Ubuntastic

Lazy Penguin Scheduler

Algorithms

⏳ 🐧⌛🐧⌛🐧⌛🐧⌛🐧⌛🐧

SRTN

1. Checks if all processes have finished their execution
    a. *yes: scheduling process has finished*
    b. *no: → step 2*
2. Checks for incoming processes at the current time step and inserts them into the heap
    a. *yes: → step 3*
    b. *no: → repeat step 2*
3. peeks the heap for the SRT process
4. Checks if there's a running process
    a. *yes: compare it with the current running process and run the SRT one (and stops the running process in case if its running time is higher)*
    b. *no: set the running process with the SRT process*
5. repeat → step 1

Assumptions:

- if the next SRT process RT = the running process RT, the running process resumes

🔝

🐧🔝🐧🔝🐧🔝🐧🔝🐧🔝🐧

HPF

1. Checks if all processes have finished their execution
   a. *yes: scheduling process has finished*
   b. *no: → step 2*
2. Checks for incoming processes at the current time step and inserts them into the heap
   a. *yes: → step 3*
   b. *no: → repeat step 2*
3. Checks if there's a running process or if the running process hasn't finished its execution yet
   a. *yes: the running process continues its execution (repeat step 3)*
   b. *no: extract the next process from the heap*
4. repeat → step 1

Assumptions:

- lower priority value means higher priority

🔄 🐧🔄🐧🔄🐧🔄🐧🔄🐧🔄🐧

RR (Queue Implementation)

1. If there's a running process at the current time step → run it and decrease its time slice by one
2. Insert All incoming processes at this time step into the ready queue
3. check if the ready list is not empty
    a. yes: → *step 4 & 5*
    b. No: → *step 1*
4. If there's a running process & if either the process finished its execution time or if its time slice has just finished → stop the running process and run the first process in the queue
5. if no process is running → reset the time slice counter and run the first process in the queue
6. repeat → step 1

Assumptions:

- FIFO Queue

Data Structures

- Generator Processes → Queue
- PCB → Dynamic Array
- SRTN processes DS → Min Heap
- HPF processes DS → Min Heap
- RR processes DS → Queue
- Scheduler → Scheduler Struct

General Assumptions:

- processes starts from time = 1
- if a process finished at time t, the next process *if there's any in the ready list*, can either start in time t or t+1 depending on when the scheduler receives the `SIGUSR2 //finish process signal`

Task	Team Member
clear IPC resources	Ahmed
HPF	Somia
SRTN	Ahmed

Note

it's important to note that Phase Two will be primarily handled by Abdulrahman, Mariam and Somia.

→ regarding the time taken for each task:

we struggled with lots of bugs and inconvenient results so it took us forever to test, debug and to re-code

Task	Team member
Read Input Files & User Input	Somia
Initiate & Create Scheduler	Somia
Initiate & Create Clock	Ahmed
Processes Data structure	Somia
Send the Information to the Scheduler	Abdulrahman & Mariam
Signal Handling	Ahmed
IPC	Abdulrahman & Mariam
Scheduler.log & Scheduler.perf	Somia

Task	Team Member
RR	Abdulrahman & Mariam
PCB DS & tracking PCB	Somia
min Heap	Mariam
System Testing	Mariam
Process Class	Ahmed
`StartProcess()` `StopProcess()` `FinishProcess()` `ContinueProcess()`	Ahmed
GUI	Ahmed