**Name - Vinay Ruhil**

**Course - BSc(H) Computer Science**

**Roll No. - 16115**

**Subject - OS**

**Qn 5: Write a program to report behaviour of Linux kernel including kernel version, CPU type and CPU information.**

#include <stdio.h>

#include <stdlib.h>

int main() {

system("clear");

system("echo");

printf("The CPU Model: ");

system("cat /proc/cpuinfo | grep -m1 'model name' | cut -c 14-");

printf("\nThe Kernel Version: ");

system("cat /proc/sys/kernel/osrelease");

printf("\nThe amount of time CPU has spent in user mode: ");

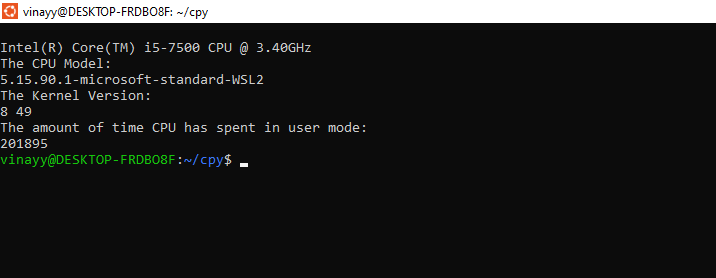
system("cat /proc/stat | grep -m1 'cpu' | cut -c 15-18");

printf("\nThe number of context switches: ");

system("cat /proc/stat | grep 'ctxt' | cut -c 6-");

return 0;

}

**Output**

**Qn 6: Write a program to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (Memory information)**

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[]) {

printf("The number of processes since the system was last booted:\n");

system("cat /proc/stat | grep 'processes' | cut -c 11-15");

printf("\nConfigured memory details:\n");

printf("The total memory in Kilobytes is:\n");

system("free –kilo | awk '/Mem:/ {print $2}'");

printf("The used memory in Kilobytes is:\n");

system("free –kilo | awk '/Mem:/ {print $3}'");

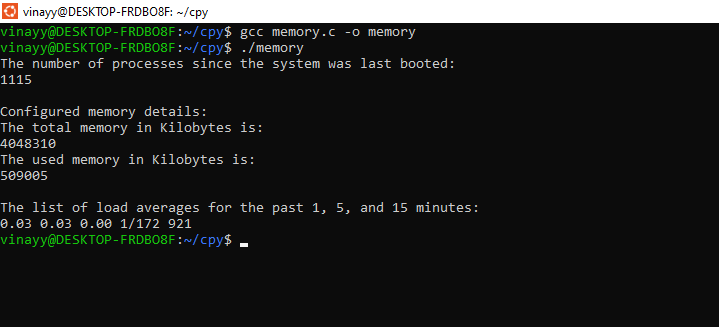
printf("\nThe list of load averages for the past 1, 5, and 15 minutes:\n");

system("cat /proc/loadavg");

return 0;

}

**Output**

****

**Qn 7: Write a program to copy files using system calls.**

**(This question has to be done via read/write system calls only.)**

#include<io.h>

#include<stdio.h>

#include<stdlib.h>

#include<fcntl.h>

#include<unistd.h>

#include<sys/stat.h>

void main()

{

char source[10];

char destination[10];

int s;

int d;

int c=0;

char temp[50];

printf("Enter the name of the source file ");

scanf("%s", source);

printf("Enter the name of the destination file ");

scanf("%s", destination);

s=open(source, 0);

if(s==-1)

{

printf("file open error!!!!!!");

exit(0);

}

d=open(destination, 1);

if(d==-1)

{

d = create(destination, 0666);

}

while((c=read(s, temp, sizeof(temp))) >0)

{

write(d, temp, c);

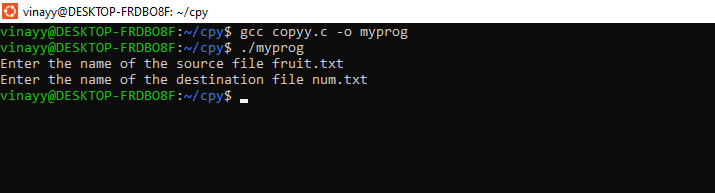
}

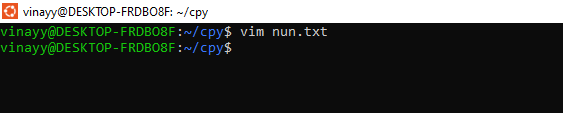
close(s);

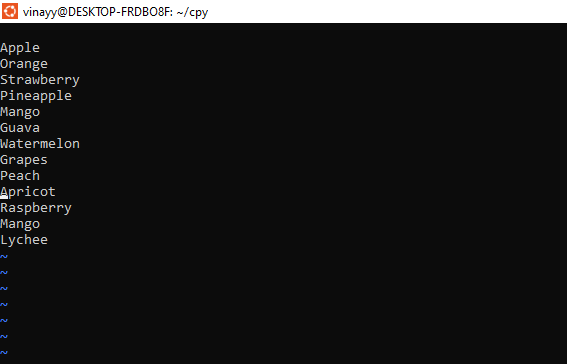
close(d);

}

**Output:**

****

****

****