Problem 1 Description:

For problem 1 there are 5 parts for increasing the vruntime of a process

1. I have edited the syscall_64.tblfile present in the directory linux-5.14.3/arch/x86/entry/syscalls and added the name of the system call as delay_vruntime at the end i.e (line 448).
2. I have made a new variable of type u64 inside the struct sched_entity, which stores the delay_vruntime to further add the vruntime.
3. Inside the file core.c, there is function which initializes the sched_entity struct with default values, here i have added a line to make the default value of the variable declared in part 2 (i.e. delay_vruntime) as 0;
4. I have defined the system call delay_vruntime inside the file core.c which have two parameters of type pid_t(pid) and u64(delay in ms).
   Further to get the sched_entity of current process, i have used variables of type struct pid and struct task_struct; find_get_pid() and pid_task() functions are used to get the pid and task_struct of the process with the given pid respectively.
   Finally, when we have the task_struct there is a field of type sched_entity inside it which further contains the variable delay_runtime which is assigned as the delay passed by the user.
   At every step in the code there is error handling and corresponding error codes are returned.
5. Finally, the last change is in the function update_curr() inside the file fair.c present in the directory linux-5.14.3/kernel/sched/fair.c.
   which function updates the vruntime after everytime a process is executed.
   In this i have added a line where vruntime variable in the sched entity of current process is updated by the delay_vruntime value (Since vruntime is stored in nanoseconds delay is multiplied by 10^6)

Test Code:

There are three files for testing:

1. e1.c: Contains a for loop for the sum of first 500000000 numbers and then prints the execution time using clock() function in time.c.
2. e2.c: Contains a for loop for the sum of first 500000000 numbers and then prints the execution time using clock() function in time.c, but there is a system call delay_vruntime by some integer.
3. test.c: forks a child process and one process runs e1.c and the second runs e2.c using exec family system calls.

Nakul Thureja

2020528