

## Project Proposal

### Group members:

Ellen Marie Andersen  
Benjamin Cathelineau  
Olexandr Matveyev

### Overview and deliverables:

The idea is to do research about the process scheduler algorithm of the current Linux Kernel 5.3.7 which is Completely Fair Scheduler (CFS), and to compare CFS to some different alternative algorithms which potentially could perform better than CFS or at least with the same performance. To be more specific we want to research and compare key criteria's for alternative algorithms, such as turnaround time, response time, deadlines, predictability, throughput, processor utilization, fairness, enforcing priorities, and balancing resources.

CFS algorithm is known as normal or non-real-time scheduler algorithm, and its time complexity is  $O(\log N)$ , where  $N$  is the number of tasks in the run-queue, and it is utilizing Red-Black Tree to maintain runnable processes. We want to find alternatives which is also non-real-time scheduler algorithms.

### Division of labor within the group:

Each of the team members will spend time learning more about the CFS algorithm and research a different scheduling algorithm and will compare it to the CFS algorithm. Once completed, we will present the options we found, or came up with as ideas, to each other and decide which ones we find the most interesting and suitable for this project.

When we will have this base established with the different algorithms we are going to include it into our project. We will look at different characteristics with each scheduling algorithm and using this to find interesting angles to look at when it comes to comparing them and discussing their advantages and disadvantages. The last part described will be both a joint and individual effort.

### Rough schedule:

We aim at finishing, so we are ready to present during the first week of presentations.

### Desired outcome:

From this project we hope to gain an even better and deeper understanding of scheduling algorithms, and CFS in particular. By studying a variety of scheduling algorithms, getting to know them and how they work, and looking at them in context of the other algorithms we are studying, we hope to observe interesting differences, advantages and disadvantages.