

OS Assignment 3

Aryan GD Singh
2019459

Implementation

The modified CFS scheduler was implemented on linux 5.9.1

We added a soft real time requirement to a process using `rtnice` system call written by us. This ensures that the process receives a certain amount of time slice to execute before other processes. For this purpose the following were done :-

1. Added a field `rt_nice` to `sched_entity` and initialise it to 0.
2. Added a syscall `rtnice` which takes `pid` and `rtval` as input and sets `rt_nice` for that process.
3. Modified the scheduling algorithm to first check `rt_nice` values and give priority based on that.
4. If multiple processes have positive `rt_nice` value, then the one with higher value is given priority.

The scheduling algorithm was implemented by modifying 2 methods in the `kernel/sched/fair.c` file

- `entity_before()` : We check if any of the 2 processes have positive `rt_nice` value. If so, we select the one with the higher `rt_nice` value. Otherwise, selection is done using the `vruntime` values.
- `update_curr()` : We update the `rt_nice` value of the executing process. It is decreased by `delta_exec`, which is the time for which process executed. If `rt_nice` value is 0, then we update the `vruntime` value as normal.

Testing our Scheduler

To test the functionality of the modified scheduler we have a file test.c, in which we fork a child process and call the syscall on the parent process.

The user has to input the rt_nice value to be set for the process(which is multiplied by 10^9).

The parent process runs a loop of $2 \cdot 10^8$ while the child process runs a loop of 10^8 , so normally the child process should finish executing first, but we will see that the parent process finishes first as it has been given priority.

Increasing the input rt_nice value will decrease the execution time of the process as it means the process is given priority for a larger time slice.

Error Handling

If the pid or rtval are < 0 , then the syscall returns -1 and sets errno to EIN, and test.c displays the error msg "invalid input".

If a process with the given pid does not exist then syscall returns -1 and sets errno to ESRCH, and test.c displays the error msg "no such process".

Both these errors are also logged to kernel by the syscall.