CSE231: Operating Systems

Assignment 3

Name: Mihir Chaturvedi
Roll no.: 2019061
Section: B
Branch: CSE

In this assignment, we are required to modify the inbuilt CFS scheduler of the Linux kernel and allow for real-time guarantees to take precedence over the default selection method (using the RB-tree and vruntime). Following are the list of modifications that have been implemented.

Addition of `rtnice` Attribute

This is a new attribute of the sched_entity attribute of a task/process that will provide information of the real-time guarantee required by the process. We add this in the new attribute sched_entity struct, and initialize the value of rtnice for a forked process to 0 in the __sched_fork() function.

Changes to the Scheduler

Here, the main changes are made to the selection of the process to next execute. In the entity_before() function, we prioritize the sched_entity that has a lower rtnice value, and return a binary 1 or 0 depending on the comparison. This check is done before comparing the vruntimes, and is only done if either sched_entities have a positive rtnice value.

Next, we need to update the rtnice value of the sched_entity once it has run on the CPU. We do this in the update_curr() function. We reduce the rtnice value of the ‘curr’ent entity by `delta_exec`, the time that the process ran on the CPU, and equate it to zero if `delta_exec` is greater than the rtnice value. Here, we also increase the vruntime of the of the entity, since once it’s back on the RB-tree (if at all) with a zero-rtnice value, then the

comparisons will be made using vruntime, which should be accurately reflected to ensure fairness.

Addition of System Call

To modify the rtnice value of the sched_entity of a process from the userspace, we must also create a system call that does so. We create a new system call, name it ‘rtnice_mod’, and assign it the numerical ID 440.

The system call accepts two inputs/arguments from the user: the PID for the process, and the new rtnice value (in seconds) for the process. Since the rtnice attribute of the sched_entity is an unsigned long long that holds the values as nanoseconds, we convert the user-inputted seconds into nanoseconds.

Testing

To test this, I have created a file that first forks the parent process into ‘n’ child processes. Each child process has a default rtnice value of 0 is made to do a long computation, after which it outputs the time taken.

Then, we fork the parent process ‘n’ times again, but give each new child process a positive, increasing rtnice value. The time taken to complete the computation is printed.

Error Handling

- **EINVAL:** This error will be returned when an invalid PID number has been passed, or a negative rtnice value.
- **ESRCH:** This error will be returned when the requested PID does not correspond to any currently running process, and so the `task_struct` for the requested PID is NULL.

Sample

The following is a sample out of the test file:

```
rtnice value step-size: 100
Number of forks/children: 5
With rtnice=0:
Child process 1; rtnice=0; duration=3.327755 seconds
Child process 3; rtnice=0; duration=3.675423 seconds
Child process 2; rtnice=0; duration=3.743091 seconds
Child process 5; rtnice=0; duration=3.744749 seconds
Child process 4; rtnice=0; duration=3.783028 seconds
With increasing rtnice:
Child process 2; rtnice=100; duration=2.979624 seconds
Child process 5; rtnice=400; duration=2.984560 seconds
Child process 3; rtnice=200; duration=2.985612 seconds
Child process 4; rtnice=300; duration=2.979846 seconds
Child process 1; rtnice=0; duration=5.641314 seconds
```