

Problem Sheet #3

Problem 1

- a) Fairness in the context of the CFS is when each process receives equal amount of CPU processing power. If there are 2, 3, or how ever many processes, each process will receive the same amount of CPU processing power with each other.
- b) CFS scheduler selects task by using the data structure the red-black tree. It uses the fair clock and wait_runtime to keep all the processes in-line, in terms of the RBTree. Therefore, CFS selects tasks that are in the left-most because they are the most important (since the newly assigned tasks are put in to the far right of the tree).
- c) No, it doesn't use time-slices. Parameters that affect the CFS time calculations are the system-wide fair_clock variable (fair clock runs in fraction of a real time) and that each process has a waiting time (meaning when the CPU is currently running the task, it is the waiting time of the other processes that needs to run).
- d) Priorities (nice values) affect the selection of tasks since it is weight based on the static priority. For example, the task with a higher weight, or higher priority, will have a slower rate of time elapse, vice versa. This causes for the wait_runtime exhausting to be slower, which means the higher priority will get more CPU time.

Problem 2

The solution is within the zip file called running_group.c