Harman Singh

Roll No. : 2019042

OS Assignment 3 Writeup

Relationship: An attribute soft_rt is added to the struct sched entity, which represents the soft-real time requirements. The higher the requirements, more is the priority compared to other processes and hence will get executed first in the CPU with lesser time(for >50% of the times). Vruntime is compared later.

Changes Made:

1) In kernel/sys.c :

Added a new system call rtnice, using the macro SYSCALL_DEFINE2 which takes 2 arguments pid of the process and the time(requirements as input)
We run through all the processes (using for_each_process) and look fo the process with the given pid. The soft_rt (real time requirements required) are thus set to be equal to the time input.
Added new system call with number 441 in syscall_64.tbl

2)In include/linux/sched.h :

added another attribute u64 soft_rt in struct sched_entity.

3)In kernel/sched/core.c:

initialised p->se.soft_rt to 0.

4)In kernel/sched/fair.c

1st change is made in entity_before function where the process with more requirements is given higher priority. (comparison between 2 task_structs.

2nd change is made in the wakeup_preempt_entity function where softdiff parameter is declared which is equal to the difference between the soft_rt's of 2 given structs. If its value is >0 then -1 is returned(it is used in pick_next_entity and the value is returned accordingly to take the process with greater requirements).

3rd change is made in the update_curr function where the runtime statistics of the current task are updated. The soft_rt is subtracted by delta_exec or is made 0 accordingly. Other times, vruntime is considered.

Testing:

In test.c, fork() is used to clone the process and both the parent and child process undergo a certain program. However in the parent, the system call rtnice is called. The time taken by the parent is less than the time taken by the child(quite a no. of times) owing to the more requirements it has(taken as user input). This is the relationship I've used, more the requirements, more the process is in the gravest need of the CPU and hence should get executed first in lesser time.

In case the user wishes to give the pid for the parent process himself/herself, an option is provided wherein the user enters 1 if he/she wishes to give the input else 0 is given and the pid of the original parent process is used for the system call.

Errors Handled

-If pid/soft_requirements < 0 : -EINVAL is returned (invalid argument) is reported. Error No. 22 is reported.

-In case the process is not found, -ESRCH is returned(which means No such process). (This will be shown when the pid input is given by the user, for eg, which is greater than the allowed value). Error No. 3 Is reported.

-Error is handled when the child can't be forked in test.c

-Error is handled when system call is not executed properly in test.c