Operating systems assignment (CA-3)

Name:- Pasupuleti L V Naga Satya Sai Varaprasad

Roll.No:- 04

Email Id:- [saivaraprasad195@gmail.com](mailto:saivaraprasad195@gmail.com)

Reg.No:- 11703382

Question

4.)

Consider the scenario, there are 3 student processes and 1 teacher process. Students are supposed to do their assignments and they need 3 things for that-pen, paper and question paper. The teacher has an infinite supply of all the three things. One student has pen, another has paper and another has question paper. The teacher places two things on a shared table and the student having the third complementary thing makes the assignment and

tells the teacher on completion. The teacher then places another two things out of the three and again the student having the third thing makes the assignment and tells the teacher on completion. This cycle continues. WAP to synchronise the teacher and the students.

* Two types of people can enter into a library- students and teachers. After entering the library, the visitor searches for the required books and gets them. In order to get them issued, he goes to the single CPU which is there to process the issuing of books. Two types of queues are there at the counter-one for students and one for teachers. A student goes and stands at the tail of the queue for students and similarly the teacher goes and stands at the tail of the queue for teachers(FIFO).Ifastudentisbeingservicedandateacherarrivesatthecounter,hewouldbe the next person to get service (PRIORITY-non preemptive). If two teachers arrive at the same time, they will stand in their queue to getservice

(FIFO). WAP to ensure that the system works in a non-chaotic manner.

if a teacher is being served and during the period when he is being served, another teacher comes, then that teacher would get the service next. This process might continue leading to increase in waiting time of students. Ensure in your program that the waiting time of students is minimized.

# Explanation of code:-

* While statement is used to check the condition continuously.
* If statements are used for individual student, and in that condition is checked for the student who has the third object other that the two objects placed by the teacher.
* The condition is checked and the respective students complete the assignment.

# Code:-

#include<stdio.h>

#include<stdbool.h>

struct requirement

{

bool pen ;

bool paper ;

bool question\_paper ;

bool all\_three ;

};

int main()

{

int n=3;

struct requirement s[n];

s[0].pen=true;

s[0].paper = false;

s[0].question\_paper = false;

s[0].all\_three= false;

s[1].pen=false;

s[1].paper = true;

s[1].question\_paper = false;

s[1].all\_three = false;

s[2].pen=false;

s[2].paper = false;

s[2].question\_paper = true;

s[2].all\_three = false ;

while(s[0].all\_three==false||s[1].all\_three==false||s[2].all\_three==fals e)

{

int ch1,ch2;

printf("\nResources:\n1.pen\n2.paper\n3.question paper\n Enter the two things which is to be placed on the shared table: ");

scanf("%d%d",&ch1,&ch2);

if(ch1==1 && ch2==2 && s[2].all\_three==false)

{

s[2].all\_three=true ;

printf("Third Student has completed the task\n");

}

if(ch1==2 && ch2==3 && s[0].all\_three==false)

{

s[0].all\_three=true;

printf("First Student has completed the task\n");

}

if(ch1==1 && ch2==3 && s[1].all\_three==false)

{

s[1].all\_three=true;

printf("Second Student has completed the task\n");

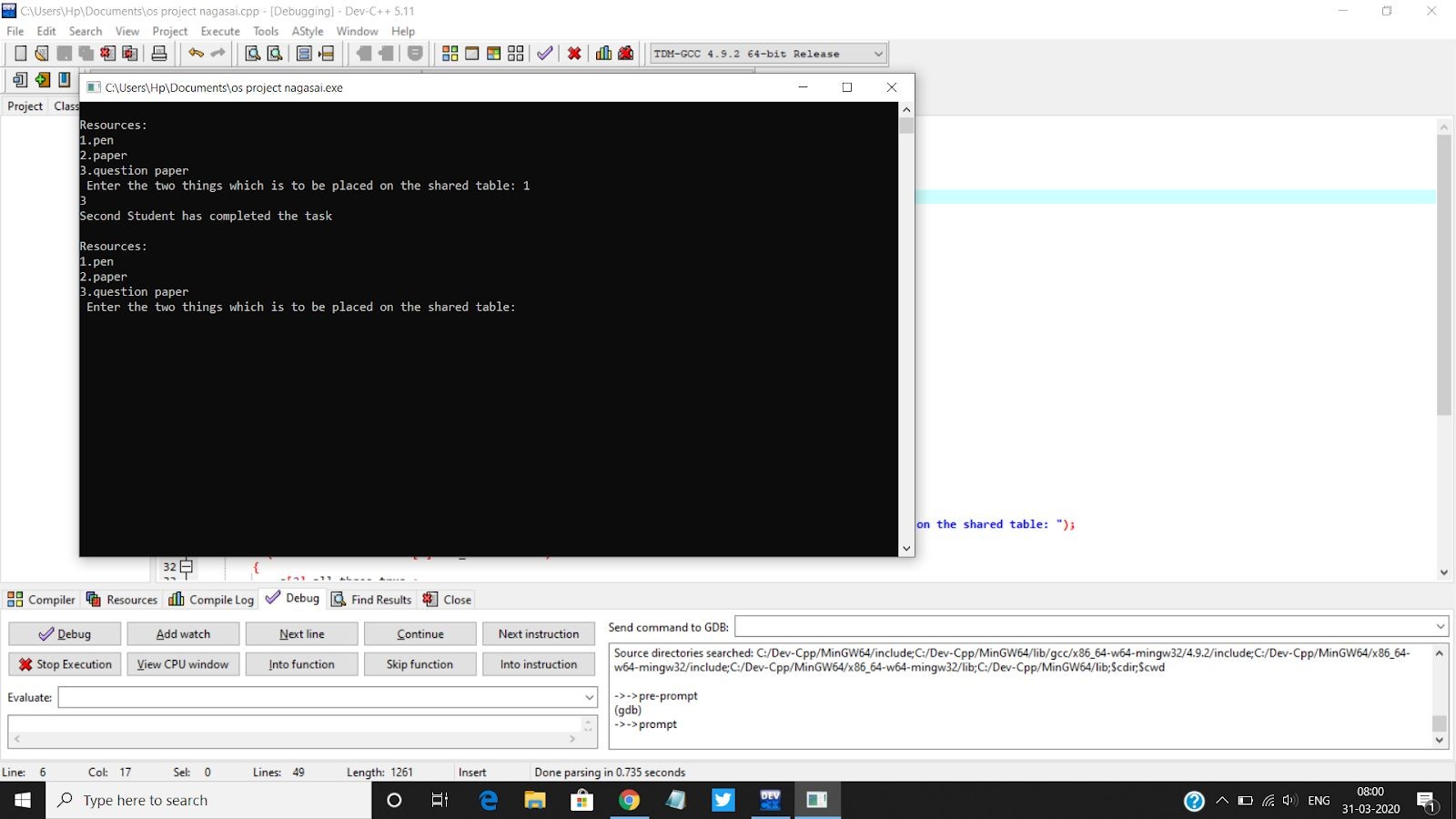
}

}

printf("All the students now have completed their respective tasks succesfully\n");

return 0; }

# Screenshot

Description

# Priority Scheduling

Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a

round-robin or FCFS basis.

Types of Priority Scheduling

Priority scheduling divided into two main types:

# Preemptive Scheduling

In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

# Non-Preemptive Scheduling

In this type of scheduling method, the CPU has been allocated to as specific process. The process that keeps the CPU busy, will releasethe CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That's because it doesn't need special hardware (for example, a timer) like preemptive scheduling.

# Characteristics of Priority Scheduling

* A CPU algorithm that schedules processes based on priority. It used in Operating systems for performing batch processes.
* If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
* In priority scheduling, a number is assigned to each process that indicates its priority level. Lower the number, higher is the priority.
* In this type of scheduling algorithm, if a newer process arrives, that is having a higher priority than the currently running process, then the currently running process is preempted.

# Complexity:-

* O(n) complexity. Because of the while loop.
* O(1) complexity. For every “ If ” condition.