Student Name- Avishek Chatterjee

Student ID-11700072

Email Address- avisheksantu@gmail.com

Github Link- sukhwindermanni/operating system

# Q1.Write a multithreading program in c that outputs prime numbers. This program should work as follows .The user will run the program and will enter a number on the command line . The program will then create a separate thread that output all the prims numbers less than or equal to the number created by the user.

**SOURCE CODE**

# #include <windows.h>

# #include <iostream>

# #define MAX\_THREADS 1

# using namespace std;

# DWORD WINAPI Prime (LPVOID);

# HANDLE hThreads [MAX\_THREADS];

# DWORD id [MAX\_THREADS];

# DWORD waiter;

# int check\_prime(int b)

# {

# int c;

# for ( c = 2 ; c <= b - 1 ; c++ )

# {

# if ( b%c == 0 )

# return 0;

# }

# if ( c == b )

# return 1;

# }

# DWORD WINAPI Prime(LPVOID Param)

# {

# DWORD Number = \*(DWORD\*)Param;

# for (DWORD i=2;i<=Number;i++)

# {

# if(check\_prime(i)==0)

# cout <<"";

# else

# cout<<i;

# }

# return 0;

# }

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

# {

# // same your code

# return 0;

# }

Description- In the above program we have to output the prime numbers. The user will run the program and will enter a number on the command line. The program creates a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.

Coding Language- C

Lines Of Code-37 lines

# Q26.CPU schedules N processes which arrives at different time intervals and each process is allocated the CPU for a specific user input time unit, processes are scheduled using a preemptive round robin scheduling algorithm. Each process must be assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to processes one task has priority 0.The length of a time Quantum is T units, where T is the custom time considered as time quantum for processing. If a process is preempted by a higher priority process the preempted process is placed at the end of the queue?

# SOURCE CODE

# #include<pthread.h>

# #include<stdio.h>

# #define NUM\_THREADS 5

# int main(int args,char \*args[])

# {

# int i,scope;

# pthread\_t tid[NUM\_THREADS];

# pthread\_attr\_t attr;

# /\*get the default attribute\*/

# pthread\_atte\_init(&attr);

# /\*first inquire on the current scope\*/

# if(pthread\_attr\_getscope(&attr,&scope)!=0)

# fprintf(stderr,"unable to get scheduling scope\n");

# else

# {

# if(scope==PTHREAD\_SCOPE\_PROCESS)

# printf("PTHREAD\_SCOPE\_PROCESS");

# elseif(scope==PTHREAD\_SCOPE\_SYSTEM)

# printf("PTHREAD\_SCOPE\_SYSTEM");

# else

# fprintf(strderr,"illegal scope value.\n");

# }

# /\*set the scheduling algorith to pcs or scs \*/

# pthread\_attr\_setscope(&attr,PTHREAD\_SCOPE\_SYSTEM);

# /\* create the threads \*/

# for(i=0;i<NUM\_THREADS;i++)

# pthread\_create(&tid[i],&attr,runner,NULL);

# /\* now join on each thread\*/

# for(i=0;i<NUM\_THREAD;i++)

# pthread\_join(tid[i],NULL);

# }

# /\* each thread will begin control in the function \*/

# void \*runner(void \*param)

# {

# /\* do some work....\*/

# pthread\_exit(0);

# }

# Description- CPU schedules N processes which arrives at different time intervals and each process is allocated the CPU for a specific user input time unit, processes are scheduled using a preemptive round robin scheduling algorithm. Each process must be assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to processes one task has priority 0.The length of a time Quantum is T units, where T is the custom time considered as time quantum for processing. If a process is preempted by a higher priority process the preempted process is placed at the end of the queue.

Code written in – C Language

Lines of Code-39

# 