

Process Scheduler Modification

Objective: In this lab you will build a simple LKM called “sched_dumper” that will search for a process with a specific name “setsched” and modify its scheduling policy and priority. In addition, you will complete code for the “setsched” usermode program from the code snippet provided below.



File(s) for this lab:

1. The “setsched” program should set its own process priority in usermode (as root) and scheduling policy as soon as it executes.

```
user@localhost Lab61$ sudo ./setsched
Scheduler set to SCHED_FIFO with priority 1...
PID 30289 sleeping..
sched_getscheduler() = SCHED_FIFO
Process priority 1...
PID 30289 sleeping..
sched_getscheduler() = SCHED_FIFO
Process priority 1...
PID 30289 sleeping..
```

2. Your LKM should search for this process and modify its scheduling policy to SCHED_RR and set a process priority of -20 (user mode). Once the scheduling policy and priority is modified by the LKM, the user mode process “setsched” should reflect the change and automatically update the result.

```
Process priority 1...
PID 30289 sleeping..
sched_getscheduler() = SCHED_FIFO
Process priority 1...
PID 30289 sleeping..
sched_getscheduler() = SCHED_FIFO
Process priority 1...
PID 30289 sleeping..
sched_getscheduler() = SCHED_RR
Process priority -20...
PID 30289 sleeping..
sched_getscheduler() = SCHED_RR
```

Hints:

- Examine sched.h for the correct scheduler policy mapping and remember that usermode uses a different range (-20 – 20) compared to kernel mode for priority
- Create a new project “Lab6” and import the files in the “LKI/Lab6” folder to get started.
- Use the following code sequence to get started with the usermode “setsched” program which is also provided in your “Lab6” folder.
- Call “setpriority()” and “sched_setscheduler()” from user mode, use man page for args and information on their usage. Remember, **setsched** is a usermode program!

PROCESS SCHEDULER MODIFICATION

```
#include <stdio.h>
#include <unistd.h>
#include <linux/sched.h>
#include <sys/resource.h>

/*
 * Scheduling policies defined in sched.h
#define SCHED_NORMAL    0
#define SCHED_FIFO      1
#define SCHED_RR        2
#define SCHED_BATCH     3
 */

struct sched_param {
    int sched_priority;
};

struct sched_param s_param;
void setscheduler(void)
{
    /* CALL sched_setscheduler() to set a process scheduling policy and priority */
}

int main()
{
    setscheduler();

    while(1)
    {
        printf("PID %i sleeping...\n", getpid());
        switch(sched_getscheduler(getpid()))
        {
            case SCHED_NORMAL: printf("sched_getscheduler() = SCHED_NORMAL\n");
                               break;
            case SCHED_FIFO: printf("sched_getscheduler() = SCHED_FIFO\n");
                               break;
            case SCHED_RR: printf("sched_getscheduler() = SCHED_RR\n");
                             break;
            case SCHED_BATCH: printf("sched_getscheduler() = SCHED_BATCH\n");
                               break;
            default:
                break;
        }
        printf("Process priority %i...\n", getpriority(PRIO_PROCESS, 0));
        sleep(5);
    }
}
```