CIS 415 Operating Systems

Assignment <2> Report Collection

Submitted to:

Prof. Allen Malony

Author:

*<John(Ziyuan) Zhou>*

**Report**

**Introduction**

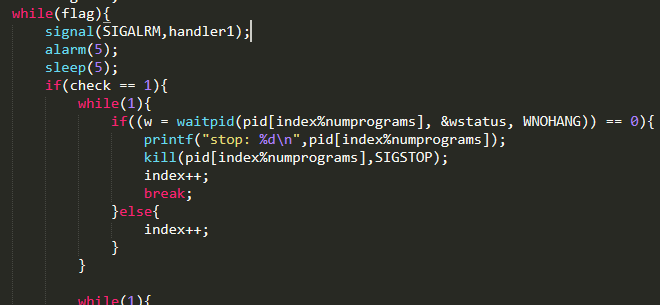
In this project, we mainly focus on how to implement the parent and child process in the MCP. We start building the MCP from simple version to a well-developed version. In the part one, we develop an MCP which can get all of the processes running together through parent and child processes. In the part two, we upgrade the MCP and let it able to stop and activate the running process by signal. However, it doesn’t have any scheduling policy. So, in the part three, we add the SIGALARM to equally share the processor by giving each process the same amount of time to run. In the part four, we let MCP able to show some relevant data about what system resources each workload process is consuming.

**Background**

In this project, I learn a lot about the signal. I have to understand how to implement signal and purpose of each signal types (such SIGUSER1, SIGSTOP, SIGCONT, SIGALARM). Furthermore, I have to spend lots of time to adjust my code because of the machine I am using. For instance, my friend and I setup the time of sleeping very differently. I have to let my program sleep at least 8 sec for catching the signal sometimes (You will see it in the part 2 and part 3). Because of this, I also spend some to comprehend sleep and found out it is very important to let your program able to catch the signal you send each time.

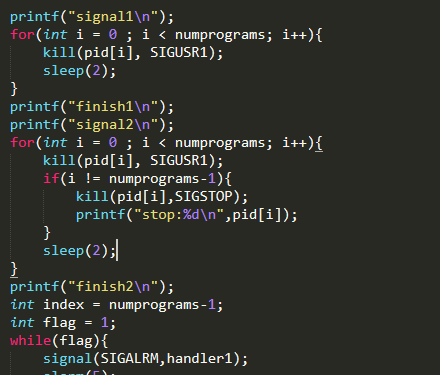
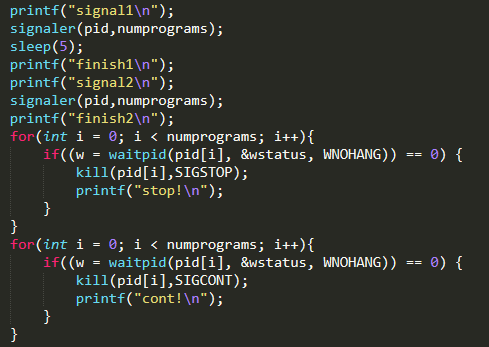
**Implementation**

I think the implementation is very straightforward; if you don’t understand the mechanism of the part you currently working, it is impossible to do the next part. We actually upgrade the MCP we build in the pervious part. Personally, I spend a lot of time in part 2 and part 3 because I had a hard time to understand how to implement handler for my signal. In the addition, I found some interesting thing in the part 3. I have to let my loop sleep some time as long as the time I setup for alarm because of the while loop. If I don’t do that, I will never enter my alarm handler because my loop will not wait for my alarm and the alarm will be reset each time of looping. Because of that, I wonder that is it necessary to have an alarm instead of just using sleep.



**Performance Results and Discussion**

So far, I didn’t find any big mistake with my code. I have some memory leak for my project, but won’t impact the execution. But there is one thing I worried about is the connection between SIGUSER1 and SIGSTOP. I just wonder is that possible my process will finish before I call the SIGSTOP to stop it, because I have sleep () after each signal call. Anyway, I think just call the SIGSTOP immediately after SIGUSER1 will not have any possibility to let this problem happen (This is what I did in part three).



**Conclusion**

Overall, I personally feel overload for my project because my classes and I never had a touch with signal. But fortunately, I finish the project on time with some memory leak. After I finish the project, I feel the man page is really my good friend. It helps me understand the structure of the signal and pid, and lets me know how to use the /proc quickly (just use stat, easy to implement).