Name: VENELIN DIMITROV

Student id: 6297262

Course: Computer Science

Module: 207SE Operating Systems, Security and Networks

Submission date: 27nd February

Portfolio 1

Lab Activity 1 – Operating Systems Tasks and Programming

a) Future of OS

For the last 25 years operating systems have changed multiple times. From being a bleeping line in a black screen in the command line, to a complicated graphical interface with multiple working environments and millions of software applications to choose from. Main goal for every OS developer has been creating a simplified interface so every user can control and customise their system by personal taste. Demands by users have changed and now just simplicity is not enough, everyone is using daily at least 4 different devices with every one of them having their own applications and just only a few of them having the functionality of multi-platforming.

- Next generation OS will have smaller size and running entirely in the cloud.
- Tracking of all applications and providing a single point of access for them
- Simplified upgrades, management and portability across platforms
- Having an artificial intelligence as an assistant to everyday tasks and researches.
- Encryption and security

b) Programming activity

PYTHON

```
#Input for the student name and his ID
student = input ("Enter the name of the student: ")
id = input ("Enter the first digit of id: ")

#Loop takes the id number and splits the students name into parts equal to the number
which was stored.
for i in range(0,len(student),int(id)):
    #prints and each split on a new line
    print(student[i:i+int(id)])
```

OUTPUT

```
Enter your name: Venelin Dimitrov
Enter first digit of your id: 1
V
e
n
e
l
i
n

D
i
m
i
t
r
o
v
>>>
```

C++

```cpp

#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int main()
{
    string name = "";
    string input = "";
    //We collect the needed Name and first number of the ID
    cout << "Enter your name: ";
    getline(cin, name);
    cout << "Enter the first digit from your ID: ";
    getline(cin, input);

    //Converting the string to an integer
    stringstream myStream(input);
    int digit = std::stoi(input);

    //declaring and int for the length of the loop
    int endloop = name.length();
    for (int x = 0; x < endloop; x = x + digit)
    {
    //Creating substring with the length equal to the number of the id.
    string name2 = name.substr(x, digit);
    std::cout << name2 << endl;

    }
    getline(cin, input);
    return 0;
}
```

OUTPUT

```
Enter your name: Venelin Dimitrov
Enter the first digit from your ID: 1
V
e
n
e
l
i
n

D
i
m
i
t
r
o
v
```

JAVA

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
public class Main {
    public static void main(String[] args) {
        String name;
        int parts;
        BufferedReader reader = new BufferedReader(new InputStreamReader(
                System.in));
        System.out.println("Enter name: ");
        name = reader.readLine();
        System.out.println("Enter student id: ");
    parts = Integer.parseInt(reader.readLine());
    List<String> strings = new ArrayList<String>();
    int i = 0;
    for (; i < name.length(); i += parts) {
        if (i + parts >= name.length()) {
            break;
        }
        strings.add(name.substring(i, i + parts));
    }
    String last = name.substring(i, name.length());
    if (null != last && !last.isEmpty()) {
    strings.add(last);
    }
    System.out.println(strings);
    }
}
```

OUTPUT

```
Enter name: Venelin Dimitrov

Enter parts:
1
[V, e, n, e, l, i, n,  , D, i, m, i, t, r, o, v]
```

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

a) How made Portfolio1 directory read/write/executable only for you and your group. That is, not for others. Show evidence of this with ls command.

mkdir Portfolio1
chmod ug+rwx Portfolio1

```
dimitrl3@hvs-its-lnx01:~$ mkdir Portfolio1
dimitrl3@hvs-its-lnx01:~$ chmod ug+rwx Portfolio1
dimitrl3@hvs-its-lnx01:~$ ls -l
total 8
drwxrwxr-x 2 dimitrl3 domain users 4096 Feb 16 14:58 Portfolio1
```

b) How downloaded the script http://www.centerkey.com/tree/tree.sh to your home directory using wget and make it executable.

wget http://www.centerkey.com/tree/tree.sh
chmod ug+rwx tree.sh

```
dimitri13@hvs-its-lnx01:~$ wget http://www.centerkey.com/tree/tree.sh
--2017-02-16 15:02:17--  http://www.centerkey.com/tree/tree.sh
Resolving www.centerkey.com (www.centerkey.com)... 66.185.31.194
Connecting to www.centerkey.com (www.centerkey.com)|66.185.31.194|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://centerkey.com/tree/tree.sh [following]
--2017-02-16 15:02:17--  http://centerkey.com/tree/tree.sh
Resolving centerkey.com (centerkey.com)... 66.185.31.194
Reusing existing connection to www.centerkey.com:80.
HTTP request sent, awaiting response... 200 OK
Length: 2107 (2.1K) [text/plain]
Saving to: 'tree.sh'

tree.sh                                             100%[=======================================

2017-02-16 15:02:17 (161 MB/s) - 'tree.sh' saved [2107/2107]

dimitri13@hvs-its-lnx01:~$ chmod ug+rwx tree.sh
dimitri13@hvs-its-lnx01:~$ ls -l
total 8
drwxrwxr-x 2 dimitri13 domain users 4096 Feb 16 14:58 Portfolio1
-rwxrwxr-- 1 dimitri13 domain users 2107 Aug  6  2015 tree.sh
dimitri13@hvs-its-lnx01:~$
```

c) Making Directories

- How created a 207se directory in your Portfolio1 directory.

  cd Portfolio1
  mkdir 207se

```
dimitri13@hvs-its-lnx01:~$ cd Portfolio1
dimitri13@hvs-its-lnx01:~/Portfolio1$ mkdir 207se
dimitri13@hvs-its-lnx01:~/Portfolio1$ ls -l
total 4
drwxr-xr-x 2 dimitri13 domain users 4096 Feb 16 15:06 207se
dimitri13@hvs-its-lnx01:~/Portfolio1$
```

- How created numbered directories for the labs. i.e. lab1 and lab2 etc.

  cd 207se
  mkdir lab1
  mkdir lab2

```
dimitri13@hvs-its-lnx01:~/Portfolio1$ cd 207se
dimitri13@hvs-its-lnx01:~/Portfolio1/207se$ mkdir lab1
dimitri13@hvs-its-lnx01:~/Portfolio1/207se$ mkdir lab2
dimitri13@hvs-its-lnx01:~/Portfolio1/207se$ ls -l
total 8
drwxr-xr-x 2 dimitri13 domain users 4096 Feb 16 15:07 lab1
drwxr-xr-x 2 dimitri13 domain users 4096 Feb 16 15:07 lab2
dimitri13@hvs-its-lnx01:~/Portfolio1/207se$
```

- Evidence of transferring lab1 activity into appropriate directory

  mv tree.sh Portfolio/207se/lab1

```
dimitri13@hvs-its-lnx01:~$ ls
Portfolio1  tree.sh
dimitri13@hvs-its-lnx01:~$ mv tree.sh /Portfolio1/
mv: cannot create regular file '/Portfolio1/': Not a directory
dimitri13@hvs-its-lnx01:~$ mv tree.sh Portfolio1/
dimitri13@hvs-its-lnx01:~$ ls
Portfolio1
dimitri13@hvs-its-lnx01:~$ cd..
cd..: command not found
dimitri13@hvs-its-lnx01:~$ cd Portfolio1/
dimitri13@hvs-its-lnx01:~/Portfolio1$ ls
207se  tree.sh
dimitri13@hvs-its-lnx01:~/Portfolio1$ mv tree.sh 207se/lab1
dimitri13@hvs-its-lnx01:~/Portfolio1$ cd 207se
dimitri13@hvs-its-lnx01:~/Portfolio1/207se$ cd lab1
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ ls
tree.sh
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$
```

- Evidence of make directory activities using tree.sh

~/Portfolio1/207se$ cd ~

```
dimitri13@hvs-its-lnx01:~$ ./tree.sh

/home/207SE/dimitri13
   .
   |-Portfolio1
   |---207se
   |-----lab1
   |-----lab2

dimitri13@hvs-its-lnx01:~$
```

d) Display todays date and using the cal command show the month that you were born.

```
It's my Birthday
dimitri13@hvs-its-lnx01:/$ cal
   February 2017
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

dimitri13@hvs-its-lnx01:/$
```

```
dimitri13@hvs-its-lnx01:/$ cal 02 1989
   February 1989
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
```

e) Move into the lab1 directory and use the appropriate command to show the current directory

cd Portfolio1/207se/lab1
pwd

```
dimitri13@hvs-its-lnx01:~$ cd Portfolio1/207se/lab1
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ pwd
/home/207SE/dimitri13/Portfolio1/207se/lab1
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$
```

f) What is talk, write and wall are for and which ones work

The write command is used to exchange private messages with other people connected to the same server. Just by simply typing "write (username)". Sadly its not working in this server and has been disabled by admin.

```
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ write daviesz2 hello
-bash: /usr/bin/write: Permission denied
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ write daviesz2 os-207se
-bash: /usr/bin/write: Permission denied
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ write
-bash: /usr/bin/write: Permission denied
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$
```

The command talk copies lines from your terminal to that of another user specified. The other user sends a request to connect to his machine and after that the messages can be exchanged. Also not working on our servers.

```
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ talk
The program 'talk' can be found in the following packages:
 * inetutils-talk
 * talk
 * ytalk
Ask your administrator to install one of them
```

The command wall is the other one disabled which we cannot use. It is used to broadcast a message to all of the users and machines connected to the server by just typing "wall (message)"

```
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ wall
-bash: /usr/bin/wall: Permission denied
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$
```

g) What command prevents the effects of those three commands from interrupting you?

The command which disables and enables the messaging through servers is "mesg (y or n)" y= yes, n=no

```
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ mesg y
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ mesg
is y
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ mesg n
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab1$ mesg
is n
```

h) The song in song.txt.

- Using wc the number of words and lines in the file.

  ~/Portfolio1/207se/lab2
  wc song

```
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ wc song
  32  247 1120 song
dimitri13@hvs-its-lnx01:~/Portfolio1/207se/lab2$
```

- Using grep to get the lines containing "and" and the number of the lines contain "and" in the document

  Grep –n and song

```
dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ grep -n and song
5:All to me come and dance to the
6:Club and then I said some too
12:I need and more you know I am so happy
23:Your attention and give your all baby
27:Came and took my hand I knew that if I
29:Into my life and you know it turns
```

- Use cat to show the contents of the file.

  cat song

```
dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ cat song
LOVE GIMME YOUR ATTENTION AND

Little voice inside me say your name I want
So lets go why dont you take it
All to me come and dance to the
Club and then I said some too
But to be your wife, I'll
Be right here she's only around me when your
Man be the one, by two y'all
Know what you like cuz I only want
You around my way someday some times all I want
I need and more you know I am so happy
That I'm putting on she keep it so hot
I don't need 'em anymore oh,
I love you boy let me show you you're
The air close your eyes are telling me
On a slow jam, baby how could I count
On me keep on shakin' it all
For me, you make me feel what's in
Front of me I'm coming to see
That your gone that your love for
Me what's the definition of love gimme
Your attention and give your all baby
Please say you care so why don't we give
Love a crazy little thing called love
Whoa ho yeah my best to wear a smile you
Came and took my hand I knew that if I
Got in my life sent from above came
Into my life and you know it turns
Me on whisper in my drop top back, relax
And let me know what I know the
Deal with you I found somebody new...
```

- Appropriate Linux command to see if the two files differ and how they differ.

  diff song song2

```
dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ pico
dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ diff song song2
14c14
< I don't need Venko anymore oh,
---
> I don't need 'em anymore oh,
18c18
< On me keep on dimitr13 shakin' it all
---
> On me keep on shakin' it all
25c25
< Love a crazy little Coventry thing called love
---
> Love a crazy little thing called love
32c32
< Deal with you I found somebody new...
---
> Deal with you I found somebody new.
```

- Use sort to sort the file and redirect the output to a new file called song2.txt

  sort song > song2.txt

```
dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ cat song2

All to me come and dance to the
And let me know what I know the
Be right here she's only around me when your
But to be your wife, I'll
Came and took my hand I knew that if I
Club and then I said some too
Deal with you I found somebody new...
For me, you make me feel what's in
Front of me I'm coming to see
Got in my life sent from above came
I don't need Venko anymore oh,
I love you boy let me show you you're
I need and more you know I am so happy
Into my life and you know it turns
Know what you like cuz I only want
Little voice inside me say your name I want
Love a crazy little Coventry thing called love
LOVE GIMME YOUR ATTENTION AND
Man be the one, by two y'all
Me on whisper in my drop top back, relax
Me what's the definition of love gimme
On a slow jam, baby how could I count
On me keep on dimitr13 shakin' it all
Please say you care so why don't we give
So lets go why dont you take it
That I'm putting on she keep it so hot
That your gone that your love for
The air close your eyes are telling me
Whoa ho yeah my best to wear a smile you
You around my way someday some times all I want
Your attention and give your all baby
```

- Use sort and rev to reverse the sorted contents of song.txt and append the output to song2.txt

  dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ sort -r song > song3
  dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ cat song3 >> song2
  dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ cat song3

```
dimitr13@hvs-its-lnx01:~/Portfolio1/207se/lab2$ cat song3
Your attention and give your all baby
You around my way someday some times all I want
Whoa ho yeah my best to wear a smile you
The air close your eyes are telling me
That your gone that your love for
That I'm putting on she keep it so hot
So lets go why dont you take it
Please say you care so why don't we give
On me keep on dimitr13 shakin' it all
On a slow jam, baby how could I count
Me what's the definition of love gimme
Me on whisper in my drop top back, relax
Man be the one, by two y'all
LOVE GIMME YOUR ATTENTION AND
Love a crazy little Coventry thing called love
Little voice inside me say your name I want
Know what you like cuz I only want
Into my life and you know it turns
I need and more you know I am so happy
I love you boy let me show you you're
I don't need Venko anymore oh,
Got in my life sent from above came
Front of me I'm coming to see
For me, you make me feel what's in
Deal with you I found somebody new...
Club and then I said some too
Came and took my hand I knew that if I
But to be your wife, I'll
Be right here she's only around me when your
And let me know what I know the
All to me come and dance to the
```

- Total memory used and the total memory available

  free –m

```
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ free -m
              total        used        free      shared  buff/cache   available
Mem:          32167         416       31263          25         486       31340
Swap:          1021           0        1021
```

- Find out how you can display your username on the screen.

  whoami
  id –u –n

```
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ whoami
dimitri3
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ id -u -n
dimitri3
```

- List the processes that are running.
  ps –aux | less
  ps –aux | wc -l

```
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$  ps -aux | less

[1]+  Stopped                 ps -aux | less
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ top
top - 16:02:20 up 12:02, 20 users,  load average: 0.83, 0.70, 0.57
Tasks: 314 total,   2 running, 311 sleeping,   1 stopped,   0 zombie
%Cpu(s):  8.2 us,  0.4 sy,  0.0 ni, 91.2 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem : 32939192 total, 31978264 free,   462264 used,   498664 buff/cache
KiB Swap:  1046524 total,  1046524 free,        0 used. 32057720 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
33758 zabarr    20   0  140832   5704   4536 R  35.5  0.0   1:07.08 sshd
34763 zabarr    20   0  317268  46612  12132 S  33.6  0.1   0:56.34 bochs-bin
30685 root      20   0       0      0      0 S   0.3  0.0   0:01.80 kworker/u128:3
34704 salantap  20   0   35024   4720   3132 S   0.3  0.0   0:00.17 pico
34777 dimitri3  20   0   51080   4484   3592 R   0.3  0.0   0:00.51 top
    1 root      20   0   38240   6376   4032 S   0.0  0.0   0:14.46 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.04 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.44 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S   0.0  0.0   0:21.71 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S   0.0  0.0   0:00.08 migration/0
   10 root      rt   0       0      0      0 S   0.0  0.0   0:00.28 watchdog/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.36 watchdog/1
   12 root      rt   0       0      0      0 S   0.0  0.0   0:00.10 migration/1
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.26 ksoftirqd/1
   15 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
   16 root      rt   0       0      0      0 S   0.0  0.0   0:00.25 watchdog/2
   17 root      rt   0       0      0      0 S   0.0  0.0   0:00.09 migration/2
   18 root      20   0       0      0      0 S   0.0  0.0   0:00.56 ksoftirqd/2
```

```
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ ps aux | wc
    341    3960   28265
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ ps aux | wc -l
341
dimitri3@hvs-its-lnx01:~/Portfolio1/207se/lab2$ ps aux| wc -l
```

- What are the differences between the Linux commands less, more and most.

  “more” “most” and “less” do mostly the same functions.
  “more” is an older version which shows a huge pile of text which can’t be displayed on the screen so it scrolls down every time you press a button but it can’t scroll back up which was a problem for most users.

  “less” was written by a user which was fed up by “more” incapability to scroll backwards in a file

  “most” most is considered an upgrade of “less” and the better version of those two. It can visualise multiple text files at once. It can hold the information about long lines and scrolls even further so they can

be displayed so it also provides left and right scrolling as an included backwards and downwards. Also it has the opportunity to decompress gunzip files before reading them.

Lab Activity 4 Bootloader

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

In this activity the main task was to create an empty file and run it as a bootloader. The file had written commands to be run and compiled through nasm and pragma. Task was to create a file which will display my names, student id, date of birth, email, second year favourite module and also my age all of these on separate lines.

The outcome of this task is to learn and see how we can use a programming language to be capable to be used by microcontrollers or other programmable device to enhance or create new tasks after being self-programmable.

b) Boot pragma linux with bochs

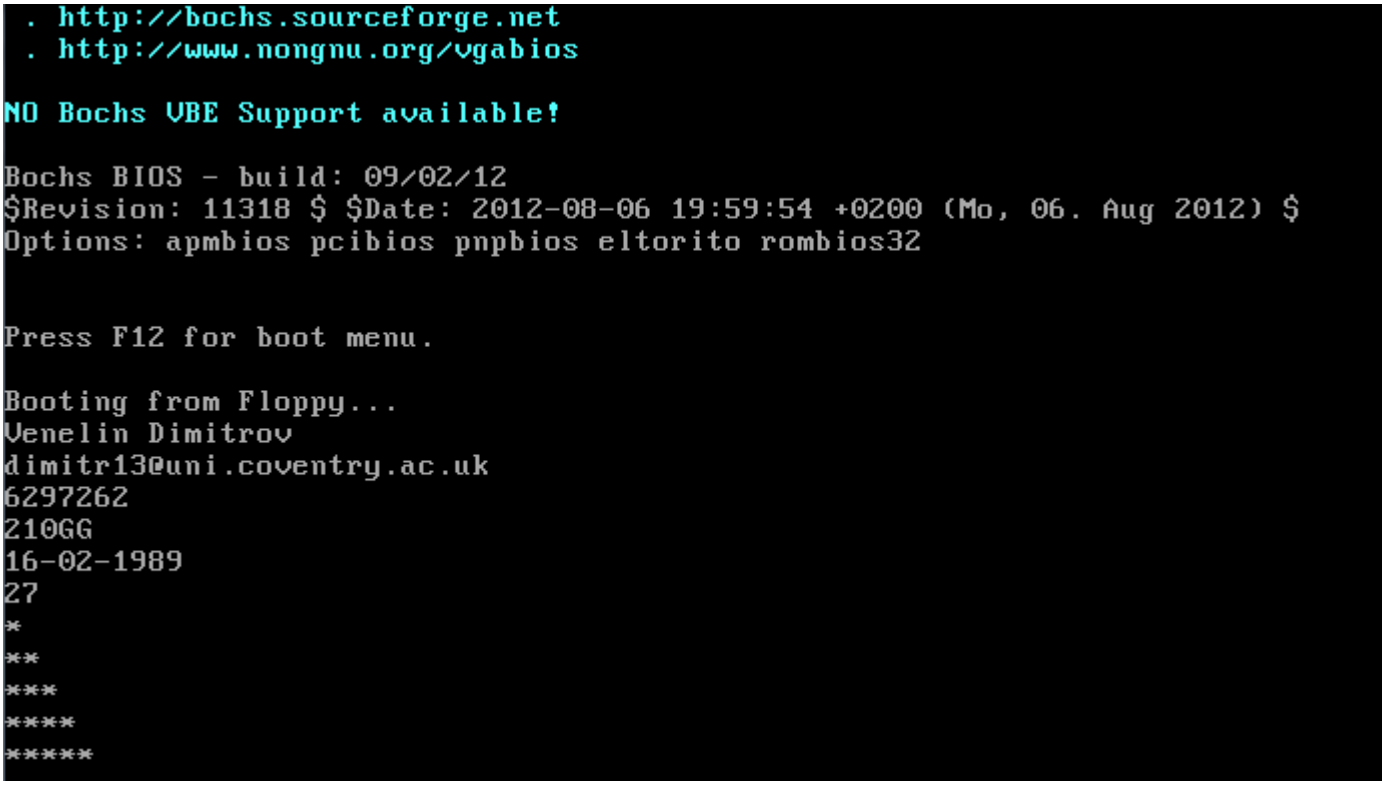
```
dimitr13@hvs-its-lnx01:~$ ls
Portfolio1  pragma-207.tgz  tree.sh
dimitr13@hvs-its-lnx01:~$ tar xvfz pragma-207.tgz
pragmalinux-img/
pragmalinux-img/README
pragmalinux-img/bootloader
pragmalinux-img/a.img
pragmalinux-img/bochsrc
pragmalinux-img/bochsrc1
pragmalinux-img/bochsout.txt
pragmalinux-img/TESTFORM.txt
pragmalinux-img/bootloader.asm
dimitr13@hvs-its-lnx01:~$ cd pragmalinux-img/
dimitr13@hvs-its-lnx01:~/pragmalinux-img$ nasm bootloader.asm
dimitr13@hvs-its-lnx01:~/pragmalinux-img$ dd if=bootloader bs=512 of=a,img
1+0 records in
1+0 records out
512 bytes copied, 0.0001253 s, 4.1 MB/s
dimitr13@hvs-its-lnx01:~/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 0x428b3f0
00000000000i[PLGIN] loaded plugin libbx_unmapped.so
00000000000i[     ] lt_dlhandle is 0x428c050
00000000000i[PLGIN] loaded plugin libbx_biosdev.so
00000000000i[     ] lt_dlhandle is 0x428c9d0
00000000000i[PLGIN] loaded plugin libbx_speaker.so
00000000000i[     ] lt_dlhandle is 0x428cd80
00000000000i[PLGIN] loaded plugin libbx_extfpuirq.so
00000000000i[     ] lt_dlhandle is 0x428e8e0
00000000000i[PLGIN] loaded plugin libbx_parallel.so
00000000000i[     ] lt_dlhandle is 0x4290590
00000000000i[PLGIN] loaded plugin libbx_serial.so
00000000000i[     ] lt_dlhandle is 0x4294170
00000000000i[PLGIN] loaded plugin libbx_gameport.so
00000000000i[     ] lt_dlhandle is 0x4294c10
00000000000i[PLGIN] loaded plugin libbx_iodebug.so
00000000000i[     ] reading configuration from bochsrc
00000000000e[     ] bochsrc:9: 'vga_update_interval' will be replaced by new 'vga: update_freq' option.
00000000000e[     ] bochsrc:10: 'i440fxsupport' will be replaced by new 'pci' option.
00000000000i[     ] lt_dlhandle is 0x42954d0
00000000000i[PLGIN] loaded plugin libbx_sdl.so
00000000000i[     ] installing sdl module as the Bochs GUI
00000000000i[     ] using log file bochsout.txt
Next at t=0
(0) [0x00000000fffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b         ; ea5be000f0
<bochs:1>
```

c) Make a bootloader that displays your student details and triangle

- Commented bootloader code to display your student details and triangle

```
[BITS 16]
[ORG 0x7C00]
top:
        ;; Add a 0 value into the Data-segment
        ;; This function cant be done directly
        mov ax,0x0000
        mov ds,ax
        ;; si is the location relative to the data segment of the
        ;; string/char to display
        mov si, Name
        call writeString ; See below
        mov si, Email
        call writeString
        mov si, id
        call writeString
        mov si, favModule
        call writeString
        mov si, DateofBirth
        call writeString
        mov si, Age
        call writeString
        mov si, star1
        call writeString
        mov si, star2
        call writestring
        mov si, star3
        call writestring
        mov si, omgstar
        call writestring
        mov si, starfinish
        call writestring
        jmp $ ; Spin
writeString:
        mov ah,0x0E ; Display a charachter when compiled
        mov bh,0x00
        mov bl,0x07
nextchar:
        Lodsb ; Loads [SI] into AL and increases SI by one
        ;; Effectively gets info on the string through AL
        cmp al,0 ; End of the string?
        jz done
        int 0x10 ; BIOS interrupt
        jmp nextchar
done:
        ret
        ;; The information needed to run all the commands
        Name db 'Venelin Dimitrov',13,10,0 ; Null-terminated
        Email db 'dimitr13@uni.coventry.ac.uk',13,10,0
        id db '6297262',13,10,0
        favModule db '210GG',13,10,0
        DateofBirth db '16-02-1989',13,10,0
        Age db '27',13,10,0
        star1 db '*',13,10,0
        star2 db '**',13,10,0
        star3 db '***',13,10,0
        omgstar db '****',13,10,0
        starfinish db '*****',13,10,0
        times 510-($-$$) db 0
        dw 0xAA55
```

- Output from Bochs showing student details and triangle

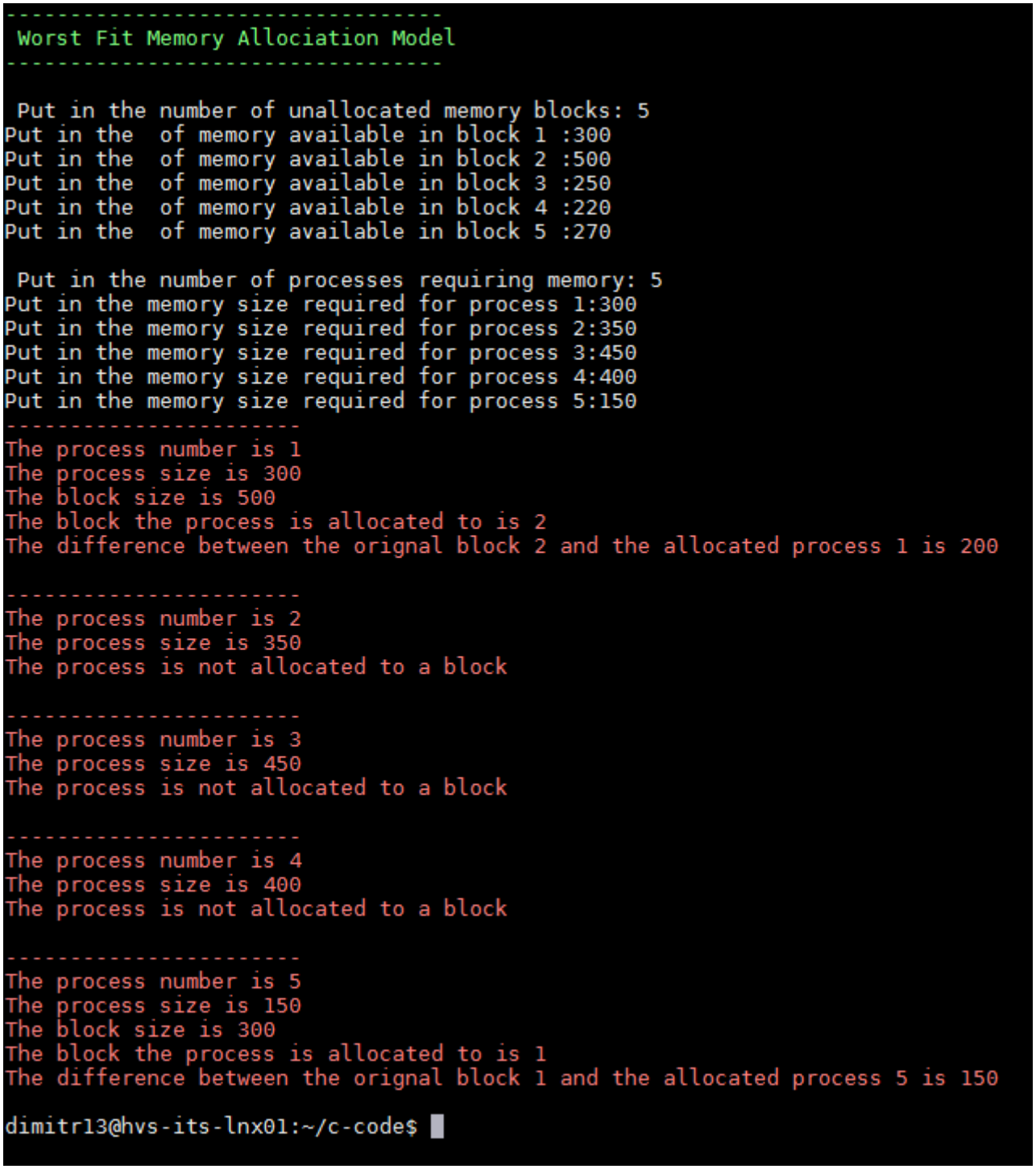

. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios
NO Bochs VBE Support available!
Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32
Press F12 for boot menu.
Booting from Floppy...
Venelin Dimitrov
dimitr13@uni.coventry.ac.uk
6297262
210GG
16-02-1989
27
*
**
***
****
*****

Lab Activity 6 Outside the processor

a) Memory Allocation Activities

```
---------------------------------------
 First-Fit Memory Allocation Approach
---------------------------------------

 Put in the number of unallociated memory blocks available: 5
Put in the unallociated memory available in block 1 :300
Put in the unallociated memory available in block 2 :500
Put in the unallociated memory available in block 3 :250
Put in the unallociated memory available in block 4 :220
Put in the unallociated memory available in block 5 :270

 Put in the number of processes requiring memory: 5
Put in the memory size of process  1:300
Put in the memory size of process  2:350
Put in the memory size of process  3:450
Put in the memory size of process  4:400
Put in the memory size of process  5:150
---------------------------
The process number is 1
The process size is 300
The block size is 300
The block the process is allocated to is 1
The difference between the orignal block 1 and the allociated process 1 is 0

---------------------------
The process number is 2
The process size is 350
The block size is 500
The block the process is allocated to is 2
The difference between the orignal block 2 and the allociated process 2 is 150

---------------------------
The process number is 3
The process size is 450
The process is not allocated to a block

---------------------------
The process number is 4
The process size is 400
The process is not allocated to a block

---------------------------
The process number is 5
The process size is 150
The block size is 250
The block the process is allocated to is 3
The difference between the orignal block 3 and the allociated process 5 is 100

dimitr13@hvs-its-lnx01:~/c-code$
```

b)


Worst Fit Memory Allociation Model
Put in the number of unallocated memory blocks: 5
Put in the of memory available in block 1 :300
Put in the of memory available in block 2 :500
Put in the of memory available in block 3 :250
Put in the of memory available in block 4 :220
Put in the of memory available in block 5 :270
Put in the number of processes requiring memory: 5
Put in the memory size required for process 1:300
Put in the memory size required for process 2:350
Put in the memory size required for process 3:450
Put in the memory size required for process 4:400
Put in the memory size required for process 5:150
The process number is 1
The process size is 300
The block size is 500
The block the process is allocated to is 2
The difference between the orignal block 2 and the allocated process 1 is 200
The process number is 2
The process size is 350
The process is not allocated to a block
The process number is 3
The process size is 450
The process is not allocated to a block
The process number is 4
The process size is 400
The process is not allocated to a block
The process number is 5
The process size is 150
The block size is 300
The block the process is allocated to is 1
The difference between the orignal block 1 and the allocated process 5 is 150
dimitri3@hvs-its-lnx01:~/c-code$

```
-------------------------------------
 Best-Fit Memory Allocation Approach
-------------------------------------

 Put in the number of unallocated blocks of memory available: 5
Put in the size of the unallocated memory available in block 1 :300
Put in the size of the unallocated memory available in block 2 :500
Put in the size of the unallocated memory available in block 3 :250
Put in the size of the unallocated memory available in block 4 :220
Put in the size of the unallocated memory available in block 5 :270

 Put in the number of processes requiring memory: 5
Put in the size of the memory required for process  1:300
Put in the size of the memory required for process  2:500
Put in the size of the memory required for process  3:250
Put in the size of the memory required for process  4:220
Put in the size of the memory required for process  5:270
--------------------------
The process number is 1
The process size is 300
The block size is 300
The block the process is allocated to is 1
The difference between the orignal block 1 and the allocated process 1 is 0

--------------------------
The process number is 2
The process size is 500
The block size is 500
The block the process is allocated to is 2
The difference between the orignal block 2 and the allocated process 2 is 0

--------------------------
The process number is 3
The process size is 250
The block size is 250
The block the process is allocated to is 3
The difference between the orignal block 3 and the allocated process 3 is 0

--------------------------
The process number is 4
The process size is 220
The block size is 220
The block the process is allocated to is 4
The difference between the orignal block 4 and the allocated process 4 is 0

--------------------------
The process number is 5
The process size is 270
The block size is 270
The block the process is allocated to is 5
The difference between the orignal block 5 and the allocated process 5 is 0

dimitr13@hvs-its-lnx01:~/c-code$
```

c)
Which approaches allocates all of the processes and with the least fragmentation.

Best fit

M1 (300)	M2 (500)	M3 (250)	M4 (220)	M5 (270)

Worst fit

M1 (300)	M2 (500)	M3 (250)	M4 (220)	M5 (270)

The process which fragments the least of the two is the **best fit**. It allocates all the memory units in the processes and does least task for fragmentation.

Paging Activities

FIRST IN FIRST OUT

	4	2	7	7	5	6	3	9	3	2	2
Page Entry 0	4	4	4	4	5	5	5	9	9	9	9
Page Entry 1	-	2	2	2	2	6	6	6	6	2	2
Page Entry 2	-	-	7	7	7	7	3	3	3	3	3
Page Fault	1	2	3	3	4	5	6	7	7	8	8

	4	2	7	7	5	6	3	9	3	2	2
Page Entry 0	4	4	4	4	4	6	6	6	6	6	6
Page Entry 1	-	2	2	2	2	2	3	3	3	3	3
Page Entry 2	-	7	7	7	7	7	7	9	9	9	9
Page Entry 3	-	-	-	-	5	5	5	5	5	2	2
Page Fault	1	2	3	3	4	5	6	7	7	8	8

The results for both of them are not that much different from each other after the algorithm has passed from the first half of the table it changes drastically as all the values for each entry of the steps have a different result because of the existing 4th page entry on the second table. The value just pushes into the next step and thus resulting in a different sequence. Page fault score stayed the same and completely identical to both tables.

Repeat the above process for the random page allocation approach.
RANDOM PAGE ALLOCATION

	4	2	7	7	5	6	3	9	3	2	2
Page Entry 0	4	4	4	4	4	6	3	3	3	2	2
Page Entry 1	-	2	2	2	5	5	5	5	5	5	5
Page Entry 2	-	-	7	7	7	7	7	9	9	9	9
Page Fault	1	2	3	3	4	5	6	7	7	8	8

	4	2	7	7	5	6	3	9	3	2	2
Page Entry 0	4	4	4	4	4	4	4	4	4	4	4
Page Entry 1	-	2	2	2	2	2	3	9	3	2	2
Page Entry 2	-	-	7	7	7	6	6	6	6	6	6
Page Entry 3	-	-	-	5	5	5	5	5	5	5	5
Page Fault	1	2	3	3	4	5	6	7	8	9	9

The results for both random page allocation with 3 and 4 page entries resulted in a completely different from each other tables. The values given by the user were identical but as there was a 4 page values for the second table the result ended with some interesting results. Page fault score did change but only after the 9th step in the sequence.

Lab Activity 7 Buffer

a) Brief description of the Buffer Activity

First of we have to edit and comment different lines of the buffer.c code so we show that we understand what is happening in the code. For the first editing part we have to include a print error messages in case something goes wrong and we have information about it. Possible errors could be that we don't have the two inserts, not a valid input or problem with copying the file.

The code below is just creating a buffer file which collects information from the input.txt file and doubles it in another output file different from the original one. Changing the size of the buffer at the beginning, changes how many bytes of characters are filled to the buffer at one time. The bigger the size of the buffer the more quicker it can get filled.

b) Commented Buffer.c code

```
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>

//defining the size of the buffer and its output mode
#define BUF_SIZE 500
#define OUTPUT_MODE 0700

int main(int argc, char *argv[])
{
  int in_fd, out_fd;            //input and output values defined
  int rd_size = 1, wr_size;
  char buf[BUF_SIZE];    //defining a variable to be stored corresponding to the size of the buffer
//if statement to check if all 3 of the arguments are present
  if (argc != 3)
    exit(1);
//checks for a valid input file
  in_fd = open(argv[1], O_RDONLY);
  if (in_fd < 0)
    exit(2);
//checks if the file is copied
  out_fd = creat(argv[2], OUTPUT_MODE);
  if (out_fd < 0)
    exit(3);

  while (rd_size > 0) {

    rd_size = read(in_fd, buf, BUF_SIZE);
    if (rd_size <0)
      exit(4);

    wr_size = write(out_fd, buf, rd_size);
    if (wr_size<=0){
              close(in_fd);
//closes all opened files
  close(out_fd);
exit(5);
}
  }
}
```

c) Update the code to so that it prints if an error has occurred or if a file is successfully created with the content of the review in it.

After running code what is in hamlet.txt

```
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>



//defining the size of the buffer and its output mode
#define BUF_SIZE 500
#define OUTPUT_MODE 0700

int main(int argc, char *argv[])
{
  int in_fd, out_fd;  //input and output values defined
  int rd_size = 1, wr_size;
  char buf[BUF_SIZE];  //defining a variable to be stored corresponding to the size of the buffer
  //if statement to check if all 3 of the arguments are present
  if (argc != 3)  {
    printf("You 2 or more inputs");
    exit(1);
  }
  //checks for a valid input file
  in_fd = open(argv[1], O_RDONLY);
  if (in_fd < 0)  {
    print("Sorry you have not selected an input file");
    exit(2);
  }
  //checks if the file is copied
  out_fd = creat(argv[2], OUTPUT_MODE);
  if (out_fd < 0) {
    printf("Sorry, file was not copied");
    exit(3);
  }
  while (rd_size > 0) {

    rd_size = read(in_fd, buf, BUF_SIZE);
    if (rd_size <0)
      exit(4);

    wr_size = write(out_fd, buf, rd_size);
    if (wr_size<=0){
                close(in_fd);
  //closes all opened files
  close(out_fd);
exit(5);
}
  }
}
```

After the code has been compiled and runed in the terminal we receive two files. One of hamlet.txt and one of the latest production of Hamlet by Shakespeare

d) Updated buffer.c code to show how many character are read to buffer, how many character read at a time into the buffer, how many words in the document and how many times the buffer is filled

```
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>



//defining the size of the buffer and its output mode
#define BUF_SIZE 500
#define OUTPUT_MODE 0700

int main(int argc, char *argv[])
{
  int characters = 0;
  int words = 0;
  int i = 0;
  int in_fd, out_fd;  //input and output values defined
  int rd_size = 1, wr_size;
  char buf[BUF_SIZE];  //defining a variable to be stored corresponding to the size of the buffer
  //if statement to check if all 3 of the arguments are present
  if (argc != 3)  {
    printf("You need two or more inputs");
    exit(1);
  }
  //checks for a valid input file
  in_fd = open(argv[1], O_RDONLY);
  if (in_fd < 0)  {
    printf("Sorry you have not selected an input file");
    exit(2);
  }
  //checks if the file is copied
  out_fd = creat(argv[2], OUTPUT_MODE);
  if (out_fd < 0) {
    printf("Sorry, file was not copied");
    exit(3);
  }
  while (rd_size > 0) {

    rd_size = read(in_fd, buff, BUF_SIZE);
    if (rd_size <0) {
      exit(4);}
    else {
characters +=rd_size;
printf("%d\n",rd_size);

    }
    for (int i=0; i<rd_size; i++){
    if (buf[i]==' ')
        words++;
    {
    if (rd_size > 0){
    printf("Characters: %d\n", characters);
    printf("Words: %d\n", words);
    }
    wr_size = write(out_fd, buff, rd_size);
```

```
  if (wr_size<=0){
            close(in_fd);
 //closes all opened files
 close(out_fd);
 printf("File Succesfully Copied");}
             }
         }
     }
 }
```

```
Characters: 3399
Words: 570
Characters: 3399
Words: 570
Characters: 3399
Words: 570
Characters: 3399
Words: 570
Characters: 3399
Words: 570
Characters: 3399
Words: 571
Characters: 3399
Words: 571
Characters: 3399
Words: 571
Characters: 3399
Words: 571
0
dimitr13@hvs-its-lnx01:~/buffer$
```

e) Impact of changing buffer size

```
dimitr13@hvs-its-lnx01:~/buffer$ ./buffer review.txt hamlet.txt
2000
Characters: 2000
Words: 338
1399
Characters: 3399
Words: 571
0
File Succesfully Copied dimitr13@hvs-its-lnx01:~/buffer$
```

If we double the size of the buffer from 500 to 2000 we receive the following output. There is less buffer refils this way

f) Updated buffer.c code to compare if two files are the same

```c
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>



//defining the size of the buffer and its output mode
#define BUF_SIZE 500
#define OUTPUT_MODE 0700
#define BUF_SIZE 500
#define OUTPUT_MODE 0700
#define SHELLSCRIPT "\
#bin/bash \n\
echo \$(diff hamlet.txt review_observer.txt)\n\




int main(int argc, char *argv[])
{
  int counter_characters = 0;
  int counter_words = 0;
  int i = 0;
  int in_fd, out_fd;  //input and output values defined
  int rd_size = 1, wr_size;
  char buf[BUF_SIZE];  //defining a variable to be stored corresponding to the size of the buffer
  //if statement to check if all 3 of the arguments are present


  if (argc != 3)
  {
       printf("You need two or more inputs ");
       exit(1);
  }



  //checks for a valid input file
  in_fd = open(argv[1], O_RDONLY);
  if (in_fd < 0)
  {
       printf("Sorry you have not selected an input file \n ");
       exit(2);
  }


  //checks if the file is copied
  out_fd = creat(argv[2], OUTPUT_MODE);
  if (out_fd < 0)
  {
```

```
        printf("Sorry, file was not copied ");
        exit(3);
    }

    while (rd_size > 0)
    {
        rd_size = read(in_fd, buf, BUF_SIZE);
        if (rd_size <0)
    {
                printf("reading from file failed\n"); //shows if there is a problem wit this statement
                exit(4);
    }
        else
    {
            counter_characters += rd_size;
            printf("%d\n",rd_size); //reads from buffer everytime its has been filled

    }
      for (i=0; i<rd_size; i++)
    {
      if (buf[i]==' ')
           counter_words++;
      }
      if (rd_size > 0){
          printf("Characters: %d\n", counter_characters);
          printf("Words: %d\n", counter_words);
      }
      wr_size = write(out_fd, buf, rd_size);
      if (wr_size<=0){
                    close(in_fd);
    //closes all opened files
            close(out_fd);
            system(SHELLSCRIPT);
            printf("File Succesfully Copied ");
    exit(5);

}
}
}
```

- Comparison of review.txt and hamlet.txt

```
dimitr13@hvs-its-lnx01:~/buffer$ gcc -o buffer buffer.c
dimitr13@hvs-its-lnx01:~/buffer$ ./buffer review.txt hamlet.txt
500
Characters: 500
Words: 78
500
Characters: 1000
Words: 167
500
Characters: 1500
Words: 250
500
Characters: 2000
Words: 338
500
Characters: 2500
Words: 420
500
Characters: 3000
Words: 501
399
Characters: 3399
Words: 571
0
File Succesfully Copied dimitr13@hvs-its-lnx01:~/buffer$
```

There is no output from the program because the files are identical

- Comparison of hamlet.txt and review_observer.txt

```
Character:  3399
Words:  571
0
1,8c1,23 < So, after all the hype, the Hamlet. Tonight we saw the first preview
performance of Benedict Cumberbatch▒s interpretation of the great Dane, at Londo
n▒s Barbican theatre. Absent from the audience were the shrieking fans, snatchin
 < Judging by the first batch of spectators, your typical Cumberbitch is a polit
e, plumpish lady in her mid thirties, hailing from Northern Europe. She might be
 more at home at Sherlock-con, the now annual convention for fans of the BBC One
 show that has turned Cumberbatch into a global star, but she knows too to turn
her mobile phone off during a play, and she may well be too sensible to fork out
 < The audience, in summary, was as impeccably behaved as you▒d expect from a us
ual London theatre crowd, granting Cumberbatch only one mid-scene ovation, when
he did a hilarious impression of a toy soldier. They stood up at the end, but on
 < Director Lyndsey Turner and designer Es Devlin have created a lavish, epic Ha
mlet for the Barbican▒s vast stage. Not perhaps since it held the barricades of
revolution for the first performances of Les Miserables in the Eighties has this
 < Cumberbatch and his fellow cast members have a palatial hall in which to play
 out their tragedy. Panelled walls painted a rich dark turquoise reach to an eno
rmous chandelier and a grand staircase descends stage right; the whole is remini
scent of a John Singer Sargent painting, with costumes to match. Oh and there▒s
 < Before this tableau is unveiled however, we meet Hamlet on his own. Cumberbat
ch opens the play bent over boxes of old possessions with a record player crackl
ing in the background. The first line in this production is the most famous in t
he play, ▒To Be or Not to Be▒, not the more prosaic ▒Who▒s there▒, said by a sol
dier, that Shakespeare bequeathed us. Indeed, Cumberbatch delivers the whole spe
ech there and then in the opening minutes, the first of several tweaks to the te
xt where Turner has dropped the soliloquies into new places throughout the play.
 Although Shakespeare novices may not realise this: the devastating keynotes of
 < A first preview is not the place to offer analysis of the performances. This
play has three weeks to run in before it opens to the critics. Cumberbatch▒s int
erpretation of the title role is going to shift and develop. He already commands
 and surprises, there are laughs and shocks, and with a cast that includes the a
lways moving and intelligent Ciaran Hinds as Hamlet▒s murdering uncle Claudius,
 < Our theatre critic Dominic Cavendish wrote in these pages yesterday that to p
lay Hamlet, such a protean character, is to take on the challenge of life itself
. For Benedict Cumberbatch, still only 39, life in recent months has meant beari
ng the pressure of leading the fastest selling London theatre show since records
 began. That is a challenge indeed and, with the first of some eighty performanc
es he will give under his belt, one he has only just embarked upon. This both po
pular and hugely talented actor has staked out the greatest part in drama, it wi
ll be fascinating to see how he masters it. \ No newline at end of file --- > Be
nedict Cumberbatch playing Hamlet is so obvious, it feels like it happened alrea
dy. One of the most talented and popular actors of this generation doing THE rol
e in theatre. Every painter paints himself and eventually every actor plays Haml
 > But more than that, the character seems like such a natural fit: the intellec
tual Dane played by the actor who has made a career out of geniuses. Hamlet has
always been a geek posterboy ▒ awkward but passionate, intelligent but crippled
by self-doubt ▒ and Cumberbatch is the go-to example of the power of modern fand
oms. In the media, the online hordes of Cumberfans have become lazy shorthand fo
r nerds and obsessives, so much so that some predicted the opening night would b
 > (Of course this was utter nonsense ▒ the audience were receptive and respectf
 > Nevertheless, it is tempting to view Cumberbatch▒s Hamlet as representing the
```

We receive an output from both files because they are different content from each other and the text has merged.

Lab Activity 8 Cache Buffer

a) Brief Description of Cache Buffer Activity

The first task of this lab is to implement the red_byte function into a code. It should read the bytes created and transfer them to the cache buffer. When all the bytes are transferred the buffer should refill. For the next task we need to edit the code, so we can view when the buffer is refiled and when a byte of data has been read.

This will involve using external ints, so we can share them between the three files created. Last change to the code will be to make it print the amount of bytes read and the number of times the buffer has been refiled.

b) Commented implementation of the cr_read_byte function

```
char cr_read_byte(cr_file* f){
if (f->usedbuffer == f->bufferlength){
refill(f); //We call refill function when the value of the usedbuffer is equal to the length of the buffer
}
return f->buffer[f->usedbuffer++];//We return the letter from the buffer which is on the place of usedbuffer
return EOF;
```

```
dimitr13@hvs-its-lnx01:~/cache-example$ ./cache example
Geographically, Iran is located in West Asia and borders the Caspian Sea, Persian Gulf, and Gulf of Oman. Its mountains have helped
ountry for several centuries. The mountains enclose several broad basins, or plateaus, on which major agricultural and urban settleme
 and railroads were constructed through the mountains to connect the population centers, these basins tended to be relatively isolate
n basin, and there were complex economic relationships between the town and the hundreds of villages that surrounded it. In the highe
rganized groups practiced transhumance, moving with their herds of sheep and goats between traditionally established summer and winte
 and historically transportation was by means of caravans that followed routes traversing gaps and passes in the mountains. The mount
spian Sea. With an area of 1,648,000 square kilometres (636,000 sq mi), Iran ranks eighteenth in size among the countries of the worl
tates: Armenia, Azerbaijan, and Turkmenistan. These borders extend for more than 2,000 kilometres (1,200 mi), including nearly 650 ki
aspian Sea. Iran's western borders are with Turkey in the north and Iraq in the south, terminating at the Arvand Rud. The Persian Gul
s (1,100 mi) southern border. To the east lie Afghanistan on the north and Pakistan on the far south. Iran's diagonal distance from A
e in the southeast is approximately 2,333 kilometres (1,450 mi). The history of Iran, commonly also known as Persia in the Western wo
o to an extent known as Greater Iran, comprising the area from Anatolia, the Bosphorus, and Egypt in the west to the borders of Ancie
s and the Eurasian Steppe in the north to the Persian Gulf and the Gulf of Oman in the south.  Iran is home to one of the world's olc
n settlements dating back to 4000 BC.[1] The southwestern and western part of the Iranian Plateau participated in the traditional Anc
er with various other peoples, such as the Kassites, Mannaeans, and Gutians. Georg Wilhelm Friedrich Hegel names the Persians as the
on and empire in 625 BC.The Achaemenid Empire (550▒330 BC), founded by Cyrus the Great, was the first of the Persian empires to rule
hing three continents, from their seat of power in Persis (Persepolis). It was the largest empire yet seen and the first world empire
of history connect over 40% of the global population, accounting for approximately 49.4 million of the world▒s 112.4 million people
an and Sasanian Empires, who successively governed Iran for almost 1000 years and made Iran once again as a leading power in the worl
r, the Byzantine Empire. The Persian Empire proper begins in the Iron Age, following the influx of Iranian peoples. Iranian people ga
 Empires of classical antiquity. Once a major empire of superpower proportions, having conquered far and wide, Iran has endured invas
 has continually reasserted its national identity throughout the centuries and has developed as a distinct political and cultural ent
ian Empire and was a turning point in Iranian history. Islamization of Iran took place during the eighth to tenth centuries and led
many of its dependancie. However, the achievements of the previous Persian civilizations were not lost, but were to a great extent ab
once again reunified as an independent state in 1501 by the Safavid dynasty, which converted Iran to Shia Islam[8] as the official re
rning points in the history of Islam. Functioning again as a leading power, this time amongst the neighboring Ottoman Empire, their a
n emperor almost without interruption from 1501 until the 1979 Iranian Revolution, when Iran officially became an Islamic republic or
n century Iran lost many of its territories in the Caucasus (which it had been ruling intermittently encompassed for millennia), comp
 Armenia, to its rapidly expanding and emerged neighboring rival the Russian Empire, following the Russo-Persian Wars between 1804▒1

dimitr13@hvs-its-lnx01:~/cache-example$
```

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

```
char cr_read_byte(cr_file* f){
        if (f->usedbuffer == f->bufferlength){
                refill(f); //We call refill function when the value of the usedbuffer is equal to the length of the buffer
                printf("\n Buffer has been refilled \n"); //Shows us when the buffer has been refiled
}       int len=f->bufferlength; //int var which holds the length for the buffer
        printf("\n We have %d bytes transfered to buffer \n", len); //prints after every byte has been read
        return f->buffer[f->usedbuffer++];//We return the letter from the buffer which is on the place of usedbuffer
        return EOF;
}
```

```
dimitr13@hvs-its-lnx01:~/cache-example$ ./cache_example

We have 20 bytes to buffer
O
We have 20 bytes to buffer
n
We have 20 bytes to buffer

We have 20 bytes to buffer
c
We have 20 bytes to buffer
a
We have 20 bytes to buffer
m
We have 20 bytes to buffer
p
We have 20 bytes to buffer
u
We have 20 bytes to buffer
s
We have 20 bytes to buffer
,
We have 20 bytes to buffer

We have 20 bytes to buffer
t
We have 20 bytes to buffer
h
We have 20 bytes to buffer
e
We have 20 bytes to buffer
y
We have 20 bytes to buffer

We have 20 bytes to buffer
c
We have 20 bytes to buffer
a
We have 20 bytes to buffer
l
We have 20 bytes to buffer
l
 Buffer has been refilled

We have 20 bytes to buffer
```

d) Commented updated code showing to show how many bytes were read in total, and how many times the buffer was refilled

```
  GNU nano 2.5.3            File: cache_reader.c                    Modified

#include "cache_reader.h"
#include "cache_reader.c"


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

int bytes = 0; //var for bytes
int refills=0; //var for refills
int refill(cr_file* buff)
{
  //Refills a buffer
  //Only works when completely used buffer
  if(buff->usedbuffer!=buff->bufferlength)
```

```
    return 0;
  else{
    refills++; // increasing refills with each loop
    buff->usedbuffer=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;
    return len, refills;
  }

}

void cr_close(cr_file* f){
  free(f->buffer);
  fclose(f->file);
}


cr_file* cr_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;
  }

  cr_file* a=(cr_file*)malloc(sizeof(cr_file));
  a->file=f;
  a->bufferlength=buffersize;
  a->usedbuffer=buffersize; //Start off with no characters, so refill will work as expec$
  a->buffer=(char*)malloc(sizeof(char)*buffersize);

  refill(a);
  return a;
}




char cr_read_byte(cr_file* f){
        if (f->usedbuffer == f->bufferlength){
                refill(f); //We call refill function when the value of the usedbuffer is$
                printf("\n Buffer has been refilled \n"); //Shows us when the buffer has$
}
        bytes++; //inc bytes counter
        int len=f->bufferlength; //int var which holds the length for the buffer
        printf("\n We have %d bytes transfered to buffer \n", len); //prints after every$
        return f->buffer[f->usedbuffer++];//We return the letter from the buffer which i$
        return EOF;
}
```

```c
#import "cache_reader.h"
#import "cache_reader.c"

//Simple file display to show how easy it is to use the cached reader functions

int main(){
  char c;

  //Open a file
  cr_file* f = cr_open("text",20);

  //While there are useful bytes coming from it
  while((c=cr_read_byte(f))!=EOF)
    //Print them
    printf("%c",c);

  //Then close the file
  cr_close(f);
  printf(" Bytes read: %i \n", bytes);
  printf(" Buffer refill: %i \n", refill);

  //And finish
  return 0;
}
```

OUTPUT

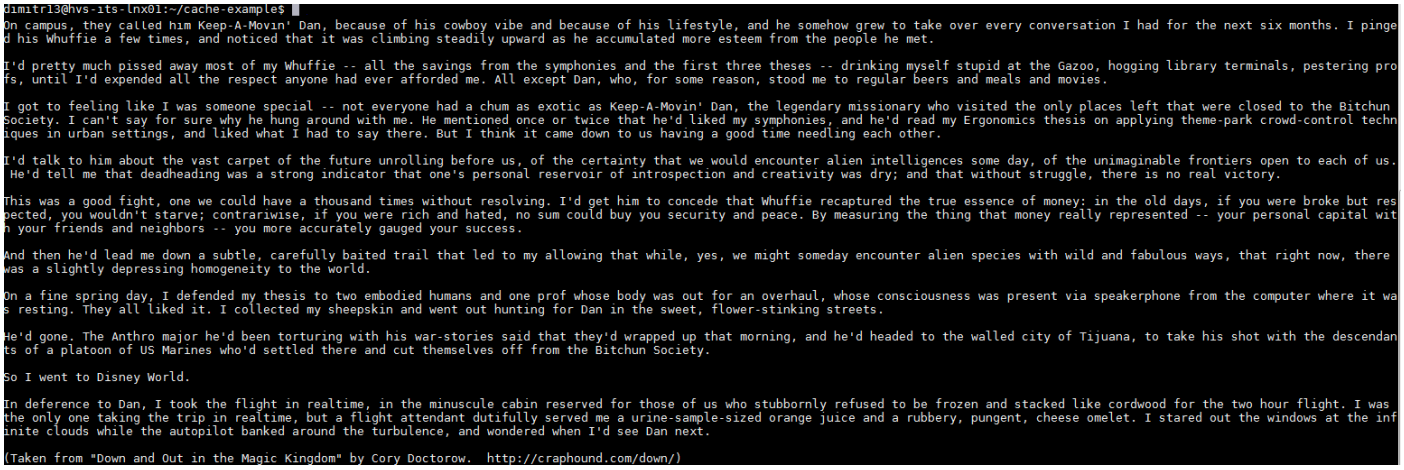
Lab 10: The Cache Buffer from week 8 with system calls

a) Brief description of the activity

For our last lab on this coursework we are exploring how can we change the old code for cache buffer and add system calls to it. From the slides and lab exercises the instructions showed us that we have to swap the already existing open, read, close with the ones fopen, fread, fclose. A few libraries have to be added “fcntl.h”,”unistd.h”,”sys/stat.h” – all need for the code to be able to send system calls to the machine (read, close and open calls). For the last task the FILE* pointer has to be changed to a simple integer. Caching on the library has to be reduced and thus performance of the buffer will get better. All of this can be seen in the outputs in the screenshots provided for this task.

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

```
    int refill(cr_file* buff){
    //Refills a buffer
    //Called when a buffer has been refiled
    if(buff->usedbuffer!=buff->bufferlength)
            return 0;
    else{
            buff->usedbuffer=0;
            int len=read(buff->file, buff->buffer, buff->bufferlength); //changing the fread with one for read
            if(len<buff->bufferlength)
                for(int i=len;i<buff->bufferlength;i++)
                        buff->buffer[i]=EOF;  //access the buffer as an array
            return len;
}


}

void cr_close(cr_file* f){
    free(f->buffer);
    close(f->file);  //fclose is changed with close
}
cr_file* cr_open(char * filename, int buffersize){

                int f;   //using the int declared in the header
            if ((f = open(filename, O_RDONLY, O_DIRECT)) == 0)     // fopen function changed to the system call function
“open”. using O_DIRECT to prevent caching

{   fprintf(stderr, "Cannot open %s\n", filename);
            return 0;
}

cr_file* a=(cr_file*)malloc(sizeof(cr_file));
a->file=f;
a->bufferlength=buffersize;
a->usedbuffer=buffersize; //When opening a file, the file is empty so refil is going to be called
a->buffer=(char*)pvalloc(sizeof(char)*buffersize);      // pvalloc insted of malloc to prevent faults in segmentation
refill(a);
return a;
}
```



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

cache_reader.h file:

```
#include <stdio.h>
#include <stdlib.h>

// func below are added to provide access to the needed system calls
#include <fcntl.h> // for the open call
#include <unistd.h> //allows us to call the close and read
#include <sys/stat.h> //allows us open call

typedef struct
{
  int file;
  int bufferlength;
  Int usedbuffer;     // current point in the buffer
  char* buffer;       // pointer to a piece of memory
                      //has same length as "bufferlength"
} cr_file;
```

Reducing the cache drastically reduces performance and it can be seen how the buffer reacts to the information received and the statistic it prints wit refills

Bibliography

CS UIC EDU. [Online]
Available at: https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/9_VirtualMemory.html
[Accessed 25 02 2017].

krishna's Operating System. [Online]
Available at: https://books.google.co.uk/books?id=DuXvCF0EfDIC&pg=SA6-PA92&lpg=SA6-PA92&dq=cache+buffer+c&source=bl&ots=vjXdiCn43g&sig=VYV0FZavaeUN4Utr80vFqKsOtqM&hl=en&sa=X&ved=0a

hUKEwjIjpb0i7bSAhUoIcAKHSPoAAIQ6AEIPzAF#v=onepage&q=cache%20buffer%20c&f=false
[Accessed 25 02 2017].

Matt How to C. [Online]
Available at: https://matt.sh/howto-c
[Accessed 22 02 2017].

Stackoverflow - What does it mean by buffer?. [Online]
Available at: http://stackoverflow.com/questions/648309/what-does-it-mean-by-buffer
[Accessed 20 02 2017].

Tutorials on point. [Online]
Available at: https://www.tutorialspoint.com/cprogramming/c_pointers.htm
[Accessed 23 02 2017].

Lab5

dimitr13@hvs-its-lnx01:~/pragmalinux-img$ cat /proc/devices

```
dimitr13@hvs-its-lnx01:~/pragmalinux-img$ cat /proc/devices
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  5 ttyprintk
  6 lp
  7 vcs
 10 misc
 13 input
 21 sg
 29 fb
 89 i2c
108 ppp
128 ptm
136 pts
180 usb
189 usb_device
248 hidraw
249 bsg
250 watchdog
251 rtc
252 dimmctl
253 ndctl
254 tpm

Block devices:
  1 ramdisk
  2 fd
259 blkext
  7 loop
  8 sd
  9 md
 11 sr
 65 sd
 66 sd
 67 sd
 68 sd
 69 sd
 70 sd
 71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
252 device-mapper
253 virtblk
254 mdp
```

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

cat /proc/cpuinfo

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 45
model name      : Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz
stepping        : 7
microcode       : 0xffffffff
cpu MHz         : 2394.211
cache size      : 20480 KB
```

```
processor       : 2
vendor_id       : GenuineIntel
cpu family      : 6
model           : 45
model name      : Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz
stepping        : 7
microcode       : 0xffffffff
cpu MHz         : 2394.211
cache size      : 20480 KB
```

dimitr13@hvs-its-lnx01:/proc$ grep "vendor" cpuinfo

```
vendor_id       : GenuineIntel
vendor_id       : GenuineIntel
vendor_id       : GenuineIntel
vendor_id       : GenuineIntel
```

dimitr13@hvs-its-lnx01:/proc$ grep "model name" cpuinfo

```
model name      : Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU E5-4640 0 @ 2.40GHz
```

Show the parameters that are passed to the kernel when starting up Linux

```
dimitr13@hvs-its-lnx01:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-4.4.0-47-generic root=UUID=11da70cc-f7d4-4e73-8746-a2b1e1f62539 ro elevator=noop
```

Using /proc/loadavg show the CPU utilisation in the last 5 and 10 minutes and the id of the last process used.

```
dimitr13@hvs-its-lnx01:~$ cat /proc/loadavg
0.18 0.12 0.17 1/282 51082
```

The first three fields are the one-minute, five-minute, and fifteen-minute load averages.

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

```
dimitr13@hvs-its-lnx01:~$ /proc/diskstats
-bash: /proc/diskstats: Permission denied
dimitr13@hvs-its-lnx01:~$ cat /proc/diskstats
   1       0 ram0 0 0 0 0 0 0 0 0 0 0 0
   1       1 ram1 0 0 0 0 0 0 0 0 0 0 0
   1       2 ram2 0 0 0 0 0 0 0 0 0 0 0
   1       3 ram3 0 0 0 0 0 0 0 0 0 0 0
   1       4 ram4 0 0 0 0 0 0 0 0 0 0 0
   1       5 ram5 0 0 0 0 0 0 0 0 0 0 0
   1       6 ram6 0 0 0 0 0 0 0 0 0 0 0
   1       7 ram7 0 0 0 0 0 0 0 0 0 0 0
   1       8 ram8 0 0 0 0 0 0 0 0 0 0 0
   1       9 ram9 0 0 0 0 0 0 0 0 0 0 0
   1      10 ram10 0 0 0 0 0 0 0 0 0 0 0
   1      11 ram11 0 0 0 0 0 0 0 0 0 0 0
   1      12 ram12 0 0 0 0 0 0 0 0 0 0 0
   1      13 ram13 0 0 0 0 0 0 0 0 0 0 0
   1      14 ram14 0 0 0 0 0 0 0 0 0 0 0
   1      15 ram15 0 0 0 0 0 0 0 0 0 0 0
   7       0 loop0 0 0 0 0 0 0 0 0 0 0 0
   7       1 loop1 0 0 0 0 0 0 0 0 0 0 0
   7       2 loop2 0 0 0 0 0 0 0 0 0 0 0
   7       3 loop3 0 0 0 0 0 0 0 0 0 0 0
   7       4 loop4 0 0 0 0 0 0 0 0 0 0 0
   7       5 loop5 0 0 0 0 0 0 0 0 0 0 0
   7       6 loop6 0 0 0 0 0 0 0 0 0 0 0
   7       7 loop7 0 0 0 0 0 0 0 0 0 0 0
  11       0 sr0 0 0 0 0 0 0 0 0 0 0 0
   2       0 fd0 0 0 0 0 0 0 0 0 0 0 0
   8       0 sda 16501 24 589590 154784 255124 27216 2966992 482384 0 387796 637024
   8       1 sda1 16362 24 581026 154128 140219 27216 2966992 308228 0 214664 462248
   8       2 sda2 2 0 4 0 0 0 0 0 0 0 0
   8       5 sda5 50 0 4408 56 0 0 0 0 0 44 56
   8      16 sdb 3675 0 52204 46464 25678 32857 493656 77372 0 78116 123824
   8      17 sdb1 469 0 9514 6468 46 12 464 404 0 1772 6872
   8      18 sdb2 3177 0 40602 39832 25630 32845 493192 76968 0 76856 116788
```

We are mostly interested in devices sda , sda1 and sdb as this are the only physical devices found on the