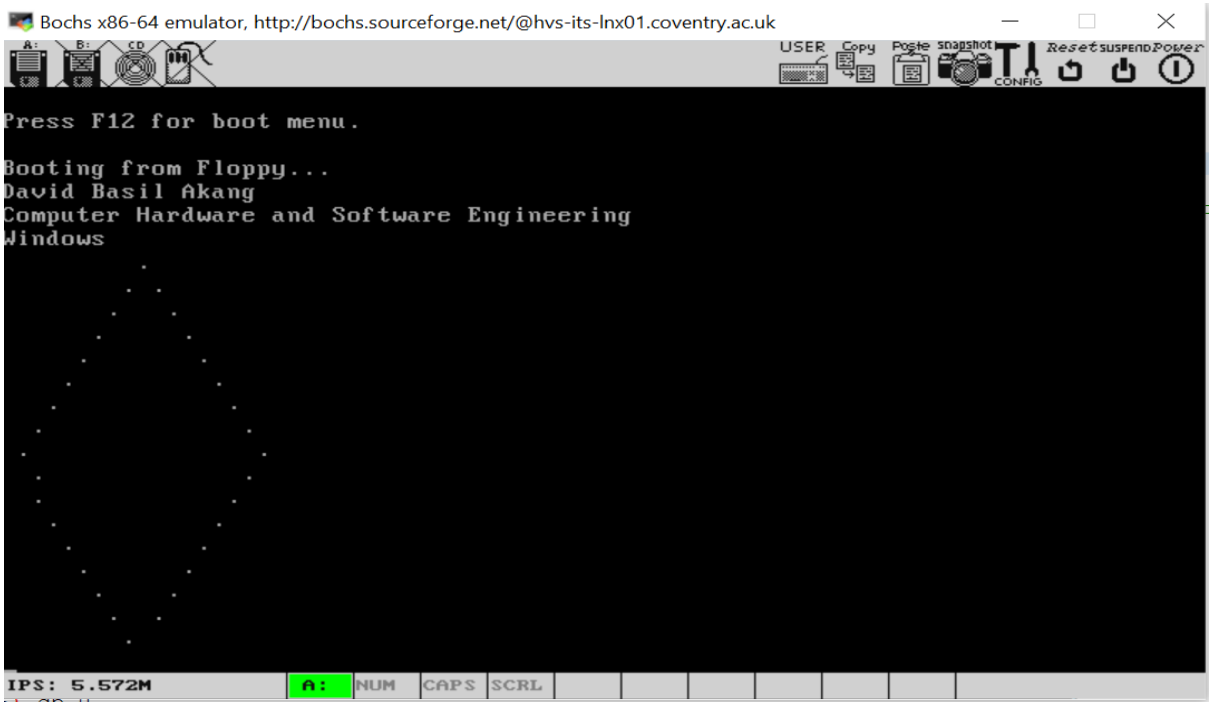
```
[BITS 16]
[ORG 0x7C00]
top:
  ;; Put 0 into ds (data segment)
  ;; Can't do it directly
  mov ax,0x0000
  mov ds,ax
  ;; si is the location relative to the data segment of the
  ;; string/char to display
  mov si, HelloWorld
  call writeString ; function  call to write string
  mov si, courseOutput ; moving the memory address called courseoutput to si
  call writeString ; function call to write string
  mov si, osOutput ; moving the memory address called osOutput to si
  call writeString; function call to write string

  mov si, diamondOutput; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput1; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput2; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput3; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput4; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput5; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput6; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput7; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput8; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput9; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput10; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string

  mov si, diamondOutput11; moving the memory address/Variable diamondOutput to si
   call writeString; function call to write string
  mov si, diamondOutput12; moving the memory address/Variable diamondOutput to si
     call writeString; function call to write string
  mov si, diamondOutput13; moving the memory address/Variable diamondOutput to si
     call writeString; function call to write string
  mov si, diamondOutput14; moving the memory address/Variable diamondOutput to si
     call writeString; function call to write string
  mov si, diamondOutput15; moving the memory address/Variable diamondOutput to si
     call writeString; function call to write string
  mov si, diamondOutput16; moving the memory address/Variable diamondOutput to si
  call writeString; function call to write string
  jmp $ ; Spin
writeString:
  mov ah,0x0E ; Display a chacter (as before)
  mov bh,0x00
  mov bl,0x07
nextchar:
  Lodsb ; Loads [SI] into AL and increases SI by one
  ;; Effectively "pumps" the string through AL s
  cmp al,0 ; End of the string?
  jz done
  int 0x10 ; BIOS interrupt
  jmp nextchar
done:
  ret
  HelloWorld db 'David Basil Akang',13,10,0 ; Here my name is returned, ASCII character 10 is for line feed and 13 is for carriage return used for
 advancing to the beginning of the next line
  courseOutput db 'Computer Hardware and Software Engineering',13,10,0; Here my course is being returned
```

```
66   courseOutput db 'Computer Hardware and Software Engineering',13,10,0; Here my course is being returned
67   osOutput db 'Windows',13,10,0; Here my operating system is being returned
68 diamondOutput db  '         .', 13,10,0; Here is the dot to be returned
69 diamondOutput1 db '        . .', 13,10,0; Here is the dot to be returned
70 diamondOutput2 db '       .   .', 13,10,0; Here is the dot to be returned
71 diamondOutput3 db '      .     .', 13,10,0; Here is the dot to be returned
72 diamondOutput4 db '     .       .', 13,10,0; Here is the dot to be returned
73 diamondOutput5 db '    .         .', 13,10,0; Here is the dot to be returned
74 diamondOutput6 db '   .           .', 13,10,0; Here is the dot to be returned
75 diamondOutput7 db '  .             .', 13,10,0; Here is the dot to be returned
76 diamondOutput8 db ' .               .', 13,10,0; Here is the dot to be returned
77 diamondOutput9 db '  .             . ', 13,10,0; Here is the dot to be returned
78 diamondOutput10 db'  .            .  ', 13,10,0; Here is the dot to be returned
79 diamondOutput11 db'   .          .   ', 13,10,0; Here is the dot to be returned
80 diamondOutput12 db'    .        .', 13,10,0; Here is the dot to be returned
81 diamondOutput13 db'     .      .', 13,10,0; Here is the dot to be returned
82 diamondOutput14 db'      .    .', 13,10,0; Here is the dot to be returned
83 diamondOutput15 db'       .  .', 13,10,0; Here is the dot to be returned
84 diamondOutput16 db'        .  ', 13,10,0; Here is the dot to be returned
85
86   times 510-($-$$) db 0
```

Proof of Compilation

```
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ nasm bootloader.asm
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ dd if=bootloader bs=512 of=a.img
1+0 records in
1+0 records out
512 bytes copied, 0.000228303 s, 2.2 MB/s
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ bochs
========================================================================
                       Bochs x86 Emulator 2.6
            Built from SVN snapshot on September 2nd, 2012
```

Bochs Output



Advanced Task (a)

Make a bootloader that displays a diamond of dots using loops as well as displaying the student details.

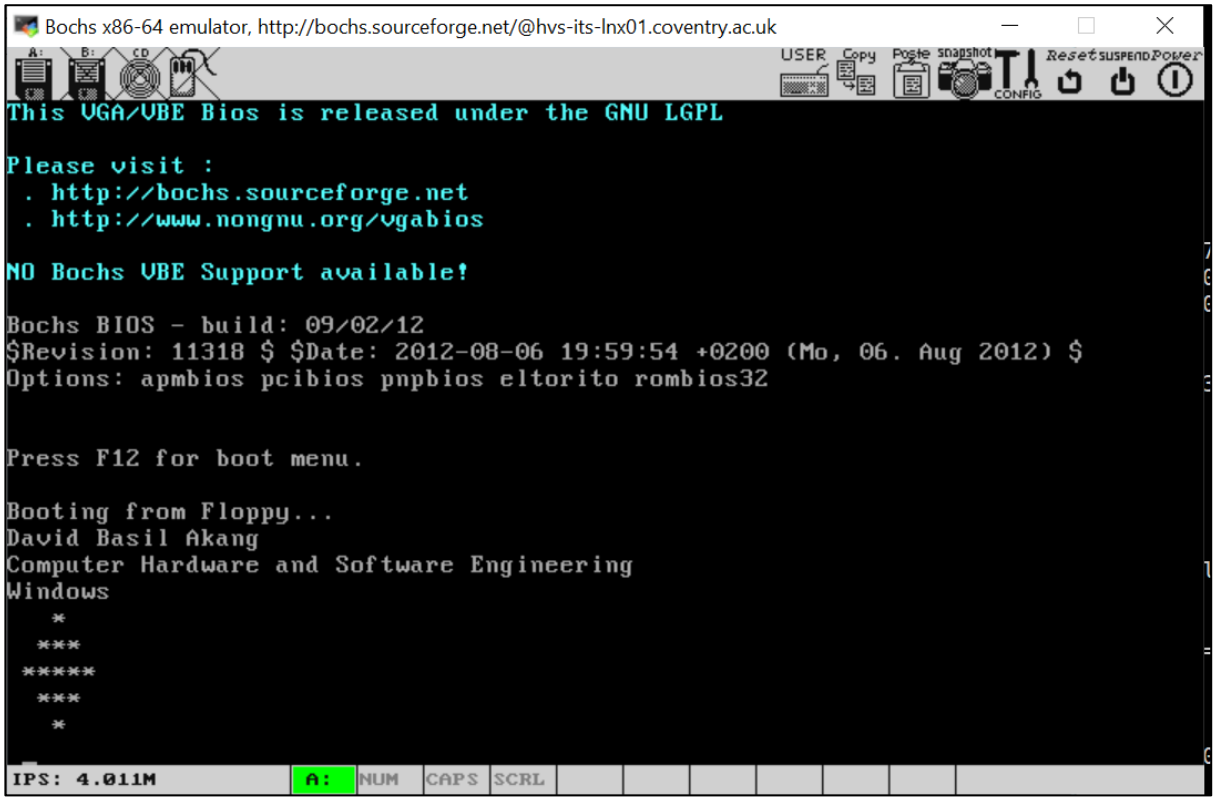
```
[BITS 16]
[ORG 0x7C00]
top:
        mov ax,0x0000   ;; Put 0 into ds (data segment)
        mov ds,ax       ;; Can't do it directly

  mov si, HelloWorld
        call writeString ; function  call to write string
  mov si, courseOutput ; moving the memory address called courseoutput to
si
  call writeString ; function call to write string
  mov si, osOutput ; moving the memory address called osOutput to si
  call writeString; function call to write string

mov cx, 1  ; pass the value of 1 to the cx register
loop_1:

  mov dx, 7  ;pass the value of 13 to the dx register

    loop_2:
      mov si, space    ; moving space into si
      call writeString ; output the space
      sub dx, 3; decrements the value of dx register
      cmp dx,cx ; compares value of dx with cx
      jge loop_2 ;loop back to loop2
      mov dx,cx     ;
        loop_3:
          mov si, diamondOutput    ;moving dots in data segment
          call writeString  ; print diamond
          dec dx         ; decrement the value inside the ax register
          cmp dx,0         ; compares the dx  value with 0
          jne loop_3      ; jump back if values not equal
mov si,newline         ; moving  newline to si
call writeString    ; print new line
add cx, 2            ; increment cx
cmp cx,7      ;  compare cx with 13
jne loop_1  ; jump back to outer loop if values not equal


              ;; Other part of Diamond
mov cx, 3     ; move 3 into cx register
loop_4:
  mov dx, 6  ; inner loop
   loop_5:
    mov si, space    ; moving space into si
    call writeString    ; printing space
    sub dx, 2             ; decrements the loop
    cmp dx,cx        ; comparison of inner loop with outer loop
    jge loop_5  ; jumps back if greater than or equal to
    mov dx,cx  ; 2nd inner loop with value of outerloop
      loop_6:
```

```
        mov si, diamondOutput  ; moves dot in data segment
        call writeString ; prints data segment
        dec dx      ; decrements the loop
        cmp dx,0    ; compares the value of dx  with zero
        jne loop_6  ; jumps back to loop top if not equal
        mov si,newline    ; moves line in data segment
        call writeString    ; prints out new line
        sub cx, 2         ; decrements the loop
        cmp cx,0  ; end of outerloop
        jge loop_4  ;jumps back if greater or equal


writeString:
        mov ah,0x0E ; Display a chacter (as before)
        mov bh,0x00
        mov bl,0x07
nextchar:
        Lodsb ; Loads [SI] into AL and increases SI by one
        ;; Effectively "pumps" the string through AL s
        cmp al,0 ; End of the string?
        jz done
        int 0x10 ; BIOS interrupt
        jmp nextchar
done:
        ret
        HelloWorld db 'David Basil Akang',13,10,0 ; Here my name is
returned, ASCII character 10 is for line feed and 13 is for carriage return
used for advancing to the beginning of the next line
  courseOutput db 'Computer Hardware and Software Engineering',13,10,0;
Here my course is being returned
  osOutput db 'Windows',13,10,0; Here my operating system is being returned
diamondOutput db '*',0; store the '.' in the memory location called
diamondoutput
newline db '', 13,10,0 ; Newline stored
space db ' ',0; Print out a space
returnCarriage db '',13,0; Return Carriage
times 510-($-$$) db 0
dw 0xAA55
```

Proof of Compilation:

```
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ nasm bootloader.asm
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ dd if=bootloader bs=512 of=a
.img
1+0 records in
1+0 records out
512 bytes copied, 0.000148102 s, 3.5 MB/s
akangd@hvs-its-lnx01:~/207SE_Sessions/Session4/pragmalinux-img$ bochs
```

Code output:


Bochs x86-64 emulator, http://bochs.sourceforge.net/@hvs-its-lnx01.coventry.ac.uk
USER
Copy
Paste
Snapshot
CONFIG
Reset
SUSPEND
Power
This VGA/VBE Bios is released under the GNU LGPL
Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios
NO Bochs VBE Support available!
Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32
Press F12 for boot menu.
Booting from Floppy...
David Basil Akang
Computer Hardware and Software Engineering
Windows
*
***
*****
***
*
IPS: 4.011M
A:
NUM
CAPS
SCRL

Lab Activity 5 Exploring what is going on outside the processor

a. List the information found in the /proc directory about the computer CPUs.

Command used: cat/proc/cpuinfo

```
akangd@hvs-its-lnx01:/proc$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 85
model name      : Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz
stepping        : 4
microcode       : 0xffffffff
cpu MHz         : 2294.606
cache size      : 25344 KB
physical id     : 0
siblings        : 8
core id         : 0
cpu cores       : 8
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 21
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb
onstant_tsc rep_good nopl xtopology cpuid pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor
3dnowprefetch invpcid_single pti ssbd ibrs ibpb stibp fsgsbase bmi1 hle avx2 smep bmi2 erms invpcid rtm mpx avx512f avx512dq rdseed adx smap
avx512cd avx512bw avx512vl xsaveopt xsavec xsaves flush_l1d
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs taa itlb_multihit
bogomips        : 4589.21
clflush size    : 64
cache_alignment : 64
address sizes   : 44 bits physical, 48 bits virtual
```

b. Provide a list of the device drivers configured into the currently running kernel. Count the number of different device drivers that are included in the kernel.

Command and Output

```
akangd@hvs-its-lnx01:/proc$ wc -l /proc/devices
57 /proc/devices
```

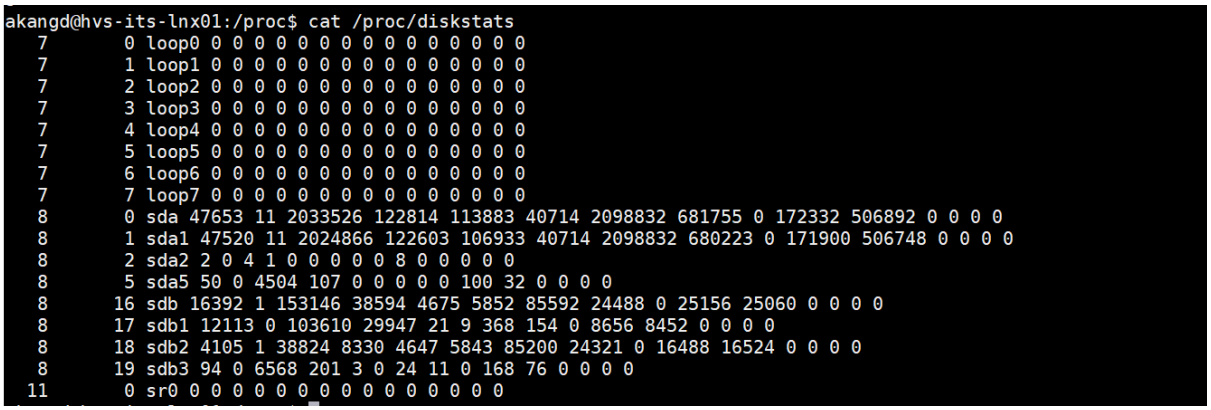
c. Show the number of CPUs, the producer of the CPUs and the CPU model.

Command showing vendor ID, physical ID and model name

```
akangd@hvs-its-lnx01:~$ lscpu | egrep 'Model name|Socket|Thread|NUMA|CPU\(s\)'
CPU(s):              8
On-line CPU(s) list: 0-7
Thread(s) per core:  1
Socket(s):           1
NUMA node(s):        1
Model name:          Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz
NUMA node0 CPU(s):   0-7
akangd@hvs-its-lnx01:~$ lscpu | egrep 'Model name|vendor_id|CPU\(s\)'
CPU(s):              8
On-line CPU(s) list: 0-7
Model name:          Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz
NUMA node0 CPU(s):   0-7
```

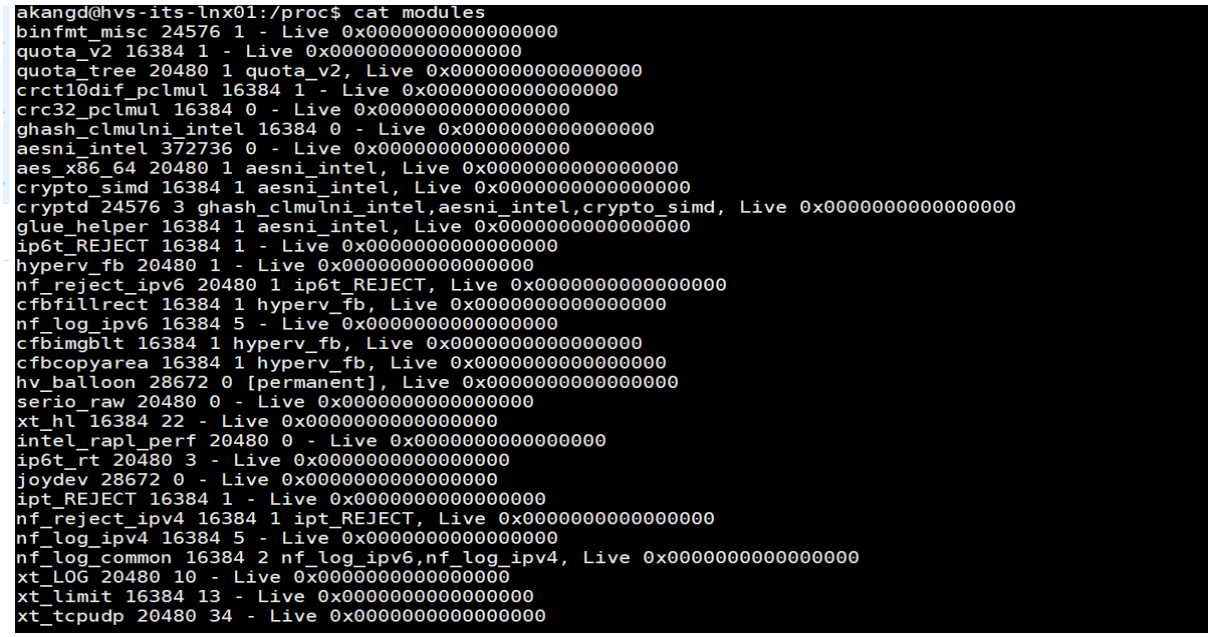
d. Using the /proc/diskstats show the names of the output devices and the number of megabytes read per second during the sampled interval.

Cat Command showing columns



Cat command showing device name and megabytes read per second

e. Display a list of all modules that have been loaded by the system.



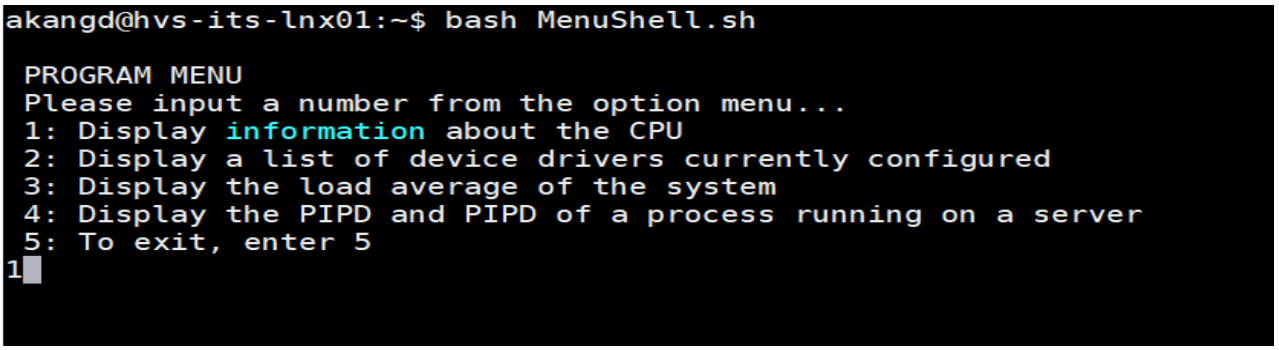
Advanced Task

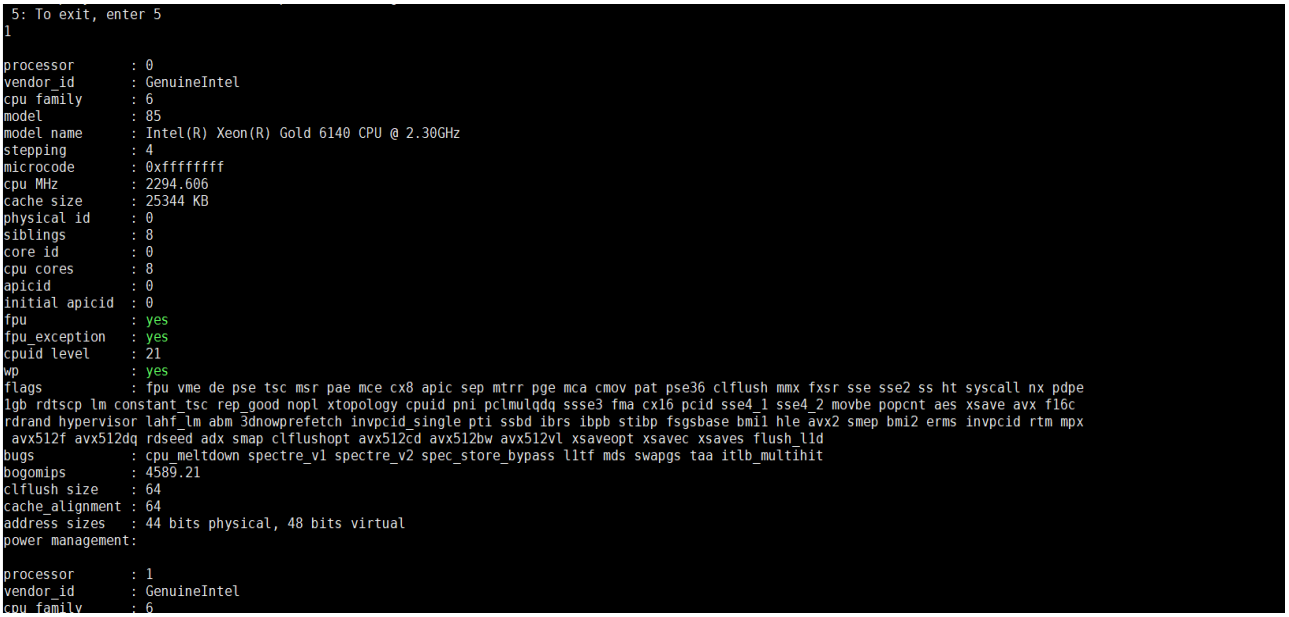
a. Using while and case statements develop a menu-based shell script that gains information from the proc directory to allow the user to select options

```
#!/bin/sh

while true;  # This loop runs so that the user may enter another value as
requested or close the program
do
echo "" #New line
echo " PROGRAM MENU" #List of user options been outputted
echo  " Please input a number from the option menu..."
echo  " 1: Display information about the CPU "
echo  " 2: Display a list of device drivers currently configured "
echo  " 3: Display the load average of the system "
echo  " 4: Display the PIPD and PIPD of a process running on a server"
echo  " 5: To exit, enter 5" #If user enters the value, the code will exit

read character # This command reads the user input
echo "" # New line
1
case $character in  #Case statement which picks an option based on user
input
  1 ) cat /proc/cpuinfo ;; #command called if user inputs 1
  2 ) cat /proc/devices ;;#command called if user inputs 2
  3 ) cat /proc/loadavg ;;#command called if user inputs 3
  4 ) awk '{print $4, $5}' /proc/stat ;; #command called if user inputs 4
  5 ) echo "Exiting now....." #command called if user inputs 5
      exit ;;
  * ) echo "Oops invalid input, try again" #command called if user input
is invalid

  esac


done # End of program
```

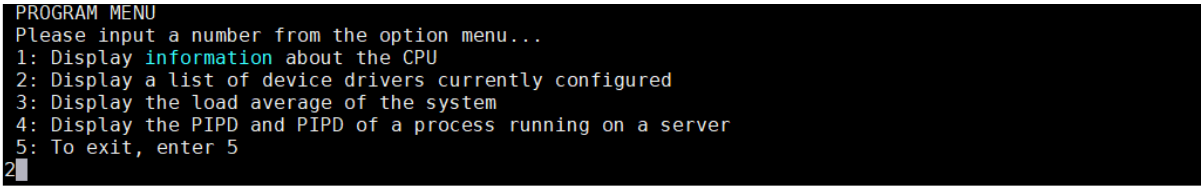
Code Output

Screenshot Showing User Inputting 1:

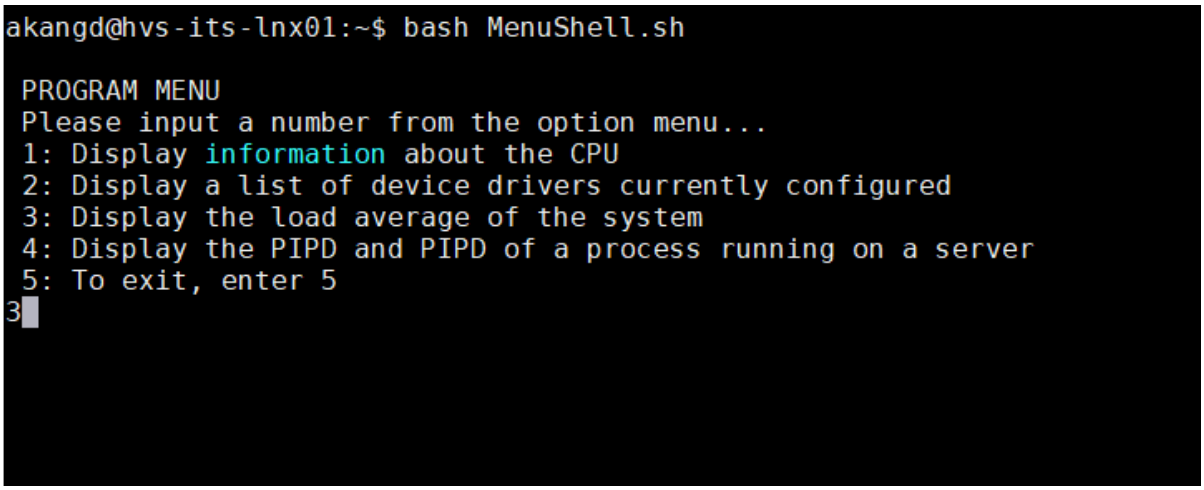


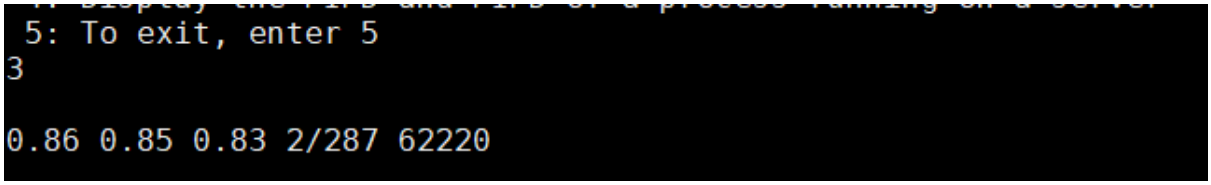


Screenshot Showing User Inputting 2:



Screenshot Showing User Inputting 3:





Screenshot Showing User Inputting 4:

```
PROGRAM MENU
Please input a number from the option menu...
1: Display information about the CPU
2: Display a list of device drivers currently configured
3: Display the load average of the system
4: Display the PIPD and PIPD of a process running on a server
5: To exit, enter 5
4
```

```
91992 39090292
9503 4921382
16684 4803185
13717 4837795
10040 4918972
10704 4909289
11503 4882071
10102 4904119
9736 4913477
9 0



5095699 5
```

Screenshot Showing User Inputting 5:

```
PROGRAM MENU
Please input a number from the option menu...
1: Display information about the CPU
2: Display a list of device drivers currently configured
3: Display the load average of the system
4: Display the PIPD and PIPD of a process running on a server
5: To exit, enter 5
5
```

```
 5: To exit, enter 5
5

Exiting now.....
akangd@hvs-its-lnx01:~$
```

Screenshot Showing User Inputting invalid entry:

```
 PROGRAM MENU
 Please input a number from the option menu...
 1: Display information about the CPU
 2: Display a list of device drivers currently configured
 3: Display the load average of the system
 4: Display the PIPD and PIPD of a process running on a server
 5: To exit, enter 5
0a
```

```
Oops invalid input, try again

 PROGRAM MENU
 Please input a number from the option menu...
 1: Display information about the CPU
 2: Display a list of device drivers currently configured
 3: Display the load average of the system
 4: Display the PIPD and PIPD of a process running on a server
 5: To exit, enter 5
```

Lab Activity 6 Memory Management

Basic Tasks

a. Memory Allocation (First Fit)

Criteria:

5 memory blocks available

5 processes require memory

Memory unallocated in block 1: 300 [M1]

Memory unallocated in block 2: 500 [M2]

Memory unallocated in block 3: 250 [M3]

Memory unallocated in block 4: 280 [M4]

Memory unallocated in block 5: 370 [M5]

Process 1 requires memory size of: 300 [P1]

Process 2 requires memory size of: 350 [P2]

Process 3 requires memory size of: 250 [P3]

Process 4 requires memory size of: 400 [P4]

First Fit Memory Allocation:

M1	M2	M3	M4	M5
P1	P2	P3	P5	X

For this task, process 4 does not go into any of the unallocated blocks as they are not any unallocated block with memory blocks enough to contain it.

b. Memory Allocation (Best fit)

Criteria:

5 memory blocks available

5 processes require memory

Memory unallocated in block 1: 300 [M1]

Memory unallocated in block 2: 500 [M2]

Memory unallocated in block 3: 250 [M3]

Memory unallocated in block 4: 280 [M4]

Memory unallocated in block 5: 370 [M5]

Process 1 requires memory size of: 300 [P1]

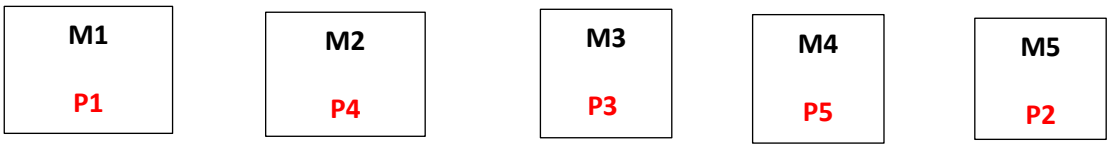
Process 2 requires memory size of: 350 [P2]

Process 3 requires memory size of: 250 [P3]

Process 4 requires memory size of: 400 [P4]

Process 5 requires memory size of: 170 [P5]

Best Fit Memory Allocation:



c. Memory Allocation (Worst fit)

Criteria:

5 memory blocks available

5 processes require memory

Memory unallocated in block 1: 300 [M1]

Memory unallocated in block 2: 500 [M2]

Memory unallocated in block 3: 250 [M3]

Memory unallocated in block 4: 280 [M4]

Memory unallocated in block 5: 370 [M5]

Process 1 requires memory size of: 300 [P1]

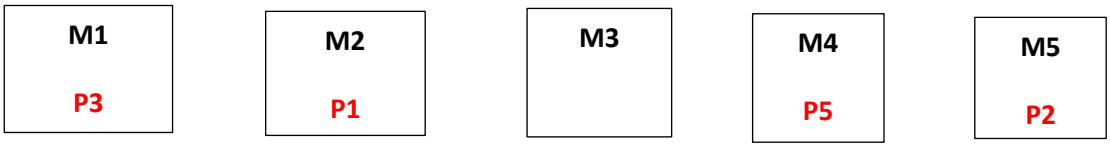
Process 2 requires memory size of: 350 [P2]

Process 3 requires memory size of: 250 [P3]

Process 4 requires memory size of: 400 [P4]

Process 5 requires memory size of: 170 [P5]

Worst Fit Memory Allocation:



P4 is not allocated as there is no suitable memory allocation.

d. Virtual Memory

Basic Task (a)

Random 12 Numbers: 12, 26, 38, 32, 46, 20, 20, 15, 2, 18,3

Paging Accessing Sequence	**12**	**26**	**38**	**32**	**46**	**20**	**20**	**15**	**2**	**18**	**3**
Page Entry 0	12	12	12	12	46	46	46	46	46	18	18
Page Entry 1		26	26	26	26	20	20	20	20	20	3
Page Entry 2			38	38	38	38	38	25	25	25	25
Page Entry 3				32	32	32	32	32	2	2	2
Page Fault	**1**	**2**	**3**	**4**	**5**	**6**		**7**	**8**	**9**	**10**

Page Fault Total: **10**

Basic Task (b)

Random 12 Numbers: 0, 7, 7, 9, 4, 4, 3, 9, 2, 5,3

Paging Accessing Sequence	**0**	**7**	**7**	**9**	**4**	**4**	**3**	**9**	**2**	**5**	**3**
Page Entry 0	0	0	0	0	0	0	0	0	2	2	2
Page Entry 1		7	7	7	7	7	3	3	3	5	5
Page Entry 2				9	9	9	9	9	9	9	9
Page Entry 3					4	4	4	4	4	4	3
Page Fault	**1**	**2**		**3**	**4**		**5**		**6**	**7**	**8**

Page Fault Total: 8

Paging Program

```
#include <iostream>       //cin and cout
#include <iomanip>        //setw
#include <vector>         //vector
#include <algorithm>     // std::find
using namespace std;

void printArray(vector <int>&pageSequence); //Vector decclaration, it is for
keeping the user inputs//
void    printPagingSequence(vector<int>&    frame0,    vector<int>&    frame1,
vector<int>& frame2, vector<int>& frame3); //Paasing the vectors by refernce
to be outputted/

int main()
{
        int sequenceNumber;
        const int FRAME_0 = 0; //1st column//
        const int FRAME_1 = 1; //2nd column//
        const int FRAME_2 = 2; //3rd column//
        const int FRAME_3 = 3; //4th column//
        int pageFault = 0; //Calculates number of page fault

        cout << "Kindly enter the length of the string to be entered: " <<
endl;
        cin >> sequenceNumber;


        int frameCount = 0; //Counter that keeps track of the current frame//

        vector <int>pageSequence; //Vector that holds user inputs//
        const int HALF_SCREEN_WIDTH = 40;
        cout << endl;
        int userInput;


        cout << "Enter the page numbers: " << endl;
        for (int i = 0; i < sequenceNumber; i++)
```

```
	{

		cin >> userInput;
		pageSequence.push_back(userInput);  //Push  user  input  into
vector

	}

	std::vector<int> frame0(pageSequence.size()); //Vector full of 0
	std::vector<int> frame1(pageSequence.size()); //Vector full of 0
	std::vector<int> frame2(pageSequence.size()); //Vector full of 0
	std::vector<int> frame3(pageSequence.size()); //Vector full of 0


	std::cout << std::string(HALF_SCREEN_WIDTH, ' ');
	cout << "The paging Sequence" << endl;
	printArray(pageSequence);


	for (int i = 0; i < pageSequence.size(); i++)
	{
		if (frameCount == FRAME_0)
		{
			if (i == 0)
			{
				i++;
				//Checks if current element is already in any
of the frame vectors
				if (pageSequence.at(i) == frame0.at(i-1) ||
pageSequence.at(i) == frame1.at(i-1) || pageSequence.at(i) == frame2.at(i-
1) || pageSequence.at(i) == frame3.at(i-1))
				{

				}
				else
				{
					i--;
					std::fill(frame0.begin()     +     i,
frame0.end(),  pageSequence.at(i));  //Fills  the  vector  from  the  current
position till the end with the current element
					frameCount++; //Increment frame count
					pageFault++;
				}

			}
			else
			{
				//Checks if current element is already in any
of the frame vectors
				if (pageSequence.at(i) == frame0.at(i - 1) ||
pageSequence.at(i) == frame1.at(i - 1) || pageSequence.at(i) == frame2.at(i
- 1) || pageSequence.at(i) == frame3.at(i - 1))
				{

				}
				else
				{
```

```
                                        std::fill(frame0.begin()     +     i,
frame0.end(), pageSequence.at(i));    //Fills the vector from the current
position till the end with the current element
                                        frameCount++; //Increment frame count
                                        pageFault++;

                                }
                        }


                }


                else if (frameCount == FRAME_1)
                {
                        //Checks if current element is already in any of the
frame vectors
                        if    (pageSequence.at(i)    ==    frame0.at(i-1)    ||
pageSequence.at(i) == frame1.at(i-1) || pageSequence.at(i) == frame2.at(i-
1) || pageSequence.at(i) == frame3.at(i-1))
                        {

                        }
                        else
                        {
                                std::fill(frame1.begin() + i, frame1.end(),
pageSequence.at(i)); // Fills the vector from the current position till the
end with the current element
                                frameCount++; //Increment frame count
                                pageFault++;

                        }


                }

                else if (frameCount == FRAME_2)
                {
                        //Checks if current element is already in any of the
frame vectors
                        if    (pageSequence.at(i)    ==    frame0.at(i-1)    ||
pageSequence.at(i) == frame1.at(i-1) || pageSequence.at(i) == frame2.at(i-
1) || pageSequence.at(i) == frame3.at(i-1))
                        {

                        }
                        else
                        {
                                std::fill(frame2.begin() + i, frame2.end(),
pageSequence.at(i)); // Fills the vector from the current position till the
end with the current element
                                frameCount++;
                                pageFault++;

                        }
```

```
                }

                else if (frameCount == FRAME_3)
                {
                        //Checks if current element is already in any of the
frame vectors
                        if    (pageSequence.at(i)    ==    frame0.at(i-1)    ||
pageSequence.at(i) == frame1.at(i-1) || pageSequence.at(i) == frame2.at(i-
1) || pageSequence.at(i) == frame3.at(i-1))
                        {

                        }
                        else
                        {
                                std::fill(frame3.begin() + i, frame3.end(),
pageSequence.at(i)); // Fills the vector from the current position till the
end with the current element
                                frameCount = 0;
                                pageFault++;

                        }


                }
                else
                {
                        frameCount = 0; //Reset frame count to 0
                }

        }

        printPagingSequence(frame0,  frame1,  frame2,  frame3);  //Fucntion
calling that prints frames
        cout << endl;
        cout << endl;
        cout << "The number of page faults is: " << pageFault << endl;

}

void printArray(vector<int>& pageSequence)
{
        const int HALF_SCREEN_WIDTH = 20;
        std::cout << std::string(HALF_SCREEN_WIDTH, ' ');

        for (int i = 0; i < pageSequence.size(); i++)
        {
                cout << pageSequence[i] <<setw(5); //Print out the sequence
entered by ther user
        }


}

void   printPagingSequence(vector<int>&   frame0,   vector<int>&   frame1,
vector<int>& frame2, vector<int>& frame3)
```

```
{
        cout << endl;
        cout << endl;

        const int HALF_SCREEN_WIDTH = 4;

        cout << " Page Frame 0 : ";
        std::cout << std::string(HALF_SCREEN_WIDTH, ' ');

        for (auto it = std::cbegin(frame0); it != std::cend(frame0); it++)
//Loops through vector and outputs each element
        {
                std::cout << *it << setw(5);
        }
        cout << endl;

        cout << " Page Frame 1 : ";
        std::cout << std::string(HALF_SCREEN_WIDTH, ' ');

        for (auto it = std::cbegin(frame1); it != std::cend(frame1); it++)
//Loops through vector and outputs each element
        {
                std::cout << *it << setw(5);
        }
        cout << endl;

        cout << " Page Frame 2 : ";
        std::cout << std::string(HALF_SCREEN_WIDTH, ' ');

        for (auto it = std::cbegin(frame2); it != std::cend(frame2); it++)
//Loops through vector and outputs each element
        {
                std::cout << *it << setw(5);
        }
        cout << endl;
        cout << " Page Frame 3 : ";
        std::cout << std::string(HALF_SCREEN_WIDTH, ' ');


        for (auto it = std::cbegin(frame3); it != std::cend(frame3); it++)
//Loops through vector and outputs each element
        {
                std::cout << *it << setw(5);
        }

}
```

Code Output/ Simulation:

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab6-207SE$  g++ -o PagingProgram PagingProgram.cpp
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab6-207SE$ ./PagingProgram
Kindly enter the length of the string to be entered:
11

Enter the page numbers:
4
2
7
7
5
6
3
9
3
2
2
                                The paging Sequence
                   4    2    7    7    5    6    3    9    3    2    2

 Page Frame 0 :    4    4    4    4    4    6    6    6    6    6    6
 Page Frame 1 :    0    2    2    2    2    2    3    3    3    3    3
 Page Frame 2 :    0    0    7    7    7    7    7    9    9    9    9
 Page Frame 3 :    0    0    0    0    5    5    5    5    5    2    2

The number of page faults is: 8
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab6-207SE$
```

```
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab6-207SE$ g++ -o PagingProgram PagingProgram.cpp
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab6-207SE$ ./PagingProgram
Kindly enter the length of the string to be entered:
6

Enter the page numbers:
1
9
1
2
4
7
                                The paging Sequence
                   1    9    1    2    4    7

 Page Frame 0 :    1    1    1    1    1    7
 Page Frame 1 :    0    9    9    9    9    9
 Page Frame 2 :    0    0    0    2    2    2
 Page Frame 3 :    0    0    0    0    4    4

The number of page faults is: 5
akangd@hvs-its-lnx01:~/AKANG207SE2020/Portfolio1-2020/Lab6-207SE$
```

Paging Program Video Implementation:

[Paging Program]

Lab Activity 8 Cache Buffer

a. Brief Description of Cache Buffer Activity

This activity is centred around creating a cache library, a function will be written which would check if the cache needs filling, if the buffer needs refilling, then the refill function will refill it. Also, the buffer will also return characters in cache_printer, the characters returned can be fed into data structures for observation and manipulation. The buffer has a struct which has a variable called alongBuffer which holds the current position in the buffer. It holds the name of the file as well as the buffer length.

b. Commented implementation of the cr_handle function

Function Definition:

```
//Read a byte.  Will return EOF if empty.
char return_character(bufferStruct* buff);
```

Function Declaration:

```
char return_character(bufferStruct* buff) //Function Declaration
{
    buffer_refill(buff); //Function that checks if buffer needs refill and refills it

    char charToReturn; //Variable to store character to return
    charToReturn = buff->buffer[buff->alongBuffer]; // the character to return is stored in the variable created

    buff->alongBuffer++; //The current position in the buffer is incremented by 1 step

    return charToReturn;
}
```

Proof of Compilation:

```
akangd@hvs-its-lnx01:~/207SE_Sessions/Session8/cache-handle$ make
gcc -std=c99 -g -o cache_printer cache_printer.c cache_handle.o
akangd@hvs-its-lnx01:~/207SE_Sessions/Session8/cache-handle$ ./cache_printer
```

Code Output:

```
Iran hacked 9,000 UK emails in 'brute force' cyber attack that was blamed on Russia
Iran has carried out a ▒brute force▒ cyber attack on Parliament that compromised MP▒s email accounts, according to a secret intelligence assessment. Aroun
d 9,000 email accounts, including those belonging to Theresa May and other Cabinet Ministers, were hacked in the 12-hour "sustained and determined" attack
 June 23.
Russia was initially blamed but investigators have traced the source of the hit to the Tehran regime, according to The Times. The House of Commons said it
 did not comment on security matters. A National Cyber Security Centre spokesman said: "It would be inappropriate to comment further while enquiries are o
ngoing."
The attack could also be that Tehran was seeking information to gain a commercial advantage.
Sources described the regime as ▒highly capable actors in the cyber world▒.
Another said: ▒It was the not most sophisticated attack but nor did it need to be.
▒It is possible they were simply testing their capability.▒
Downing Street did not comment but a senior British official acknowledged that the revelation had complicated Mrs May▒s response to Mr Trump.
The revelations come as Britain and other European powers have been trying to keep the Iran nuclear deal on track after President Donald Trump's refusal t
o back it.
The Prime Minister joined Germany's Angela Merkel and France's Emmanuel Macron to issue a statement insisting preserving the pact.
They said it was "in our shared national security interest" and they have called for Washington to "consider the implications" of undermining it.
A statement from the UK, France and Germany said the International Atomic Energy Agency has "repeatedly confirmed" Iran's compliance with the terms it sig
ned up to.
It said: "We, the leaders of France, Germany and the United Kingdom take note of President Trump's decision not to recertify Iran's compliance with the Jo
int Comprehensive Plan of Action (JCPA) to Congress and are concerned by the possible implications. "We stand committed to the JCPA and its full implement
ation by all sides. Preserving the JCPA is in our shared national security interest. "The nuclear deal was the culmination of 13 years of diplomacy and wa
s a major step towards ensuring that Iran's nuclear programme is not diverted for military purposes." Trump accused Tehran of violating the spirit of the
landmark 2015 agreement and believes the international community is being naive in its dealings with the regime.
The President stopped short of ripping up the deal but said without measures to toughen it up "the agreement will be terminated".
Shadow foreign secretary Emily Thornberry said it was "high time" the Government challenged Mr Trump on his actions and accused him of ▒vandalism▒.
She said: "It is an act of wanton vandalism for Donald Trump to jeopardise the future of that deal today, and to move the goalposts by linking it to impor
tant but utterly extraneous issues around Iran's wider activities in the region.
A brute force cyber attack on Parliament that compromised MPs' email accounts was carried out by Iran, it has emerged.
Blackmail fears were raised when hackers tried to break into the system used by MPs, peers and staff by searching for weak passwords.
Around 90 of the 9,000 email accounts were undermined in the "sustained and determined" attack in June.
Donald Trump's speech on Iran may actually end up saving nuclear deal
Russia faced accusations it was behind the attack but investigators have traced the source of the hit to the Tehran regime, according to The Times.
The House of Commons said it did not comment on security matters.
A National Cyber Security Centre spokesman said: "It would be inappropriate to comment further while enquiries are ongoing."
The US president accused Tehran of violating the spirit of the landmark 2015 agreement and believes the international community is being naive in its deal
ings with the regime.
Theresa May joined Germany's Angela Merkel and France's Emmanuel Macron to issue a statement insisting preserving the pact was "in our shared national sec
urity interest" and calling for Washington to "consider the implications" of taking action that undermine it.akangd@hvs-its-lnx01:~/207SE_Sessions/Session
8/cache-handle$
```

c. Comment updated code to show that each byte is being read, and when the buffer is being refilled.

For this activity, a char is one byte, so to show that each byte is being read, a (*) would be added after every character. To show when the buffer is being refilled a prompt will be outputted.

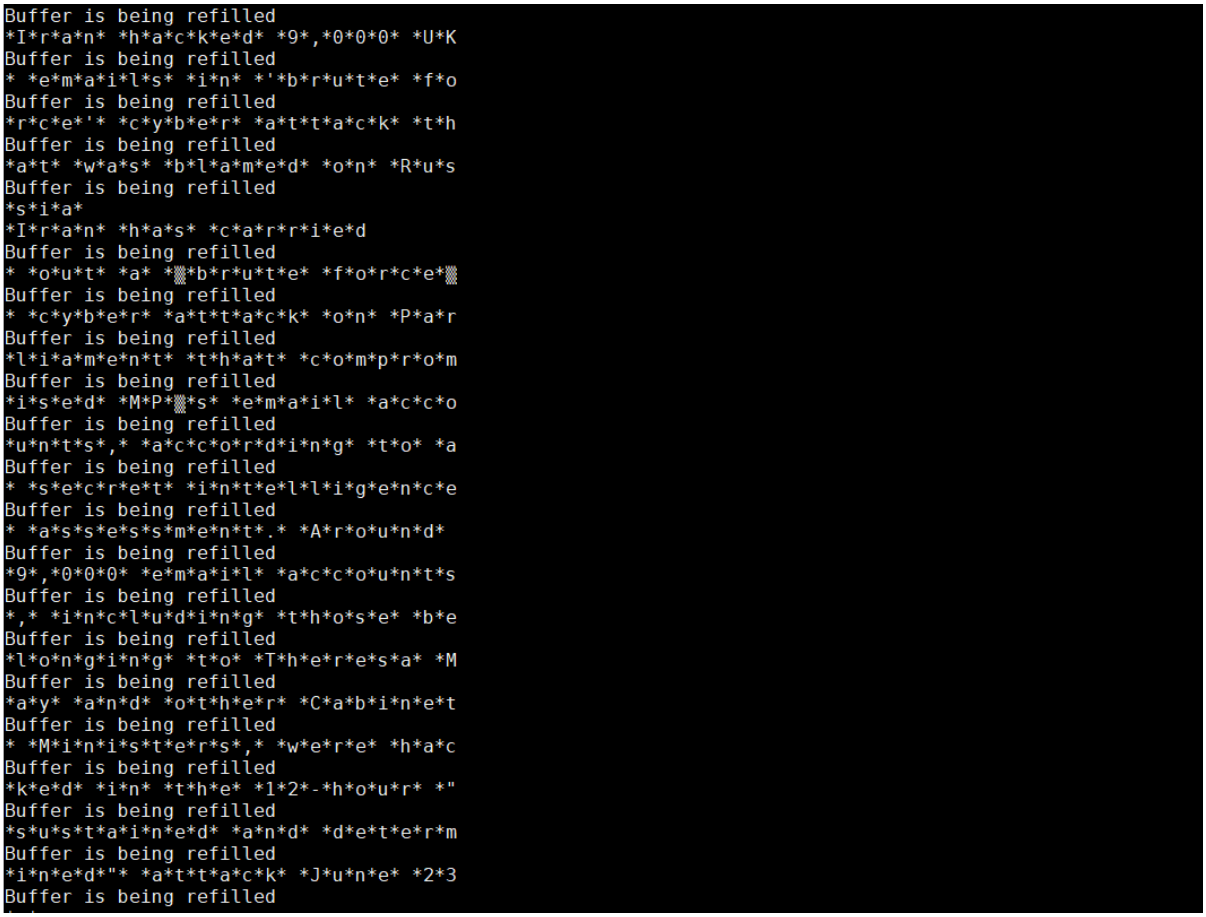
Updated Code responsible for each byte is being read

```c
char return_character(bufferStruct* buff) //Function Declaration
{
    buffer_refill(buff);

    char charToReturn; //Variable to store character to return
    charToReturn = buff->buffer[buff->alongBuffer]; // the character to return is stored in the variable created
    printf("*"); // The + means represeents that each byte is being read//
    buff->alongBuffer++; //The current position in the buffer is incremented by 1 step

    return charToReturn;
}
```

Updated Code responsible for showing when buffer is refilled

```c
int buffer_refill(bufferStruct* buff) {
    //Refills a buffer
    //Only works when completely used buffer

    if (buff->alongBuffer != buff->bufferLength)
        return 0;
    else {
        count++;
        printf("\n"); //Newline
       printf("Buffer is being refilled"); //when the code branches to this else, a buffer output is shown
        printf("\n"); //New line
        buff->alongBuffer = 0;
        int len = fread(buff->buffer, sizeof(char), buff->bufferLength, buff->file);
        //If we didn't fill the buffer, fill up with EOF
        if (len < buff->bufferLength)
            for (int i = len; i < buff->bufferLength; i++)
                buff->buffer[i] = EOF;  //Accessing like an array!
        return len;
    }
```

Code output showing byte being read and buffer being refilled



d. Commented updated code showing the required statistical information as well as how many times the words ‘Iran’, ‘Tehran’ and ‘email’ appear.

Code from cache_handle.h

```
#pragma once
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//The internals of this struct aren't important
//from the user's point of view

extern int count;
typedef struct {
    FILE* file;         //File being read
    int bufferLength;   //Fixed buffer length
    int alongBuffer;     //Current point in the buffer
    char* buffer;       //A pointer to a piece of memory                    //
same length as "bufferlength"
} bufferStruct;
```

```
static int arrayPosition = 0;
char tempArray[6];

void compareWord(char a, int* iranWordCount, int* tehranWordCount, int*
emailWordCount, int* nonActiveWordCount, char myArray[], char myArray1[],
char myArray2[], char* nonActiveWordArray[], int *
NON_ACTIVE_WORD_ARRAY_COUNT, char nextChar); //Function declaration

//Open a file with a given size of buffer to cache with
bufferStruct* file_open(char* filename, int buffersize);
//Close an open file
void file_close(bufferStruct* buff);
//Read a byte.  Will return EOF if empty.
char return_character(bufferStruct* buff, char * nextChar);
//-------------------------------------------------------

//Refill an empty buffer.  Not intended for users
int buffer_refill(bufferStruct* buff);
```

Code from cache_handle.c

```
#include "cache_handle.h"

#pragma warning(disable : 4996)
count = 0;

int buffer_refill(bufferStruct* buff) {
    //Refills a buffer
    //Only works when completely used buffer

    if (buff->alongBuffer != buff->bufferLength)
        return 0;
    else {
        count++;
        // printf("\n"); //Newline
       // printf("Buffer is being refilled"); //when the code branches to
this else, a buffer output is shown
       //  printf("\n"); //New line
        buff->alongBuffer = 0;
        int len = fread(buff->buffer, sizeof(char), buff->bufferLength,
buff->file);
        //If we didn't fill the buffer, fill up with EOF
        if (len < buff->bufferLength)
            for (int i = len; i < buff->bufferLength; i++)
                buff->buffer[i] = EOF;  //Accessing like an array!
        return len;
    }

}
void file_close(bufferStruct* buff) {
    free(buff->buffer);
    fclose(buff->file);
}
```

```
void compareWord(char a, int* iranWordCount, int* tehranWordCount, int*
emailWordCount, int* nonActiveWordCount, char myArray[], char myArray1[],
char myArray2[], char* nonActiveWordArray[] , int *
NON_ACTIVE_WORD_ARRAY_COUNT, char nextChar)
{
   tempArray[arrayPosition] = a; //Pushing the character values returned into
a temporary array
    arrayPosition++; //Inrement the current array position


    for (int i = 0; i < NON_ACTIVE_WORD_ARRAY_COUNT; i++)
    {
        if (strcmp((nonActiveWordArray[i]), tempArray) == 0) //If the value
in the array is equal to the required word
        {
            if (nextChar == ' ' || nextChar == '\n')
            {

                (*nonActiveWordCount)++; //Increment iran word count passed
in to this function by reference
                memset(tempArray, '\0', sizeof tempArray + 1); // for
clearing the array created//
            }
        }

    }

    if (strcmp(myArray, tempArray) == 0) //If the value in the array is equal
to the required word
    {
        (*iranWordCount)++; //Increment iran word count passed in to this
function by reference
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (strcmp(myArray1, tempArray) == 0)  //If the value in the array is
equal to the required word
    {
        (*tehranWordCount)++; //Increment tehran word count passed in to this
function by reference
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (strcmp(myArray2, tempArray) == 0) //If the value in the array is
equal to the required word
    {
        (*emailWordCount)++; //Increment email word count passed in to this
function by reference
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (a == ' ' || a == '\n') // If there is a space, that means we have
read a word and can reset array position
    {
```

```
        arrayPosition = 0;
        memset(tempArray, '\0', sizeof tempArray + 1);       //Clear temop
array

    }
}

bufferStruct* file_open(char* filename, int buffersize) {

    //Info on malloc
    //http://www.space.unibe.ch/comp_doc/c_manual/C/FUNCTIONS/malloc.html
    FILE* f;
    if ((f = fopen(filename, "r")) == NULL) {
        fprintf(stderr, "Cannot open %s\n", filename);
        return 0;
    }

    bufferStruct* initBuffer = (bufferStruct*)malloc(sizeof(bufferStruct));
    initBuffer->file = f;
    initBuffer->bufferLength = buffersize;
    initBuffer->alongBuffer = buffersize; //Start off with no characters, so
refill will work as expected
    initBuffer->buffer = (char*)malloc(sizeof(char) * buffersize);

    buffer_refill(initBuffer);
    return initBuffer;
}

char   return_character(bufferStruct*   buff,   char   *nextChar)   //Function
Declaration
{
    buffer_refill(buff);

    char charToReturn; //Variable to store character to return
    charToReturn  =  buff->buffer[buff->alongBuffer];  //  the  character  to
return is stored in the variable created
    buff->alongBuffer++; //The current position in the buffer is incremented
by 1 step

    *(nextChar) = buff->buffer[buff->alongBuffer]; //For fidning if the next
position is a space
    buff->alongBuffer--; //Go back to previous position

   // printf("*"); // The + means represeents that each byte is being read//
    buff->alongBuffer++; //The current position in the buffer is incremented
by 1 step

    return charToReturn;

}
```

Code from cache_printer.c

```
#include "cache_handle.h"
/*Updated changes showing statistics like how many how many vowels and other
characters were read in total, how many non - active words were read in, how
many sentences, how many times the words
'Iran', 'Tehran'and 'email' appearand how many times the buffer
was refilled. */
int main() {
    char character;
    char nextChar;
    char* nextCharPtr = &nextChar;
    int vowelCount = 0; //Variable for counting the number of vowels//



    int otherCharCount = 0; //Variable for counting the number of other
characters//
    int sentences = 0; //Variable for counting the number of sentences//

    int iranWordCount = 0; //Vraiable declartion for the iran word count//
    int* iranWordPtr = &iranWordCount;

    int tehranWordCount = 0;//Vraiable declartion for the tehran word count//
    int* tehranWordPtr = &tehranWordCount;

    int emailWordCount = 0;//Vraiable declartion for the email word count//
    int* emailWordPtr = &emailWordCount;

    int bufferRefillCount = 0;//Vraiable declartion for the email word
count//
    int* bufferRefillPtr = &bufferRefillCount;

    int nonActiveWordCount = 0;
    int* nonActiveWordPtr = &nonActiveWordCount;

    char myArray[] = "Iran"; //String of characters assigned with the value
Iran has been declared//
    char myArray1[] = "Tehran"; //String of characters assigned with the
value Tehran has been declared//
    char myArray2[] = "email"; //String of characters assigned with the value
email has been declared//
    char* nonActiveWordArray[] = { "be", "own", "sound", "appear", "smell",
"possess", "feel", "want", "prefer", "am", "are", "was", "were", "is", "had"
};
    const int NON_ACTIVE_WORD_ARRAY_COUNT = 15;
    //Open a file
    bufferStruct* f = file_open("text", 20);
    //While there are useful bytes coming from it
    while ((character = return_character(f, nextCharPtr)) != EOF)
    {
        compareWord(character, iranWordPtr, tehranWordPtr, emailWordPtr,
nonActiveWordPtr, myArray, myArray1, myArray2, nonActiveWordArray,
NON_ACTIVE_WORD_ARRAY_COUNT, nextChar); //Function call with paramters
passed
```

```
        if (character == '.')  //At the end of sentences is a full stop, so
the count gets incremented
        {
            sentences++;
        }

        if (character == 'a' || character == 'e' || character == 'i' ||
character == 'o' || character == 'u') //if a vowel is found, the count is
incremented
        {
            vowelCount++;
        }
        else
        {
            otherCharCount++; //any other character that is not a vowel is
incremented
        }
        //Print them
       /* printf("%c", character);*/


    }
    printf("\n");
    printf("The vowel count is: %d", vowelCount);
    printf("\n");
    printf("The count of other chars are: %d", otherCharCount);
    printf("\n");
    printf("The number of sentences is: %d", sentences);
    printf("\n");
    printf("The number of words with Iran is: %d", iranWordCount);
    printf("\n");
    printf("The number of words with Tehran is: %d", tehranWordCount);
    printf("\n");
    printf("The number of words with email is: %d", emailWordCount);
    printf("\n")
    printf("The number of times the buffer was refilled is : %d", count);
    printf("\n");

    printf("The number of non active words found is : %d",
nonActiveWordCount);
    printf("\n");




    //Then close the file
    file_close(f);

    //And finish
    return 0;
}
```

Output of running code

```
The vowel count is: 1274
The count of other chars are: 2953
The number of sentences is: 30
The number of words with Iran is: 9
The number of words with Tehran is: 5
The number of words with email is: 5
The number of times the buffer was refilled is : 212
The number of non active words found is : 28
```

Lab 10: The Cache Buffer from week 8 with system calls

a. Brief description of the activity

For this activity, we would use the cache library previously used in Lab 8. For that task fopen, fread and fclose function calls and declarations were used to manipulate the file. For this task, system calls would be used instead, the system calls used here would open, read and close. Each of these system calls will have certain parameters so when they are called, the appropriate parameters will be passed. Also, each of the system calls have their libraries which will have to be included in the header file.

b. Changes the cache_handle library from using the fopen, fread, fclose functions to the system call versions open, read, close

Commented Code showing .h file changes

```
#pragma once
#include <stdio.h>  //Standard input and output functions
#include <stdlib.h>
#include<sys/types.h> // used for opening the file
#include <sys/stat.h> //used for reading and closing the file
#include <fcntl.h> //used for reading only
#include <string.h>


extern int count;
typedef struct {
    int file;         //File being read- Open file returns an integer
    int bufferLength;  //Fixed buffer length
    int alongBuffer;     //Current point in the buffer
    char* buffer;       //A pointer to a piece of memory


                        //  same length as "bufferlength"
} sy_file;

static int arrayPosition = 0;



char tempArray[6];

sy_file * file_open(char* filename, int buffersize); //Open a given file
with the filename and buffersize passed as parameters//
```

```
void compareWord(char a, int* iranWordCount, int* tehranWordCount, int*
emailWordCount, char myArray[],char myArray1[], char myArray2[]);

int refillCount(int * b);

void file_close(sy_file * a); //Close an open file

//Read a byte.  Will return EOF if empty.
char return_character(sy_file * a);



//-------------------------------------------------------

//Refill an empty buffer.  Not intended for users
int buffer_refill(sy_file * buff);
```

Output showing .c file changes

```
#pragma once
#include "cache_handle.h"
#pragma warning(disable : 4996)


//http://www.phim.unibe.ch/comp_doc/c_manual/C/SYNTAX/struct.html
//http://vergil.chemistry.gatech.edu/resources/programming/c-
tutorial/structs.html

count = 0;

int buffer_refill(sy_file * buff) {
    //Refills a buffer
    //Only works when completely used buffer
    if (buff->alongBuffer != buff->bufferLength)
        return 0;
    else {
        count++;
        buff->alongBuffer = 0;
        int len = read(buff->file, buff->buffer , buff->bufferLength);
//reads files up to bufferlength bytes into the buffer starting at buffer
        //If we didn't fill the buffer, fill up with EOF
        if (len < buff->bufferLength)
            for (int i = len; i < buff->bufferLength; i++)
                buff->buffer[i] = EOF;  //Accessing like an array!
        return len;
    }
}

void file_close(sy_file * a) {
    free(a->buffer); //free the buffer and meory allocated
    close(a->file); //close the file
}
```

```
void compareWord(char a, int* iranWordCount, int* tehranWordCount, int*
emailWordCount, char myArray[], char myArray1[], char myArray2[])
{



    tempArray[arrayPosition] = a; //Putting the values  for Iraninto the
array to compare with the string
    arrayPosition++;

    if (strcmp(myArray, tempArray) == 0)
    {
        (*iranWordCount)++;
        memset(tempArray, '\0', sizeof tempArray +1); // for clearing the
array created//
    }

    if (strcmp(myArray1, tempArray) == 0)
    {
        (*tehranWordCount)++;
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (strcmp(myArray2, tempArray) == 0)
    {
        (*emailWordCount)++;
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (a == ' ' || a == '\n')
    {
        arrayPosition = 0;
        memset(tempArray, '\0', sizeof tempArray +1);
        memset(tempArray, '\0', sizeof tempArray + 1);
        memset(tempArray, '\0', sizeof tempArray + 1);

    }



}



sy_file * file_open(char* filename, int buffersize) {

    //Info on malloc
    //http://www.space.unibe.ch/comp_doc/c_manual/C/FUNCTIONS/malloc.html
    int f;
    if ((f = open(filename, O_RDONLY)) == -1) { //If the file is not opened
sucessfully, open returns -1 indicating an error has occured
        fprintf(stderr, "Cannot open %s\n", filename); //Cannot open file
prompt is outputted
        return 0;
    }
```

```
    sy_file* b = (sy_file*)malloc(sizeof(sy_file));
    b->file = f;
    b->bufferLength = buffersize;
    b->alongBuffer = buffersize; //Start off with no characters, so refill
will work as expected
    b->buffer = (char*)malloc(sizeof(char) * buffersize); //Allocates a
memory block of  size- buffersize

    buffer_refill(b); //Refill is called
    return b;
}



//-----------------------------------------------------------------
char return_character(sy_file * a) //Function Declaration
{
    buffer_refill(a);


    char charToReturn; //Variable to store character to return
    charToReturn = a->buffer[a->alongBuffer]; // the character to return is
stored in the variable created
   // printf("+"); // The + means represeents that each byte is being
read//
    a->alongBuffer++; //The current position in the buffer is incremented
by 1 step



    return charToReturn;
}
```

[Output from running code]

```
akangd@hvs-its-lnx01:~/207SE_Sessions/Session10/cache-handle$ make
gcc -std=c99 -g -o cache_printer cache_printer.c cache_handle.o
akangd@hvs-its-lnx01:~/207SE_Sessions/Session10/cache-handle$ ./cache_printer
Iran hacked 9,000 UK emails in 'brute force' cyber attack that was blamed on Russia
Iran has carried out a ▒brute force▒ cyber attack on Parliament that compromised MP▒s email accounts, according to a secret intelli
gence assessment. Around 9,000 email accounts, including those belonging to Theresa May and other Cabinet Ministers, were hacked in
 the 12-hour "sustained and determined" attack June 23.
Russia was initially blamed but investigators have traced the source of the hit to the Tehran regime, according to The Times. The H
ouse of Commons said it did not comment on security matters. A National Cyber Security Centre spokesman said: "It would be inapprop
riate to comment further while enquiries are ongoing."
The attack could also be that Tehran was seeking information to gain a commercial advantage.
Sources described the regime as ▒highly capable actors in the cyber world▒.
Another said: ▒It was the not most sophisticated attack but nor did it need to be.
▒It is possible they were simply testing their capability.▒
Downing Street did not comment but a senior British official acknowledged that the revelation had complicated Mrs May▒s response to
 Mr Trump.
The revelations come as Britain and other European powers have been trying to keep the Iran nuclear deal on track after President D
onald Trump's refusal to back it.
The Prime Minister joined Germany's Angela Merkel and France's Emmanuel Macron to issue a statement insisting preserving the pact.
They said it was "in our shared national security interest" and they have called for Washington to "consider the implications" of u
ndermining it.
A statement from the UK, France and Germany said the International Atomic Energy Agency has "repeatedly confirmed" Iran's complianc
e with the terms it signed up to.
It said: "We, the leaders of France, Germany and the United Kingdom take note of President Trump's decision not to recertify Iran's
 compliance with the Joint Comprehensive Plan of Action (JCPA) to Congress and are concerned by the possible implications. "We stan
d committed to the JCPA and its full implementation by all sides. Preserving the JCPA is in our shared national security interest.
"The nuclear deal was the culmination of 13 years of diplomacy and was a major step towards ensuring that Iran's nuclear programme
is not diverted for military purposes." Trump accused Tehran of violating the spirit of the landmark 2015 agreement and believes th
e international community is being naive in its dealings with the regime.
The President stopped short of ripping up the deal but said without measures to toughen it up "the agreement will be terminated".
Shadow foreign secretary Emily Thornberry said it was "high time" the Government challenged Mr Trump on his actions and accused him
 of ▒vandalism▒.
She said: "It is an act of wanton vandalism for Donald Trump to jeopardise the future of that deal today, and to move the goalposts
 by linking it to important but utterly extraneous issues around Iran's wider activities in the region.
A brute force cyber attack on Parliament that compromised MPs' email accounts was carried out by Iran, it has emerged.
Blackmail fears were raised when hackers tried to break into the system used by MPs, peers and staff by searching for weak password
s.
Around 90 of the 9,000 email accounts were undermined in the "sustained and determined" attack in June.
```

c. Changes cache_handle library to remove (as far as possible) the effects of caching on the library.

Cache handle.c file:

```c
#pragma once
#include "cache_handle.h"
#pragma warning(disable : 4996)


//http://www.phim.unibe.ch/comp_doc/c_manual/C/SYNTAX/struct.html
//http://vergil.chemistry.gatech.edu/resources/programming/c-
tutorial/structs.html

count = 0;

int buffer_refill(sy_file * buff) {
    //Refills a buffer
    //Only works when completely used buffer
    if (buff->alongBuffer != buff->bufferLength)
        return 0;
    else {
        count++;
```

```
        buff->alongBuffer = 0;
        int len = read(buff->file, buff->buffer , buff->bufferLength);
//reads files up to bufferlength bytes into the buffer starting at buffer
        //If we didn't fill the buffer, fill up with EOF
        if (len < buff->bufferLength)
            for (int i = len; i < buff->bufferLength; i++)
                buff->buffer[i] = EOF;  //Accessing like an array!
        return len;
    }

}

void file_close(sy_file * a) {
    free(a->buffer); //free the buffer and meory allocated
    close(a->file); //close the file
}


void compareWord(char a, int* iranWordCount, int* tehranWordCount, int*
emailWordCount, char myArray[], char myArray1[], char myArray2[])
{



    tempArray[arrayPosition] = a; //Putting the values  for Iraninto the
array to compare with the string
    arrayPosition++;

    if (strcmp(myArray, tempArray) == 0)
    {
        (*iranWordCount)++;
        memset(tempArray, '\0', sizeof tempArray +1); // for clearing the
array created//
    }

    if (strcmp(myArray1, tempArray) == 0)
    {
        (*tehranWordCount)++;
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (strcmp(myArray2, tempArray) == 0)
    {
        (*emailWordCount)++;
        memset(tempArray, '\0', sizeof tempArray + 1); // for clearing the
array created//
    }

    if (a == ' ' || a == '\n')
    {
        arrayPosition = 0;
        memset(tempArray, '\0', sizeof tempArray +1);


    }
```

```
}

sy_file * file_open(char* filename, int buffersize) {

    //Info on malloc
    //http://www.space.unibe.ch/comp_doc/c_manual/C/FUNCTIONS/malloc.html
    int f;
    if ((f = open(filename, O_RDONLY|__O_DIRECT|O_SYNC) == -1)) { //If the
file is not opened sucessfully, open returns -1 indicating an error has
occured
    // O_DIRECT minimize cache effects of the  I/O to and from this file.
O_RDONLY -- requires the caller to have read only permissions on the object
-- O_sync tries to give the guaranatee that data and neccessary meta data
are transferred.
        fprintf(stderr, "Cannot open %s\n", filename); //Cannot open file
prompt is outputted
        return 0;

    }



    sy_file* b = (sy_file*)malloc(sizeof(sy_file));
    b->file = f;
    b->bufferLength = 512;
    b->alongBuffer = 512; //Start off with no characters, so refill will
work as expected
    b->buffer = (char*)memalign(sizeof(char)*512,  sizeof(char)*512); //
Memalign accepts 512 as the least size

    buffer_refill(b); //Refill is called
    return b;
}




//-----------------------------------------------------------------------
char return_character(sy_file * a) //Function Declaration
{
    buffer_refill(a);


    char charToReturn; //Variable to store character to return
    charToReturn = a->buffer[a->alongBuffer]; // the character to return is
stored in the variable created
   // printf("+"); // The + means represeents that each byte is being
read//
    a->alongBuffer++; //The current position in the buffer is incremented
by 1 step


    return charToReturn;
}
```

Code Output:

```
akangd@hvs-its-lnx01:~/207SE_Sessions/Session10/cache-handle$ make
gcc -std=c99 -g -o cache_printer cache_printer.c cache_handle.o
akangd@hvs-its-lnx01:~/207SE_Sessions/Session10/cache-handle$ ./cache_printer
Iran hacked 9,000 UK emails in 'brute force' cyber attack that was blamed on Russia
Iran has carried out a ▒brute force▒ cyber attack on Parliament that compromised MP▒s email accounts, according to a secret intelli
gence assessment. Around 9,000 email accounts, including those belonging to Theresa May and other Cabinet Ministers, were hacked in
 the 12-hour "sustained and determined" attack June 23.
Russia was initially blamed but investigators have traced the source of the hit to the Tehran regime, according to The Times. The H
ouse of Commons said it did not comment on security matters. A National Cyber Security Centre spokesman said: "It would be inapprop
riate to comment further while enquiries are ongoing."
The attack could also be that Tehran was seeking information to gain a commercial advantage.
Sources described the regime as ▒highly capable actors in the cyber world▒.
Another said: ▒It was the not most sophisticated attack but nor did it need to be.
▒It is possible they were simply testing their capability.▒
Downing Street did not comment but a senior British official acknowledged that the revelation had complicated Mrs May▒s response to
 Mr Trump.
The revelations come as Britain and other European powers have been trying to keep the Iran nuclear deal on track after President D
onald Trump's refusal to back it.
The Prime Minister joined Germany's Angela Merkel and France's Emmanuel Macron to issue a statement insisting preserving the pact.
They said it was "in our shared national security interest" and they have called for Washington to "consider the implications" of u
ndermining it.
A statement from the UK, France and Germany said the International Atomic Energy Agency has "repeatedly confirmed" Iran's complianc
e with the terms it signed up to.
It said: "We, the leaders of France, Germany and the United Kingdom take note of President Trump's decision not to recertify Iran's
 compliance with the Joint Comprehensive Plan of Action (JCPA) to Congress and are concerned by the possible implications. "We stan
d committed to the JCPA and its full implementation by all sides. Preserving the JCPA is in our shared national security interest.
"The nuclear deal was the culmination of 13 years of diplomacy and was a major step towards ensuring that Iran's nuclear programme
is not diverted for military purposes." Trump accused Tehran of violating the spirit of the landmark 2015 agreement and believes th
e international community is being naive in its dealings with the regime.
The President stopped short of ripping up the deal but said without measures to toughen it up "the agreement will be terminated".
Shadow foreign secretary Emily Thornberry said it was "high time" the Government challenged Mr Trump on his actions and accused him
 of ▒vandalism▒.
She said: "It is an act of wanton vandalism for Donald Trump to jeopardise the future of that deal today, and to move the goalposts
 by linking it to important but utterly extraneous issues around Iran's wider activities in the region.
A brute force cyber attack on Parliament that compromised MPs' email accounts was carried out by Iran, it has emerged.
Blackmail fears were raised when hackers tried to break into the system used by MPs, peers and staff by searching for weak password
s.
Around 90 of the 9,000 email accounts were undermined in the "sustained and determined" attack in June.
```

References

Typical Architecture Of An Embedded System (n.d.) available from https://ebrary.net/22041/computer_science/typical_architecture_embedded_system> [27 January 2020]

Vega, A., Bose, P. and Buyuktosunoglu, A. (2017) Rugged Embedded Systems. 1st edn. Amsterdam: Morgan Kaufmann

Olivka, P. (2020) Architecture Of Computers And Parallel Systems [online] available from http://poli.cs.vsb.cz/edu/apps/eng/p01-archcomp.pdf [27 January 2020]

Teach ICT GCSE Computer Science AQA 8520 Vonneumann Layout (n.d.) available from http://teach-ict.com/2016/GCSE_Computing/AQA_8520/3_4_computer_systems/344_systems_architecture/von_neumann/miniweb/pg2.htm [27 January 2020]

Typical Architecture Of An Embedded System (n.d.) available from https://ebrary.net/22041/computer_science/typical_architecture_embedded_system [27 January 2020]

Von Neumann Architecture | History & Use | Computer Science (n.d.) available from https://teachcomputerscience.com/von-neumann-architecture/ [28 January 2020]

Techopedia.com. (n.d.). What is a Boot Loader? [online] Available at: https://www.techopedia.com/definition/3324/boot-loader [Accessed 16 Feb. 2020].

Mckay, D. (2019) *How To Use The Rename Command On Linux* [online] available from https://www.howtogeek.com/423214/how-to-use-the-rename-command-on-linux/ [24 January 2020]