Simple Scheduler in C

Contributors: Aditya Gupta & Abhishek Bansal

Individual Contribution

- Aditya: Refining the scheduler function , history with execution and waiting time , refining in shared memory implementation .
- Abhishek: starting code for scheduler , enqueue and dequeue functions ,priority setting , starting code for shared memory implementation .

Implementation

the steps for the implementation of the Scheduler are as follows:

1. We had previously implemented shell in the Assignment-2 . In this we had to add the funtionality of the submit command . We added it as an else if clause in our shell code . Now the command is added in a process_table with 4 queues for each priority .
2. Now we fork to make scheduler process to make it child of the shell process and in the scheduler we fork in while loop to make it's grandchild which will run the run_scheduled function , which has the main scheduler logic .
3. .Now the scheduler run every 10 secs or when you use the start command , which is used as a signal . In the scheduler process , we take the first n processes from the CPU if there are more than n processes or else the number of processes and run then for TSLICE seconds and stop them and take the next n cpu processes and so on till they terminate and exit there way out of the process_table , else if they are still running they are enqueued to the end of there priority queue .
4. . We have used kill(process.pid , SIGSTOP) to stop a process and kill(process.pid SIGCONT) to continue a process which was previously stopped and if it is a new process we execvp to make it's own process and store it in the process_table . We have used shared memory for the scheduler which runs as a daemon and independent process in the background to communicate with the shell by passing the process_table and running queue and helper variables in it .
5. Thus it is a complete implementation of a scheduler which receives process using submit command from the shell sends it to the scheduler background process and it shows the output on the STDOUT .

Github Repository

Steps to run:

1. you have to simply start the shell with giving NCPU and TSLICE as command line arguments with space seperated .You will get the printed message for the NCPU and TSLICE on the STDOUT for confirmation , now we are giving 2 options to the user either they can let the scheduler run at every 10 seconds or give a command as a signal .
2. now submit all the commands with their priority or if you are not giving priority it will be saved as 1 to the shell . Now the scheduler will give the output using a RR scheduling scheme and show it on the STDOUT . This is how you can schedule the process and get their outputs .