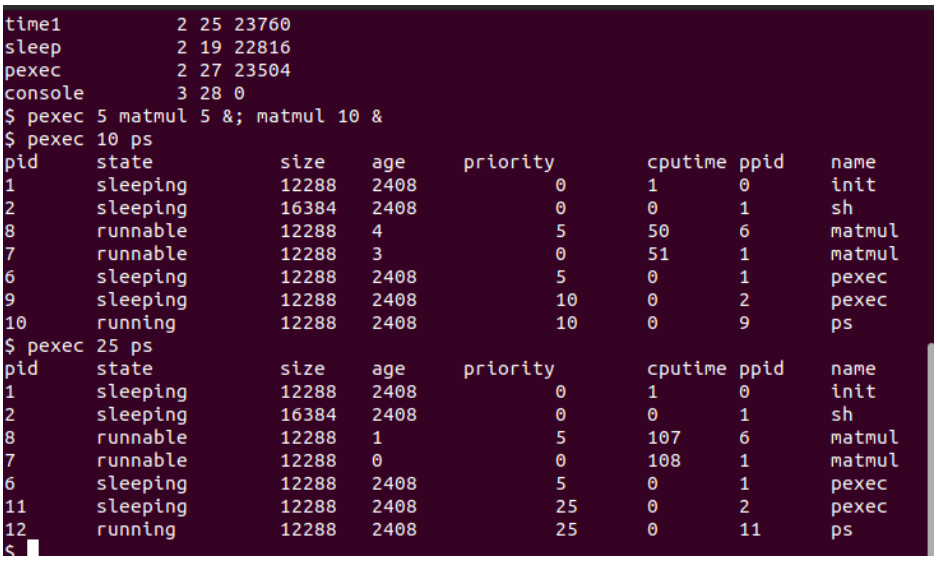
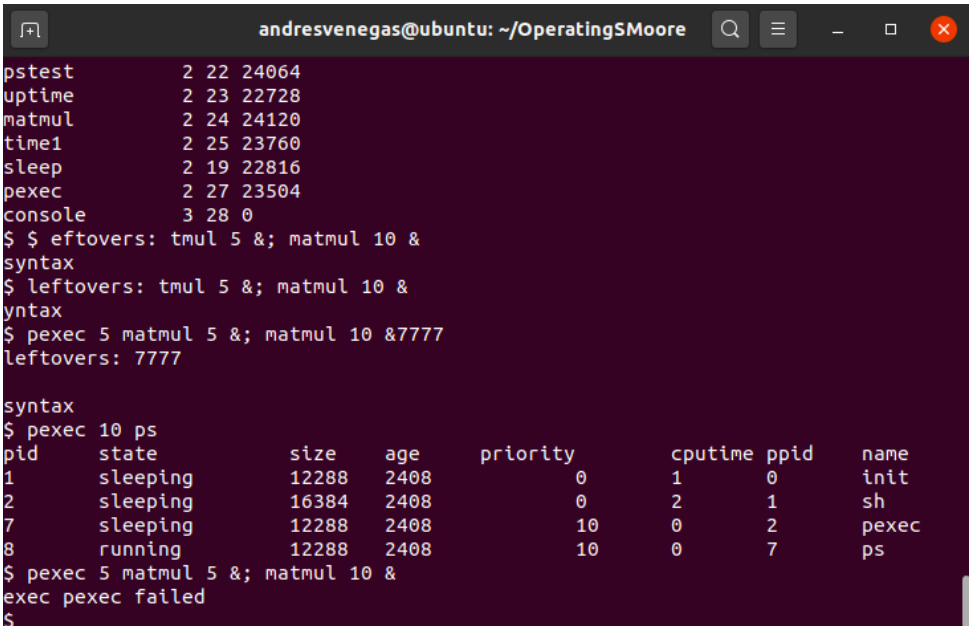
CS4375-13948 Fall 2023 Homework Report 3
<Javier Venegas> Submitted on
<10/20/2023>
javenegas8@miners.utep.edu
https://github.com/javenegas8/OperatingSMoore

HW 3: Priority-based Scheduler for xv6





Task 1. Modify the provided ps command to print the priority of each process.

To modify the getprocs() system call and its helper function procinfo() to include the new priority field for each process, and make changes to the kernel source code. The getprocs() system call and its helper function procinfo() will include the priority field in the process information structure, allowing to access process priorities using this system call.

Task 2. Add a readytime field to struct proc, initialize it correctly, and modify ps to print a process's age.

Explain how you calculate a process's age. Show the results of running your modified ps command.

To calculate a process's age, determine the time that has passed since the process entered the RUNNABLE state. The age is typically measured in milliseconds or some other time unit

Task 3. Implement a priority-based scheduler.

List the files and functions you changed and explain the purpose of each change.

param.h and proc.

Summarize what you learned by carrying out this task.

Describe any difficulties you ran into with this task and if/how you overcame them.

Overall, the task taught me the intricacies of kernel development and the importance of careful planning, thorough testing, and documentation when implementing new scheduling policies and other critical components in an operating system kernel. It highlighted the significance of debugging tools and techniques for identifying and resolving issues in kernel code.

Task 4. Add aging to your priority based scheduler.

Explain your aging policy. Show results from running your priority-based scheduler with aging. Your tests should show the benefits of using aging.

Update Priority:

Whenever you increment a process's priority as part of the aging policy, be sure to consider the priority range to prevent it from exceeding the maximum priority.

Summarize what you learned by carrying out this task.

Describe any difficulties you ran into with this task and if/how you overcame them.

In summary, this task allowed me to gain practical experience in enhancing scheduling policies by implementing aging and evaluating its impact. It highlighted the importance of fine-tuning policy parameters and balancing fairness with system efficiency, while also providing insights into debugging and testing kernel code.