Operating Systems - Monsoon 2021

Sambuddho Chakravarty, Arani Bhattacharya

November 29, 2021

Assignment 3 (Total points: 92.5)

Due: (Hard deadline: Dec. 13, 2021 @2359hrs; delay penalties apply)

1 Linux Scheduler (Total points: 42.5).

The first exercise requires you to change the Linux kernel scheduler in the following ways. For any process, there should be a system call to reduce (and NOT increase) the chances that the process would be selected for being dispatched. More specifically, let us assume that a certain process is chosen by the scheduler. The system call must pass on information, the additional delay (in msec) (that has been accepted from the userland) to the scheduler such that anytime the said process is chosen by the scheduler, it adjusts the *vruntime* of the said process so as to delay its selection.

For every process with the same nice value, you must log the timestamps corresponding to when each process is selected and dispatched to run. It must reflect the additional delay specified by the user. You may need to have a fresh look at how `vruntime` is calculated. This would help you in figuring out how `vruntime` could be adjusted to reflect the additionally requested latency.

What to submit/rubric.

1. Fully functional and modified kernel diff which can be patched to a stock kernel and used (preferably 5.10 upwards). The diff should correspond to the code that must include the system call implementation and the changes to the scheduler functions. Additionaly, you would also require a test program to test the system call and show the functionality (by printing timeslices given to the program each time scheduled). **[35 points]**. Full compilation and correct functionality demonstrated using the test program (35 points). Patched kernel compiles successfully but meets only a few functionality requirements (25 points). Patched kernel doesn't compile or compiles but doesn't boot up. But the functionality and changes presented seem valid and correct, though cannot be validated (12.5 points). The patched kernel doesn't compile or compiles but doesn't boot but the functionality doesn't seem correct, let alone be validated (0 points).

2. A readme describing the logic of the programs with relevant description of the data structures used, including description of the kernel data struc-

tures and source files modified (which ones were modified, how and why) [**7.5 points**].

2 Unix domain sockets IPC (flow-controlled) (Total points: 50).

The goal of this second exercise is to develop a better understanding of the Linux interprocess communication mechanisms. This would require you to write two programs P1 and P2. The first program P1 needs to generate an array of 50 random strings (of characters) of fixed length each. P1 then sends a group of **five** consecutive elements of the array of strings to P2 along with the ID's of the strings, where the ID is the index of the array corresponding to the string. The second program P2 needs to accept the received strings, and send back the **highest** ID received back to P1 to acknowledge the strings received. The program P2 simply prints the ID's and the strings on the console. On receiving the acknowledged packet, P1 sends the next five strings, with the string elements starting from the successor of the acknowledged ID.

The above mechanism needs to be implemented using three different techniques: (i) Unix domain sockets, (ii) FIFOs, and (iii) message passing queues. Please note that you may NOT make assumptions about the reliability of the interprocess communication mechanism, unless they are guaranteed by the mechanism itself.

What to submit/rubric.

1. Three variants of the program P1 (one each for communicating using Unix domain sockets, FIFOs and message passing queues respective) [**20 points**]. Full compilation and correct functionality of all the programs (20 points). Program compiles successfully but doesn't meet all the functionality requirements (15 points). Program doesn't compile, even if program logic seems apparently correct (0 points).

2. Three variants of the program P2 (one each for communicating using Unix domain sockets, FIFOs and message passing queues respective) [**20 points**]. Full compilation and correct functionality of all the programs (20 points). Program compiles successfully but doesn't meet all the functionality requirements (15 points). Program doesn't compile, even if program logic seems apparently correct (0 points).

3. A single Makefile to build each program [**5 points**].

4. A readme describing the logic of the programs with relevant description of the data structures used [**5 points**].