OS Assignment-3 Design Document

Topic: SimpleScheduler: A Process Scheduler in C from Scratch

Team Members: Medha Kashyap (2022292)
Himanshu Raj (2022216)

Group Number: 10

GitHub Repository Link: https://github.com/rahi-senpai/OS

Members Contribution:

- Medha Kashyap: Starter code, documentation, design doc, error handling
- Himanshu Raj: Round robin implementation and advanced functionalities

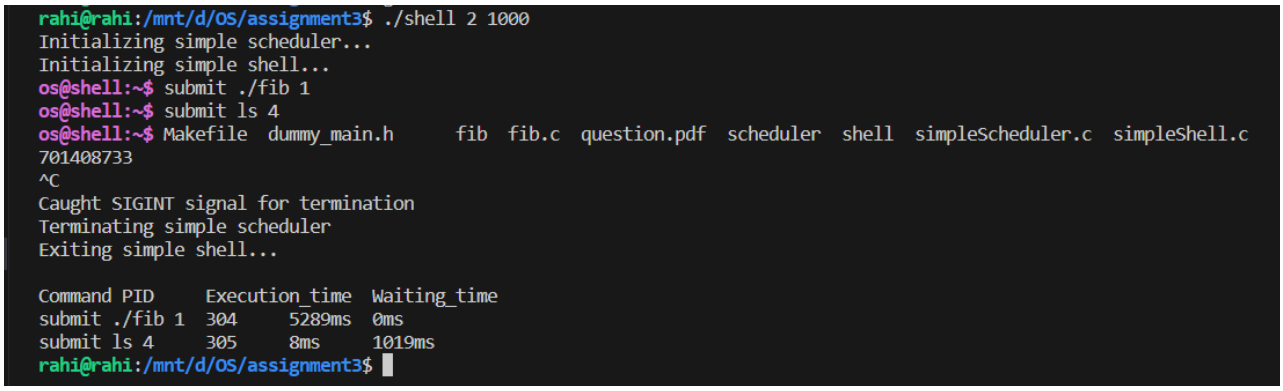
References: API man pages (SHM, semaphore, signal, kill APIs)
Lecture slides

Explanation:

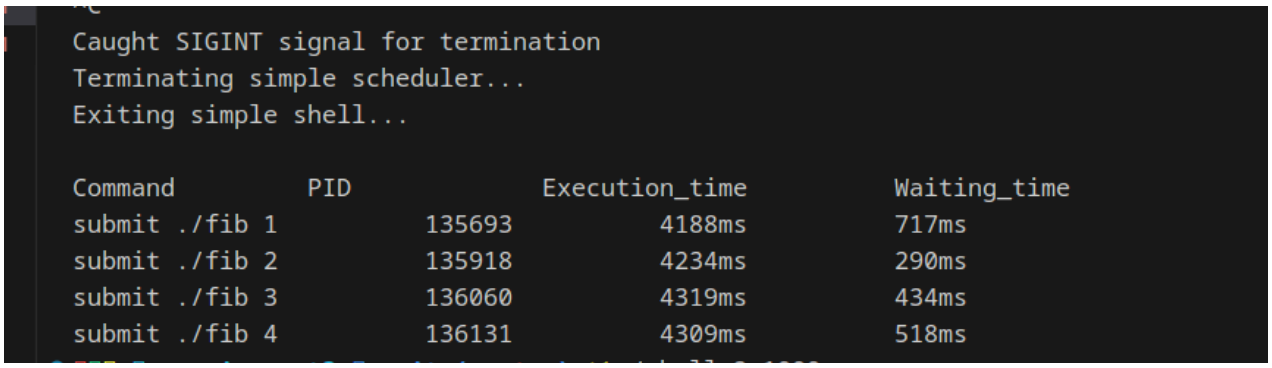
We have used shared memory to communicate between shell and scheduler processes. Scheduler is launched when you launch the shell. We have shared the history array (contains everything related to a process) between processes and used the kill API to send SIGCONT and SIGSTOP signals to processes with their PIDs after a time quantum (which is taken as input in milliseconds). Ready queue is a priority queue and running queue is a normal queue. For scheduling policy we have implemented a simple (naive) version of linux CFS, where we run a process from the ready queue till the specified tslice. We considered vruntime to be the comparing attribute and extract the processes with minimum vruntime to enqueue in the running queue and number of maximum processes in the running queue is also taken as input.

Execution time is the CPU burst time of a process. We have used sempahores every time we access shm so it can affect time due to sem_wait API.
We have not used dummy_main.h in our implementation.

Statistics (Conclusions):



This screenshot shows that a short job with least priority is executed before a long job with highest priority implying fair scheduling policy (no starvation).



This screenshot shows that resources are shared very fairly across processes in the ready queue as the wait time for the same processes with different priorities is almost the same, and execution time varies a little due to context switch overheads.