CS 3841 – The Linux Scheduler

Name(s):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The purpose of this lab is to explore and familiarize yourself with the process scheduler in Linux. Explore the manual pages and/or the web for information. Be sure to document all resources used.

Submit your completed lab through Canvas by the due date. Include, in your submission, the source code AND build instructions for any program you write to complete this lab.

You can work with one other person from class on this lab. Make sure you put both names on your submission and notify Prof Lembke who you will be working with. Only one submission for the group is needed.

1. (10 Points) Describe the following Linux scheduling classes in your own words?  
     
   SCHED\_FIFO  
     
     
   SCHED\_RR  
     
     
   SCHED\_OTHER  
     
     
   SCHED\_BATCH  
     
     
   SCHED\_IDLE
2. (5 Points) How does the scheduling priority relate to the scheduling classes in Linux? Explain in turns of the following priority levels.  
     
   Priority Levels – 0 to 99  
     
     
     
     
     
   Priority Levels – 100 to 139
3. (10 Points) Describe the Linux “Completely Fair Scheduler” in your own words (you may have to search the web to find a description be sure to document your resource(s))? In your opinion, is it completely fair? Why or why not?
4. (10 Points) Prior to the “Completely Fair Scheduler”, Linux used an O(1) scheduler. How did this scheduler work? What advantage does the O(1) scheduler have over the “Completely Fair Scheduler” in terms of the scheduling overhead for a context switch?
5. (2 points) Write a program that is very CPU intensive. For example, a ‘for’ loop that increments a volatile variable many times. Run this program from the command line and view process information using ‘top’. What are the values of the following?  
     
   Priority:  
   Nice:  
   CPU Utilization (on average):
6. (3 Points) Run your program again, this time reduce the program priority by 10. Write the exact command you had to run to set the priority (Hint: man nice).   
     
     
   What are the values of the following?  
     
   Priority:  
   Nice:  
   CPU Utilization (on average):  
     
   Do the values make sense? Why or why not?
7. (5 Points) Run two instances of your program at the same time. The first instance run with default priority settings, the second increase the priority (you will have to run the second as root with ‘sudo’). Write the exact command you had to run to set the priority for the second process (Hint: man nice).  
     
     
   For each process, what are the values of the following?  
     
   Priority:  
   Nice:  
   CPU Utilization (on average):  
     
   Priority:  
   Nice:  
   CPU Utilization (on average):  
     
   Do the values make sense? Why or why not?
8. (10 Points) What real-time priority processes are running on your Linux VM? Find a description of these processes by doing some web searching (don’t forget to document your resources). What are these processes used for? Why are they run with real-time priority?
9. (10 Points) Besides the command line, you can change the process priority and scheduling properties using system calls. What system calls are used to get and set the processor preference (affinity) for a process. Give an example of when you would want to set the processor preference?
10. (10 Points) Linux lets you modify the settings of the scheduler through the /proc file system. This file system contains configuration files that when changed, modify the values stored within kernel space. For the scheduler, these files are located in /proc/sys/kernel and begin with sched\_  
      
    Search the web for information on the values stored in the following files (document your resources). Describe them in your own words. What are the values stored in these files on your Linux machine? Given what you learned, is it possible to really know what the time quantum for the “Completely Fair Scheduler” is?  
      
    sched\_latency\_ns  
      
      
      
      
      
      
    sched\_wakeup\_granularity\_ns  
      
      
      
      
      
      
    sched\_min\_granularity\_ns