CS4375-13948 Fall 2023
Geoff Waller
Gwaller@miners.utep.edu

Homework Report
Submitted on 10/18/2023

HW 3: Priority-based Scheduler for xv6

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

In order to get the system call to work, I did the following steps. First, I added a variable of integer type to proc.h. Then, I added setpriority and getpriority to syscall.h and user.h. After that, I changed the makefile to get the new users calls to work. In the proc.c, the procinfo function was changed in order to access priority. Procinfo.priority was set to a pointer to a struct process that then pointer to priority.

```
$ ps
Uptime: 20261
Current Priority: 0
pid     state           size    ppid    name    priority
1       sleeping        12288   0       init    0
2       sleeping        16384   1       sh      0
7       running         12288   2       ps      0
$ testpriority 4
new_priority 4
Before: 0
After: 4
$
```

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

For task 2,a readytime integer was added to struct proc in proc.h. From there proc.c was changed in scheduler method when the state was changed to "RUNNABLE". The algorithm I used was to make to compare a highest priority integer set at a negative value to a pointer to the priority from the current process. If that priority was higher, then the current priority would become the highest priority.

Unfortunately, I could not get this to run as well as I had hoped. I struggled to get my cputime accurate from the last assignment.

```
$ ps
Uptime: 19429
Current Priority: 0
pid     state           size    ppid    name    priority        cputime age
1       sleeping        12288   0       init    0               19              0
2       sleeping        16384   1       sh      0               13              0
13      running         12288   2       ps      0               0               0
$
```

Task 3. Implement a priority-based scheduler.
For task 3, I edited constants in param.h that would define a round robin schedule as a “0” and a priority scheduler as a “1.” My idea was to edit the makefile, to make an if check for the value of the scheduler. If that if check return “1”, my code in the scheduler method in proc.c would check for the highest priority and run that process.

Again, I did execute this section as well as I had hoped. I believe I was close to the correct strategy on how to achieve this but my implementation was not accurate. Tracing different methods through several files, as well as understanding pointers, continues to be difficult for me, however I am improving on each assignment.

With this task, even though my execution was implemented 100% correctly, I leaned the steps in how to switch processes using a priority based scheduler versus a round robin type.

```
$ pexec 5 matmul 5 &; matmul 10 &
$ pexec 10 ps
Uptime: 11393
Current Priority: 0
pid     state           size    ppid    name    priority        cputime age
1       sleeping        12288   0       init    0               19              0
2       sleeping        16384   1       sh      0               9               0
9       runnable        12288   7       matmul  0               0               -84213899
8       runnable        12288   1       matmul  0               1               -84213899
7       sleeping        12288   1       pexec   5               3               0
10      sleeping        12288   2       pexec   10              0               0
11      running         12288   10      ps      0               0               0
$ Time: 268 ticks
Time: 421 ticks
```

Task 4. Add aging to your priority based scheduler.
My aging policy did not work correctly. I used the same formula as above. I used the formula: int effective_priority = min(MAXEFFPRIORITY, p->priority + (p->cputime - p->readytime));. I attempted to use print statements to understand where my strategy when wrong in my aging formula but I was unable to figure it out.