Assignment 3

-Harsh Bansal

-2016UCP1468

1. Perform these linux commands:

cal-print the current month's calender

date-prints date

echo- gives as o/p on terminal

bc-terminal calculator

eg-echo "12+7"|bc

passwd- change password

who-The basic who command with no command-line arguments shows the names of users that

are currently logged in, and depending on which Unix/Linux system you are using, may

also show the terminal they're logged in on, and the time they logged in

uname-prints the name, version and other details about the current machine and

the operating system running on it.

tty-print the file name of the terminal connected to standard input.

cancel-force quit

wc-word count

pg-pg is a terminal pager program on Unix and Unix-like systems for viewing text files.

It can also be used to page through the output of a command via a pipe.

pg uses an interface similar to vi, but commands are different.

more-more is a command to view the contents of a text file one screen at a time.

head-head will print the first 10 lines of its input to the standard output.

tail-isplay the tail end of a text file or piped data

gzip-common way of compressing files

unzip-unzip a file

chmod-chmod is the command and system call which may change the access permissions to file system objects.

2. Write a program for matrix multiplication using pthread\_join(). No mutex should be used.

Code:

#include<stdio.h>

#include<pthread.h>

int step;

int n = 4;

int a[4][4],b[4][4],c[4][4];

void\* multiply(void \*arg)

{

int iterator = step++;

int i,j,k;

for(i=iterator;i<iterator+1;i++)

{

for(j=0;j<n;j++)

for(k=0;k<n;k++)

c[i][j]+=a[i][k]\*b[k][j];

}

}

void main()

{

int i,j;

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

a[i][j]=rand()%10;

b[i][j]=rand()%10;

}

}

printf("\nMatrix A is:\n");

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

printf("%d ",a[i][j]);

}

printf("\n");

}

printf("\nMatrix B is:\n");

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

printf("%d ",b[i][j]);

}

printf("\n");

}

pthread\_t p[4];

for(i=0;i<4;i++)

{

int\* var;

pthread\_create(&p[i],NULL,multiply,(void \*)(var));

}

for(i=0;i<4;i++)

pthread\_join(p[i],NULL);

printf("\nThe multiplication of the matrix is:\n");

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

printf("%d ",c[i][j]);

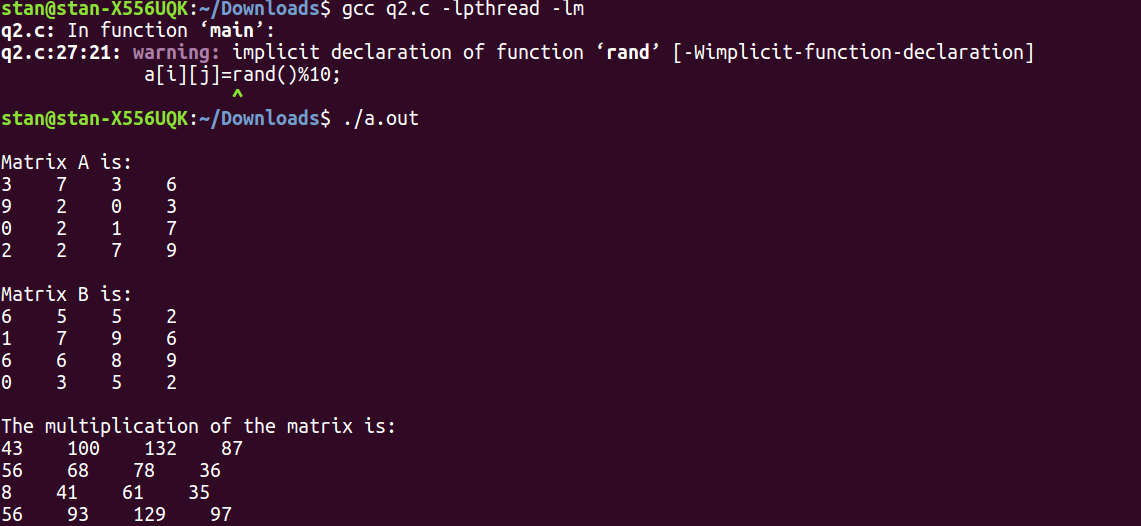
}

printf("\n");

}

}

Output:



3. Write a C program in which a filename is passed as a command line argument. In case a wrong name or no file name is passed is passed using CLA it should print an error message using perror().

Code:

#include<stdio.h>

void main(int argc, char\*\* argv)

{

int i;

if(argc<2)

{

perror("ERROR:No file name entered");

return;

}

else

{FILE \*fp;

fp = fopen(argv[1],"r");

if(fp==NULL)

{

perror("Error: ");

return;

}

else

{

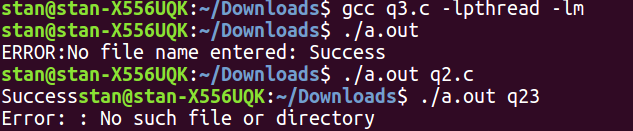
printf("Success");

}

fclose(fp);}

}

Output:



4. Write a C program that continuously print(with a sleep time of 1 second) the process id and the total sleep time. This program should send a SIGINT signal using signal() call, when a particular key is pressed.

Code:

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <signal.h>

#include <pthread.h>

int tsleep=0;

void sighandler(int signum)

{

printf("Caught signal %d, coming out...\n", signum);

exit(1);

}

void\* thread\_func(void\* parameter)

{

int pid=\*((int\*)parameter);

signal(SIGINT, sighandler);

while(1)

{

printf("%d",pid);

sleep(1);

tsleep++;

printf(" Total sleep time: %d\n",tsleep);

}

return(0);

}

int main()

{

pthread\_t p[10];

int i;

int \*parameter = malloc(sizeof(int));

for(i=0;i<10;i++)

{

\*parameter = i;

void \*val;

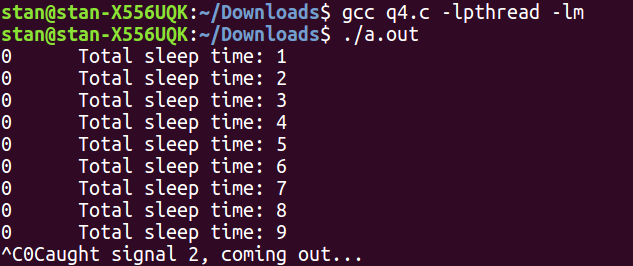
pthread\_create(&p[i],0,thread\_func,(void \*)parameter);

pthread\_join(p[i],&val);

}

}

Output:



5.Write a program using pthread to find out the sum of following series:

1+4+9+16+.......+ n. Here main function should write the final output on screen. Main thread will create the child threads and child threads will find out the sum of series.

Code:

#include<stdio.h>

#include<pthread.h>

#include<stdlib.h>

#include<unistd.h>

int sum;

void\* sumOfNo(void\* args)

{

int x=\*((int\*)args);

sum+=x\*x;

return NULL;

}

int main()

{

int n;

printf("\nEnter the number of terms");

scanf("%d",&n);

pthread\_t p[n];

int i;

int \*parameter = malloc(sizeof(int));

for(i=1;i<=n;i++)

{

\*parameter = i;

void \*val;

pthread\_create(&p[i],0,sumOfNo,(void \*)parameter);

pthread\_join(p[i],&val);

}

printf("Sum of first %d numbers is %d",n,sum);

}

Output:

