Operating Systems: Assignment-2
Harshil Mital (2021050)
November 2022

1. Changing Scheduling Priority and Policy for Threads and Processes

1.1 `make main`

3 threads namely Thr_A, Thr_B and Thr_C count from 1 to 2^{32} by calling their respective functions which execute while loops.

The scheduling policies and priorities of these threads are changed by assigning these properties to the attributes of the respective threads before their creation.

This is done using the following functions:

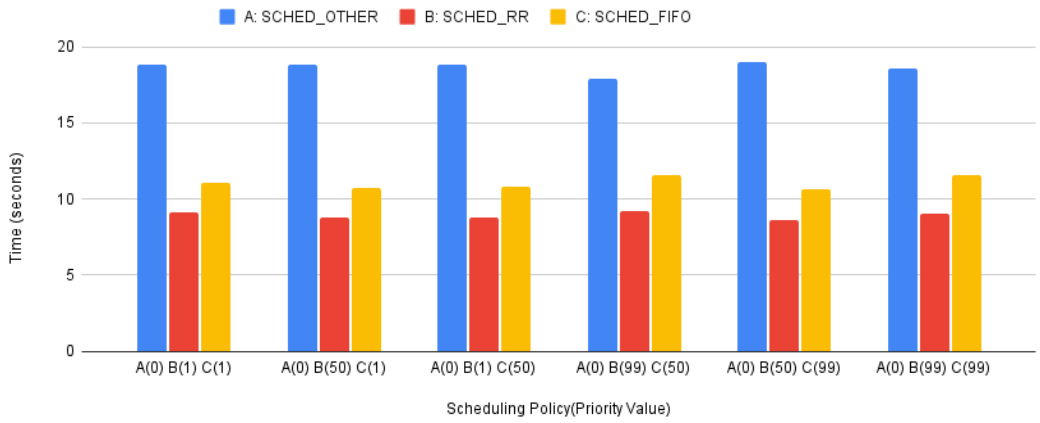
`pthread_attr_init()`: to initialize the `attr` object

`pthread_attr_setinheritsched()`: to change the inhert scheduler from `PTHREAD_INHERIT_SCHED` to `PTHREAD_EXPLICIT_SCHED`

`pthread_attr_setschedpolicy()`: to change the scheduling policy from `SCHED_OTHER` to `SCHED_RR` and `SCHED_FIFO`

`pthread_attr_setschedparam()`: to set the changed `sched_priority` of `struct param` to `attr`

These are the timings achieved after running the threads for different priority values:



Since `SCHED_RR` and `SCHED_FIFO` are real-time processes, they are significantly faster than `SCHED_OTHER` which makes use of CFS policy.

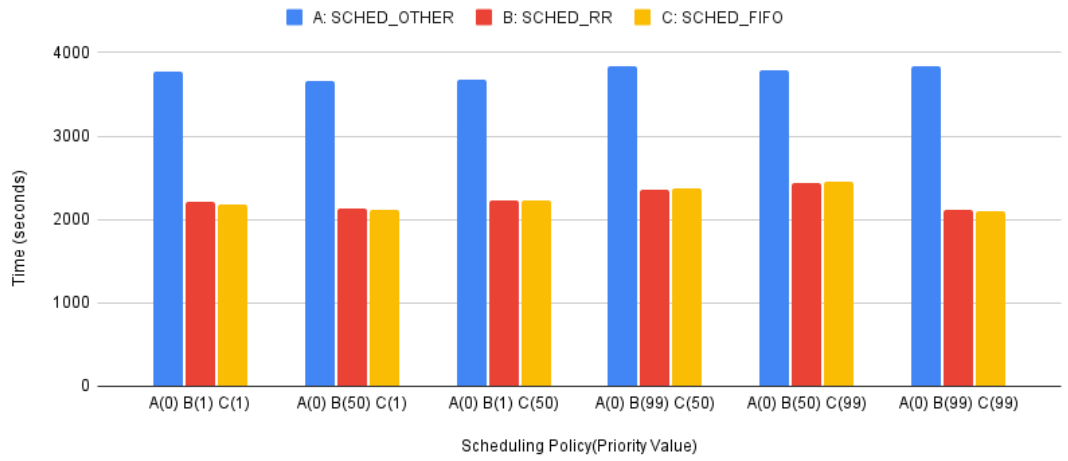
1.2 `make kern`

3 child processes of a parent process simultaneously compile copies of a custom linux kernel.

The child processes are created using `fork()` system call and `execvp()` is used to execute the bash script to start the compilation process.

To change the scheduling policy and priority of the processes `sched_setscheduler()` was used.

These are the timings achieved after running the processes for different priority values:



Since `SCHED_RR` and `SCHED_FIFO` are real-time processes, they are significantly faster than `SCHED_OTHER` which makes use of CFS policy.

2. Adding a custom system call (`kernel_2d_memcpy`)

Firstly the following line is added to `syscall_64.tbl` in the path `/root/new_kernel/linux-5.19.9/arch/x86/entry/syscalls:`

```
------
450 common  set_mempolicy_home_node sys_set_mempolicy_home_node
451 common  kernel_2d_memcpy  sys_kernel_2d_memcpy
------
```

Then a directory `kernel_2d_syscall` is created in `/root/new_kernel/linux-5.19.9`

In this 2 files are created: `kernel_2d_syscall.c` and `Makefile`

`kenrel_2d_syscall.c` uses the system calls `copy_from_user()` and `copy_to_user()` to copy a 2 dimensional floating point matrix from user space to kernel space and then back to user space.

Then the kernel is recompiled and the system call is tested by `kernel_2d_memcpy_test.c` in `/home/harshil/OS/OS_Assignment_2/Q2`

`diff` is used to generate a PatchFile which can be used to add the system call to a stock kernel.