Report

1. Design

   This project applies two CPU structure, one for scheduling and the other for running child process.

   When the policy is FIFO, the scheduler simply forks all processes in the order of their ready time and sets their priority to the same. When the policy is SJF or PSJF, the scheduler maintains a linked list of readied processes that is sorted by expected remaining execution time. When the policy is RR, the scheduler maintains a queue of readied process.

   For SJF, PSJF and RR, the scheduler will set the priority of all readied processes to the same, and set that of the process to be run higher than the other processes.

   There is no inter process communication or synchronization of time implemented in this project, therefore, the scheduler does not know the exact finish time of child processes. It uses its own time and expected execution time of child process to determine whether to do context switch.

2. Kernel Version

   Linux 4.14.25 x86_64

3. Comparison

   The finish order of all test cases seems to be the same as theoretical results. However, as mentioned above, the scheduler does not know the exact finish time of child processes and the time of two CPU are not synchronized, these could cause wrong results in some cases of round-robin policy. For example, a process run on CPU1 will finish right before the end of current time quantum, but the scheduler run on CPU0 thought it has used up the time quantum and set it to a lower priority.