**Os-C programs**

---------------------------------------

Last la Notes Erruku Marakaama

Check Pannikonga Friends 👇🏼

---------------------------------------

1. Fcfs Cpu Scheduling

2.Fcfs Process Scheduling

( same program for both )

#include <stdio.h>

int main() {

int n;

printf("Enter number of processes: ");

scanf("%d", &n);

int bt[n], wt[n], tat[n];

printf("Enter burst times:\n");

for (int i = 0; i < n; i++) scanf("%d", &bt[i]);

wt[0] = 0;

for (int i = 1; i < n; i++) wt[i] = wt[i-1] + bt[i-1];

for (int i = 0; i < n; i++) tat[i] = bt[i] + wt[i];

float tw = 0, tt = 0;

printf("Process\tBT\tWT\tTAT\n");

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

printf("%d\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);

tw += wt[i]; tt += tat[i];

}

printf("Avg WT = %.2f\nAvg TAT = %.2f\n", tw/n, tt/n);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Shortest Job First

#include <stdio.h>

int main() {

int n, bt[10], p[10], i, j, t=0, temp;

printf("No. of processes: "); scanf("%d",&n);

printf("Burst times:\n");

for(i=0;i<n;i++){ printf("P%d: ",i+1); scanf("%d",&bt[i]); p[i]=i+1; }

for(i=0;i<n-1;i++) // sort by burst time

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

if(bt[i]>bt[j]){ temp=bt[i]; bt[i]=bt[j]; bt[j]=temp;

temp=p[i]; p[i]=p[j]; p[j]=temp; }

for(i=0;i<n;i++){ t+=bt[i]; printf("P%d finished at %d\n",p[i],t); }

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. IPC

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <signal.h>

void h(int s) {

printf("Child received SIGUSR1\n");

}

int main() {

pid\_t p = fork();

if (p < 0)

exit(1);

if (p == 0) { // Child

signal(SIGUSR1, h);

printf("Child waiting...\n");

pause();

} else { // Parent

sleep(1);

printf("Parent sending SIGUSR1...\n");

kill(p, SIGUSR1);

sleep(1);

}

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Deadlock Detection

#include <stdio.h>

int main() {

int n, wait[10][10];

printf("Enter number of processes: ");

scanf("%d", &n);

printf("Enter wait matrix (1 if Pi waits for Pj, else 0):\n");

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

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

scanf("%d",&wait[i][j]);

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

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

if(wait[i][j] && wait[j][i]) {

printf("Deadlock between P%d and P%d\n", i+1, j+1);

return 0;

}

printf("No Deadlock Detected\n");

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Mutual Exclusive

#include <stdio.h>

#include <unistd.h>

int main() {

int turn = 1, n = 2, i;

for(i = 1; i <= 5; i++) {

if(turn == 1) {

printf("Process 1 in Critical Section\n");

sleep(1);

printf("Process 1 leaving Critical Section\n");

turn = 2;

}

else {

printf("Process 2 in Critical Section\n");

sleep(1);

printf("Process 2 leaving Critical Section\n");

turn = 1;

}

}

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Round Robin

#include <stdio.h>

int main() {

int n, bt[10], tq, rem[10], t=0, done;

printf("No. of processes: "); scanf("%d",&n);

printf("Burst times:\n");

for(int i=0;i<n;i++){ printf("P%d: ",i+1); scanf("%d",&bt[i]); rem[i]=bt[i]; }

printf("Time quantum: "); scanf("%d",&tq);

do {

done=1;

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

if(rem[i]>0){ done=0;

if(rem[i]>tq){ rem[i]-=tq; t+=tq; }

else { t+=rem[i]; printf("P%d finished at %d\n",i+1,t); rem[i]=0; }

}

} while(!done);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Note : [👋🏻]

👉🏼 Fcfs Cpu Scheduling and Fcfs Process Scheduling (Write same program for Both) No Need to Worry

👉🏼 I Think ,

Shortest Job First And Round Robin Program Having Output, So you can write same program but change the names at certain positions to not get caught

👉🏼 In Deadlock Detection Program the Output must contain Matrix values

👉🏼 Mutual Exclusive Program the Output must wait for the process ( i.e , sleep() )

👉🏼 All the Best Friends