ASSIGNMENT 3 - Modifying the CFS scheduler

Ayush Mahant
2019353
Section A

Code (test .c) -

Using omp_get_wtime() we get elapsed wall clock time in seconds. The time is measured per thread. We need to include the omp.h package.
To show the difference between using soft real time guarantee and not using it, I have used two while loops.
First while loop for with the soft real time guarantee. For each loop a child process is created and it's corresponding parent's rtnice is increased using the syscall call rtnice.Here every loop or every child created have a specific soft real time guarantee given.
Then we print each process' execution time.
Second while loop for without the soft real time guarantee. For each loop a child process is created
Then we print each process' execution time.

Code(rtnice.c)

If the pid is wrong then an error is send else we search for the pid in the task-struct
Correspondingly, we then intiatize the rtnice to the soft-real guarantee provided.
If the soft-real guarantee is less than 0, error is send.

Changes made-

These changes are made in the linux-5.9.1/kernel/sched/fair.c
  1.) entity_before - We compare the rt_nice of inputted sched_entities. When both are zero, then comparison is according to the vruntimes.
  2.)update_curr - Now we check if rt_nice of any sched_entity >0 then the rt_nice gets updated rather than the vruntime.
  We have also intitialized the rt_nice in the linux-3.19_new/kernel/sched/core.c.

   INPUT-

   I have hardcored, the inputs which can be changed.

We need to provide the pid, which is generated using getpid() after fork.

Also, along with it, the soft real time guarantee is needed.

OUTPUT-

The execution time of the processes are stated.

First for with the soft real time guarantee, the execution time is stated which should be less than for without the soft real time guarantee in most of the cases.

SAMPLE TEST CASE OR SCRIPT-

WITH THE SOFT REAL GUARANTEE-

Pid is already generated, if to specify otherwise for error we can give it as <0.

Also, for the 4 guarantees, I have provided one as -1 to show error.

Others are provided in the screenshot.

NOTE: The values can be changed in the file.

ERROR HANDLING-

1)When the pid <0 an error is shown.

2) when the soft real time guarantee is <0 error is shown.

SCREENSHOTS-

ayushmahant@ayushmahant-VirtualBox: ~/Desktop

```
ayushmahant@ayushmahant-VirtualBox:~$ cd Desktop
ayushmahant@ayushmahant-VirtualBox:~/Desktop$ subl Makefile
ayushmahant@ayushmahant-VirtualBox:~/Desktop$ make
gcc test.c -lgomp
ayushmahant@ayushmahant-VirtualBox:~/Desktop$ make run
gcc test.c -lgomp
./a.out
it is variable so doesnot always give the correct process time because of other
 factorsWhen soft real time is given-------
ERROR is reported-- (INVALID DETAIL PROVIDED)  -1
ERROR is reported-- (INVALID DETAIL PROVIDED)  -1
Process 3: 10.961275 sec
Process 2: 11.005579 sec
When soft real time is        NOT        given-------
Processes ran in seconds is 0: 12.404449
Processes ran in seconds is 1: 12.406941
Processes ran in seconds is 3: 12.526272
Processes ran in seconds is 2: 12.615232
ayushmahant@ayushmahant-VirtualBox:~/Desktop$
```

ayushmahant@ayushmahant-VirtualBox: ~/Desktop

```
./a.out
it is variable so doesnot always give the correct process time because of other
 factorsWhen soft real time is given-------
ERROR is reported-- (INVALID DETAIL PROVIDED)  -1
ERROR is reported-- (INVALID DETAIL PROVIDED)  -1
Process 3: 10.961275 sec
Process 2: 11.005579 sec
When soft real time is        NOT        given-------
Processes ran in seconds is 0: 12.404449
Processes ran in seconds is 1: 12.406941
Processes ran in seconds is 3: 12.526272
Processes ran in seconds is 2: 12.615232
ayushmahant@ayushmahant-VirtualBox:~/Desktop$ make
gcc test.c -lgomp
ayushmahant@ayushmahant-VirtualBox:~/Desktop$ make run
gcc test.c -lgomp
./a.out
it is variable so doesnot always give the correct process time because of other
 factorsWhen soft real time is given-------
ERROR is reported-- (INVALID DETAIL PROVIDED)  -1
ERROR is reported-- (INVALID DETAIL PROVIDED)  -1
Process 2: 11.000823 sec
Process 3: 11.132616 sec
When soft real time is        NOT        given-------
Processes ran in seconds is 0: 12.444231
Processes ran in seconds is 3: 12.519828
Processes ran in seconds is 2: 12.522264
Processes ran in seconds is 1: 12.566796
ayushmahant@ayushmahant-VirtualBox:~/Desktop$
```