

RTOS operation and simple demonstration of RTOS task scheduling

Readme File

Submitted by :

Manav Gopal (B19CSE112)

Kartik Narayan (B19CSE110)

NOTE :

- There are many files in the Files folder. The files referred here in the readme are all present in the folder "Files".
- The main.cpp will create the files for all five scheduling algorithms which contains the average, maximum and minimum AWT, ART and ATT.
- This readme file contains the procedure for running the attached files.
- The screenshots for the results and the produced graphs are being included in the report document.
- It's preferred to run files in a UNIX based operating system.

Go into the Files directory and perform the following parts to get the result.

----- PART A -----

Compiling :

```
g++ a.cpp -o a
```

Running :

```
./a
```

Input :

This program waits for the user to give an input 'n' which is the number of processes.

Output :

Running produces a series of numbers with each row of format "Process ID, CPU Burst, Arrival Time, Priority".

The output can be seen in the file "out.txt" as well as on terminal.

For this, I assume the mean for exponential distribution to be 2.

----- PART B -----

Compiling :

```
g++ -c b.cpp fcfs.cpp sjfnp.cpp sjfp.cpp rr.cpp priority.cpp
```

```
g++ b.o fcfs.o sjfnp.o sjfp.o rr.o priority.o -o final
```

Running :

```
./final
```

Input :

The input for this code is taken from the "out.txt" generated in PART A of the project.

Output :

The output will show the ready queue simulation for all 5 algorithms, which is followed by AWT, ART, and ATT of all 5 scheduling algorithms.

Also Three .txt document namely AWT.txt, ATT.txt, ART.txt are created and the values are saved into that.

----- PART C -----

Running :

python3 c.py

Note : Prior installation of matplotlib is required.

Command : pip install matplotlib

Output :

Upon running this, we get 3 txt files, ATT.txt ART.txt an ATT.txt which have maximum, minimum and average respective times for N=10,20,30,40,50 .

Each row is formatted in the form "N FCFS SJFNP SJFP RRPS"

The graphs will be displayed on the screen and as well as stored in the graphs directory.

Now come to the directory where the main.cpp file is present.

----- PART D -----

NOTE :

- This part is done in the extra 3 days taken.
- If the program takes more time then run interrupt the terminal and run it again

Now the main.cpp file can be ran directly to get the AWT, ART and ATT values of all scheduling algorithms :

Compiling :

g++ -o out main.cpp

Running :

./out <n>

Here n is the number of the process:

Input :

Input will be the number of processes which should be given along with running the executable file as parameter

Output :

As an output of running this code we will get 6 text files. One file will contain the process details (Burst time, arrival time, priority etc) generated randomly. The other five files will contain the ART, AWT and ATT information.