Operating Systems–II: CS3523
Spring 2020
Lab Assignment 3: Linux Scheduler
Last date for submission: N/A

Write a C program to illustrate various scheduler options available in default (aka stock) Linux kernel. On a system with 'n' logical processors, create a team of 'n+1' processes which has one leader and 'n' members.

Command line syntax of the program must be:

```
./99_sched_test --leader_policy <sched_policy>  --leader_priority <sched_priority>
                --member_policy <sched_policy>  --member_priority <sched_priority>
                     Where <sched_policy> can be other|batch|idle|fifo|rr
                          <sched_priority> can be 0 to 99.
                     Default <sched_policy> is 'other', <sched_priority> is 0
```

Using sched_setscheduler(2) set policy and priority of leader and member processes as per given command line arguments.

The logic of the main function in the program must be:

1. Validates scheduling normal policy (other or batch or idle) or real-time policy (fifo or rr).
2. Find np = number of CPUs using get_nprocs(3).
3. Create np child process for members, each of them run a CPU bound function "sched_func()" with respective command line inputs.
4. Create one more child process for the leader which also runs function "sched_func(int policy, int priority, int leader)" with respective command line inputs.

The logic of sched_func(int policy, int priority, int leader) function must be:

1. If leader = true, Start a timer
2. Iterate for a fixed number of iterations (about 5 billion. Your system may required lesser iterations)
3. For each iteration call random() and assign value to an array of 1,000,000 in a circular fashion.
   Ensure that code is not sleepy, to prevent voluntary context switches.
4. If leader = true, End timer and calculate elapsed time.
5. If leader = true, Print number of page faults, context switches (voluntary and involuntary).

Test your code with leader's policy to be SCHED_OTHER, SCHED_IDLE, SCHED_BATCH, SCHED_FIFO, SCHED_RR as shown below

```
$ sudo ./sched_test --leader_policy other --leader_priority 0
Leader:: policy:OTHER priority:0
leader:: Number of page faults = 0
leader:: Number of voluntary switches = 0
```

```
leader:: Number of involuntary switches = 1794
leader:: Time taken: 71120921 us

$ sudo ./sched_test --leader_policy idle --leader_priority 0
Leader:: policy:IDLE priority:0
leader:: Number of page faults = 0
leader:: Number of voluntary switches = 0
leader:: Number of involuntary switches = 281
leader:: Time taken: 97110019 us

$ sudo ./sched_test --leader_policy batch --leader_priority 0
Leader:: policy:BATCH priority:0
leader:: Number of page faults = 0
leader:: Number of voluntary switches = 0
leader:: Number of involuntary switches = 1329
leader:: Time taken: 71963047 us

$ sudo ./sched_test --leader_policy fifo --leader_priority 99
Leader:: policy:FIFO priority:99
leader:: Number of page faults = 0
leader:: Number of voluntary switches = 0
leader:: Number of involuntary switches = 16
leader:: Time taken: 65783364 us

$ sudo ./sched_test --leader_policy rr --leader_priority 99
Leader:: policy:RR priority:99
leader:: Number of page faults = 0
leader:: Number of voluntary switches = 0
leader:: Number of involuntary switches = 11
leader:: Time taken: 41746693 us

$ sudo ./99_sched_test --leader_policy fifo --leader_priority 99 --member_policy fifo
--member_priority 50
Leader:: policy:FIFO priority:99
leader:: Number of page faults = 0
leader:: Number of voluntary switches = 0
leader:: Number of involuntary switches = 88
leader:: Time taken: 68887780 us
```

References:

1. http://man7.org/linux/man-pages/man2/sched_setscheduler.2.html
2. http://man7.org/linux/man-pages/man2/sched_getscheduler.2.html
3. http://man7.org/linux/man-pages/man7/sched.7.html
4. https://linux.die.net/man/3/get_nprocs