Operating Systems Assignment

Mohamed Hassan Dawod

Ahmad Sameh El-Sayed

Data Structure used :

I used single “Priority Queue” for all the schedulers
in any time the top process in the ready queue will be the process which needs to run according to the chosen scheduler.

Algorithm explanation and results :

At first I tried to achieve the assumption as (single CPU system) so I forced all processes which are children of process generator to work on CPU[0] .

HPF :

the scheduler will run the top process in the queue .
if any new process arrived , the process generator will send it but the scheduler will not receive it till the running process finish ,then the scheduler will receive the new messages from the queue to add them to the ready queue .

SRTN :

the scheduler will run the top process in the queue.
If any new process arrived , the process generator will send it as a message in the queue and will send also a signal to the scheduler , so the scheduler will stop the running process and save it’s states then the scheduler will receive the new messages and add them to the ready queue .
The scheduler will check the top of the queue if it is already have process id then the scheduler will resume it ; if not (pid=0), the scheduler will fork it to start running.

RR :

the scheduler will run the top process in the queue.
Process will stop itself and the scheduler will save it’s states and it’s new position in the queue will be in the end (actually this is done by adding constant value to the process RR_value).
If new process arrived the scheduler will add it to the queue and it will have the high priority to run first (because it’s RR-value is less than all RR-values of the previously run processes).

Assumptions:

1- the process generator is the only process that run according to the Linux system scheduler (CFS) , so I consider it as a human that turn on the system by forking the clk & scheduler (HPF or SRTN or RR) and run tasks or processes at random times .

2- in SRTN , I assumed that at new signal arrival , the running process should be stopped even it’s remaining time is less than the running time of the new process
that’s because the scheduler needs to work at this moment to check the running time of the new processes .

3-in RR , to work in perfect way I assumed that the process will stop it self then the scheduler will start handling the situation .
If I let the process send alarm signal instead of stop signal . The output will be not correct because of the time and sleep system call.

Time taken for each task:

understanding the project files and planning : 4 hours
process generator : 3 hours
HPF : 5 hours.
SRTN : 10 hours.
RR : 2 hours.
Testing and debugging : 3 hours