NAME:K.V.Sai Manoj

SEC:K17YJ

Roll no:B(53)

Ques. 9. Design a scheduler that uses a preemptive priority scheduling algorithm based on dynamically changing priority.

Larger number for priority indicates higher priority.

Assume that the following processes with arrival time and service time wants to execute (for reference):

ProcessID Arrival Time Service Time

P1 0 4

P2 1 1

P3 2 2

P4 3 1

When the process starts execution (i.e. CPU assigned), priority for that process changes at the rate of m=1.

When the process waits for CPU in the ready queue (but not yet started execution), its priority changes at a rate n=2.

All the processes are initially assigned priority value of 0 when they enter ready queue for the first time .

The time slice for each process is q = 1. When two processes want to join ready queue simultaneously,

the process which has not executed recently is given priority.

Calculate the average waiting time for each process. The program must be generic i.e. number of processes,

their burst time and arrival time must be entered by user.

#include<unistd.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<stdlib.h>

#include<stdio.h>

int main()

{

int fd,pt;

struct flock r1,r2;

fd=open("myfile22",O\_CREAT|O\_RDWR,0666);

r1.l\_type=F\_WRLCK;

r1.l\_whence=SEEK\_SET;

r1.l\_start=40;

r1.l\_len=20;

pt=fcntl(fd,F\_SETLK,&r1);

if(pt==-1)

{

printf("Error in locking region1 in p2\n");

}

else

{

printf("Rescource allocated1 in p2 40 - 60\n");

}sleep(5);

r2.l\_type=F\_WRLCK;

r2.l\_whence=SEEK\_SET;

r2.l\_start=10;

r2.l\_len=20;

pt=-1;

while(pt==-1)

{

pt=fcntl(fd,F\_SETLK,&r2);

if(pt==-1)

{

printf(" waiting for resource(10-30) in P2\n"); sleep(3);

}

else

{

printf("Region2Locked in p2 10 - 30\n");

}

}

sleep(20);

}