

Lab 2: Scheduler


Handed out Saturday Feb. 1, 2020

Due Saturday Feb. 15, 2020

Implement priority scheduling

In this assignment, you will change the scheduler from a simple round-robin to a priority scheduler. Add a priority value to each process (lets say taking a range between 0 to 31). The range does not matter, it is just a proof of concept. When scheduling from the ready list you will always schedule the highest priority thread/process first.

Add a system call to change the priority of a process. A process can change its priority at any time. If the priority becomes lower than any process on the ready list, you must switch to that process.

To get started, look at the file `proc.c`  look at scheduler, sched and allocproc methods in `proc.c`. Implement one of the next three items. If you implement more, each is a 5% bonus. To get credit for a bonus part, you must also develop a user test that will illustrate it.

- To avoid starvation, implement aging of priority. If a process waits increase its priority. When it runs, decrease it. (Possible Bonus 1)
- Implement priority donation/priority inheritance. (Possible Bonus 2)
- Add fields to track the scheduling performance of each process. These values should allow you to compute the turnaround time and wait time for each process. Add a system call to extract these values or alternatively print them out when the process exits. (Possible Bonus 3)

Goals: Understand how the scheduler works. Understand how to implement a scheduling policy and characterize its impact on performance. Understand priority inversion and a possible solution for it.