Operating Systems
Jacobs University Bremen
Dragi Kamov

Course: CO20-320202
October 25th, 2018

Homework 3

Problem 3.1

Solution:

I based my answers on this questions on these two websites: sched-design-CFS.txt and Linux Journal

a) Fairness in the context of CFS means that the tasks get divided into categories and each group gets an equal part of the processor to process tasks.

b) CFS scheduler selects tasks based on a such called *p->se.vruntime* value and it always tries to run the task with the smallest *p->se.vruntime* value.
The data structure chosen for this CFS is rbtree and it is chosen because it stores the tasks according to their *p->se.vruntime* value and when it needs to run a task it takes the most left one and runs it for some period of time until another task becomes the most left task.

c) The CFS doesn't use time-slices and has no notion of them, although the previous scheduler did. The following parameters are affected by CFS time calculations:

- There is a wait time for which each process has to wait until the CPU is assigned that task.
- There is a *fair-clock* variable, which is basically what the CPU is defined to give to a certain process so that all processes have the same amount of processing. For example if there are 5 processes, each one would get 20% of the CPU power.

d) Each task gets a weight according to its priority. So in that way the time would pass faster for low-priority tasks. This means that low-priority tasks get less CPU time rather than high-priority tasks.

Problem 3.2

Solution:

The code for this problem can be found in *runner.c*.