

Problem Sheet¹ #3

Problem 3.1: *the effect of being nice*

(3+3 = 6 points)

Unix processes can lower their priority by setting their “nice” value (the idea is to be nice to others by lowering the priority of compute intensive processes, e.g., a simulation performed in the background).

- a) Write a program that forks several child processes. Every child process executes an endless computationally intensive loop with a different `nice()` value $n \in \{0..20\}$.
- b) Watch the execution of the program using a utility like `top` using a long update interval. What is the resulting CPU distribution among the processes? Provide a table with the data you have collected and a suitable plot. Provide relevant information about the hardware and software configuration you used for the experiment.

Problem 3.2: *Completely Fair Scheduler and Brain Fuck Scheduler*

(2+2 = 4 points)

With the Linux kernel version 2.6.23, the Completely Fair Scheduler (CFS) was introduced. Read about the design of the CFS scheduler and the general ideas behind fair scheduling. An alternative to CFS is the Brain Fuck Scheduler (BFS). Answer the following questions:

- a) Read the CFS scheduler document (contained in the Linux kernel source tree). What does fairness mean in the context of a scheduler? How does the CFS scheduler select tasks to run?
- b) Read the BFS scheduler documentation (contained in the BFS kernel patch). How does the BFS design differ from the CFS design?

¹See the [course web page](#) for submission instructions and grading details.