Rupanshoo Saxena | 2019096 | CSE

Operating Systems - Assignment 3
Date: 1st December 2020 (used the late submission policy with A3)

Modifying CFS Scheduler

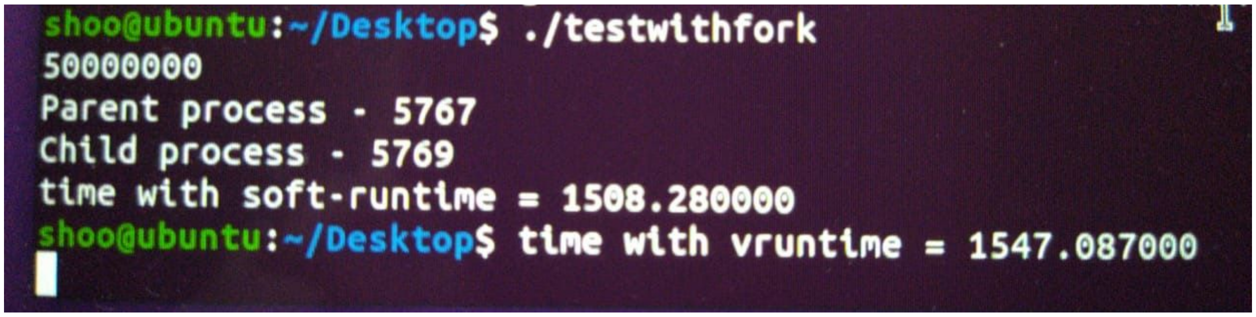
- Changes made:
    1. Added the name of the syscall - rtnice to the syscall table.
    2. Made a key called ‘rtnice’ in struct task_struct in sched.h
    3. Declared ‘rtnice’ syscall in syscalls.h
    4. Assigned the value of rtnice to 0 for a task_struct in core.c
    5. Added a few conditions in fair.c so that the process with soft real time is given the priority
    6. Created a folder for the ‘rtnice’ system call and added this folder to the kernel’s makefile.
    7. Created syscall in the rtnice folder, along with its makefile.

- Syscall rtnice
    - It takes 2 parameters, the process PID which is to be given soft real time, and the soft real time to be assigned to that PID.
    - It checks whether the soft real time provided is negative, if yes, then exits with an error that is printed in the kernel log message which can be viewed with the help of dmesg command.
    - It traversed over all the processes that are going on to search for the PID entered. If found, then soft real time of the task_struct is assigned.

- test.c
    - test.c file takes 1 input from the user - soft real time.
    - It performs a fork and assigns the soft real time to the parent process by calling the rtnice syscall and passing the PID of the parent process along with the soft real time taken as user input.
    - Both the parent and the child process have the same task to perform, a loop which goes from 9*10^8 to 0.
    - The program first displays the PID of both the parent and the child process and also displays the time taken by both the processes to finish the same task.
- Error messages are generated when:
    - Soft real time given as input is negative
    - The fork fails
    - PID is not found

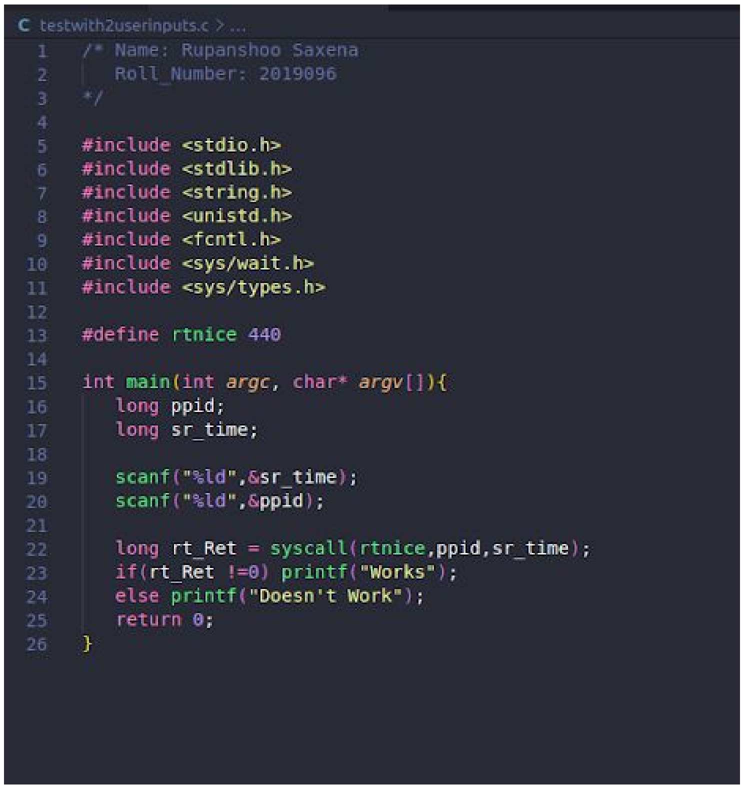
- Checking the system call:

1. Test.c file → using the fork method (as described above and an example is given below)



```
shoo@ubuntu:~/Desktop$ ./testwithfork
50000000
Parent process - 5767
Child process - 5769
time with soft-runtime = 1508.280000
shoo@ubuntu:~/Desktop$ time with vruntime = 1547.087000
```

2. To check with an ongoing process.
   This can be done in the following way:
   1) command shows all the ongoing procedures.
   2) For eg if we want to give our priority to firefox, we can look for the firefox process' PID using the above command.
   3) Use the following code to check. It takes two arguments - the process PID(in this example, firefox's PID) and soft real time to be allocated to it.



testwith2userinputs.c > ...

```c
/* Name: Rupanshoo Saxena
   Roll_Number: 2019096
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/wait.h>
#include <sys/types.h>

#define rtnice 440

int main(int argc, char* argv[]){
    long ppid;
    long sr_time;

    scanf("%ld",&sr_time);
    scanf("%ld",&ppid);

    long rt_Ret = syscall(rtnice,ppid,sr_time);
    if(rt_Ret !=0) printf("Works");
    else printf("Doesn't Work");
    return 0;
}
```

   4) This process prints "Works" in case the process is found and soft real time is allocated successfully.

5) To check whether the priority of the firefox process has changed or not, **top** command is used where we can see that more than 7/10 times, the firefox process is among the top 10 processes, meaning that its priority has increased.

Resources:

1. OS course slides and course material
2. https://josefbacik.github.io/kernel/scheduler/2017/07/14/scheduler-basics.html