

If you have the difficulty in figuring out the coding for Phase 2, we recommend the following:

1) Determine the important parameters of your scheduler and the Linux Default Scheduler. We have discussed them already in the lab.

2) Determine what you have to do in the fork step and the periodic scheduling step. This is for determining the algorithm.

3) Linux Default Scheduler already determined its important parameters in `task_struct`. This structure is in the `sched.h` as you know. This structure is determined for storing the parameters of a process. If you need to determine any new parameter for the procedure of your scheduler, you should add this parameter at the end of this structure.

4) If the system need any initialization for a variable or parameter of a process, you should do it in `do_fork` function in `fork.c`. But be careful, you should do any initialization after line 715. In this code, `p` is the pointer representing the new process to be created in the system.

5) `schedule` function is called periodically by the system. This function is defined in `sched.c`. So, for the mechanism of the scheduler, you should define into this function. The periodic procedure of the Linux Scheduler is defined in this scheduler. So, if your scheduler has a periodic procedure, you should define into that function. But be careful, you are not to completely change the logic in this function. For your project, you will change only the process selection mechanism in the code. Additionally, you should preserve the code of the Linux Scheduler and so, you will separate the recent code of the default schedule and your schedule with `if/else` blocks. Selection criteria for the code blocks will be a simple control of a global variable. This variable is a flag and the value in it determines the scheduler which the system will run. You can only do this coding procedure for the selection mechanism of the scheduler. For further information, please read the Page 9 in the Testing Slides.

6) From starting of `repeat_schedule` label in the code, the default scheduler selects the next process to be run. As you know, the default scheduler has its own mechanism for selecting the process and we have discussed it in the labs. For further information, you can check the Linux Process Manager ebook. You can check the additional links file for the web pages prepared for the default scheduler.

7) In summary, Linux Scheduler selects the next process to be run in a mechanism which relies upon a global doubly linked list of the running processes. `list_for_each` macro is used for traversing the doubly linked list given as a argument to it. The loop block added in line 606 traverses the processes which are runnable and candidate for the next process. The selection criteria was discussed in the lab, as you know. The details for the process is given in the Linux Process Manager e-book as you know. `for_each_process` macro is used for traversing all of the process in the system, so default scheduler uses this macro when a new epoch is starting. We have discussed this in the lab, as you know. Again if you would desire additional information, you can check Linux Process Manager book and the slides prepared for the linux scheduler.

8) In the scheduler code, next represent the next process to run, so when you assign process to it, system selects this process to run. For a scheduler, it is crucial to determine

the next process according to the selection mechanism of it. Therefore, you should have a certain mechanism for selection of a process for your scheduler. For this, we would suggest you to determine the certain selection points of the algorithm of your scheduler. In the lab, we discussed the mechanism of the lottery scheduler in detail as you know.

Note: Any assignment, change in the value of the task\_struct of a process by the pointers will be persistent because system stores these structures globally.