

# ECE4730J

## Advanced Embedded System

### LAB 3: REAL-TIME SCHEDULING

October 23, 2021

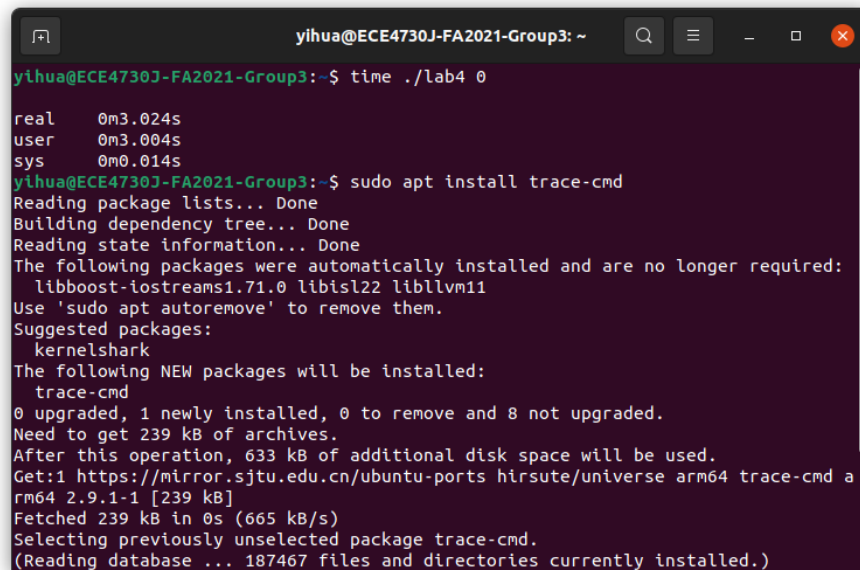
Name	Student ID
Yihua Liu	518021910998
Shuocheng Chen	517021911139
Yiming Ju	518370910059

---

1. Please show the output from running your new program (compiled without any optimizations) within the time command.

```
1  #define _GNU_SOURCE
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <sched.h>
5
6  int main(int argc, char **argv) {
7      int n = atoi(argv[1]); // 0 <= n <= 3
8      size_t x;
9      // scanf("%d", &n);
10     cpu_set_t set;
11     CPU_ZERO(&set);
12     CPU_SET(n, &set);
13     sched_setaffinity(0, sizeof(cpu_set_t), &set);
14     for (size_t i = 0; i <= 500000000; ++i) {
15         x = (i + 1) * 2;
16     }
17     return 0;
18 }
```

```
gcc -o lab3-1 lab3-1.c
time ./lab3-1 0
```

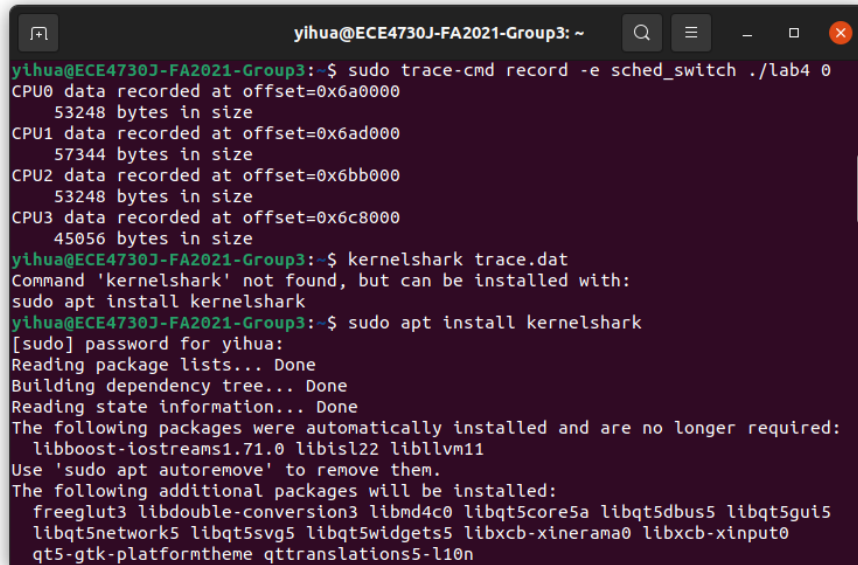


```
yihua@ECE4730J-FA2021-Group3: ~  
yihua@ECE4730J-FA2021-Group3:~$ time ./lab4 0  
real    0m3.024s  
user    0m3.004s  
sys     0m0.014s  
yihua@ECE4730J-FA2021-Group3:~$ sudo apt install trace-cmd  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libboost-iostreams1.71.0 libisl22 libllvm11  
Use 'sudo apt autoremove' to remove them.  
Suggested packages:  
  kernelshark  
The following NEW packages will be installed:  
  trace-cmd  
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.  
Need to get 239 kB of archives.  
After this operation, 633 kB of additional disk space will be used.  
Get:1 https://mirror.sjtu.edu.cn/ubuntu-ports hirsute/universe arm64 trace-cmd a  
rm64 2.9.1-1 [239 kB]  
Fetched 239 kB in 0s (665 kB/s)  
Selecting previously unselected package trace-cmd.  
(Reading database ... 187467 files and directories currently installed.)
```

Figure 1. The output from our new program.

2. Please name three processes that interfered with the execution of your program on that core, and explain briefly how you know they did that based on the trace you examined.

```
sudo apt install trace-cmd  
sudo trace-cmd record -e sched_switch ./lab3-1 0  
sudo apt install kernelshark  
kernelshark trace.dat
```

A terminal window with a dark purple background and white text. The window title is 'yihua@ECE4730J-FA2021-Group3: ~'. The user enters the command 'sudo trace-cmd record -e sched\_switch ./lab4 0'. The output shows data recorded for CPU0, CPU1, CPU2, and CPU3, including offsets and sizes. Then, the user enters 'kernelshark trace.dat', which results in an error: 'Command 'kernelshark' not found, but can be installed with:'. The user then enters 'sudo apt install kernelshark'. The output shows the installation process, including package lists, dependency trees, and a list of additional packages to be installed.

```
yihua@ECE4730J-FA2021-Group3:~$ sudo trace-cmd record -e sched_switch ./lab4 0
CPU0 data recorded at offset=0x6a0000
53248 bytes in size
CPU1 data recorded at offset=0x6ad000
57344 bytes in size
CPU2 data recorded at offset=0x6bb000
53248 bytes in size
CPU3 data recorded at offset=0x6c8000
45056 bytes in size
yihua@ECE4730J-FA2021-Group3:~$ kernelshark trace.dat
Command 'kernelshark' not found, but can be installed with:
sudo apt install kernelshark
yihua@ECE4730J-FA2021-Group3:~$ sudo apt install kernelshark
[sudo] password for yihua:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libboost-iostreams1.71.0 libisl22 libllvm11
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
freeglut3 libdouble-conversion3 libmd4c0 libqt5core5a libqt5dbus5 libqt5gui5
libqt5network5 libqt5svg5 libqt5widgets5 libxcb-xinerama0 libxcb-xinput0
qt5-gtk-platformtheme qttranslations5-l10n
```

Figure 2. trace-cmd original\_track.dat.

- `gnome-system-mo`
- `kworker`
- `v3d_render`

There are also a switch to migration and a switch to swapper in the beginning and in the end.

They trigger events `sched_switch` from `lab3-1` to them or from them to `lab3-1` on the line of CPU 0, so we can know they interfered with the execution of our program on that core (core 0).

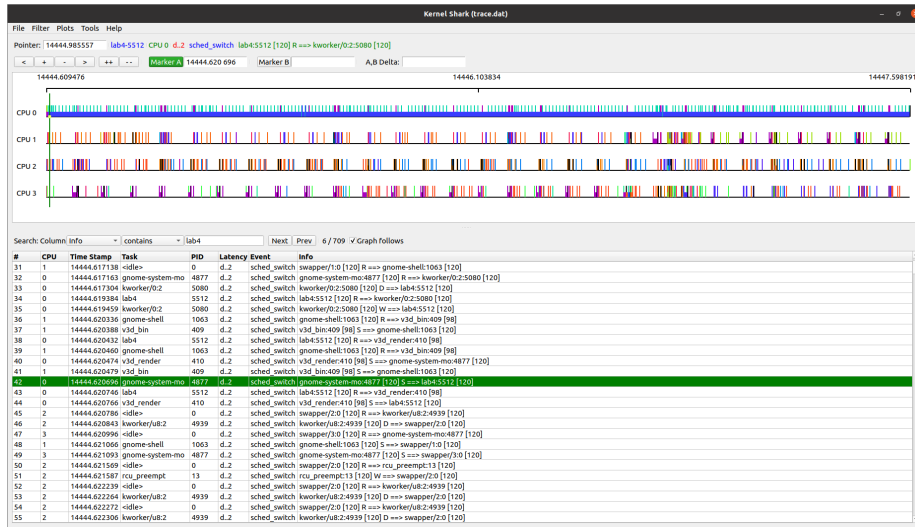


Figure 3. kernelshark original\_track.dat.

3. Find the largest number for which sched\_setscheduler() succeeds when run as root, and the largest number for which sched\_setscheduler() succeeds when run not as root, and report those values as the answer to this exercise.

```

1  #define _GNU_SOURCE
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <sched.h>
5
6  int main(int argc, char *argv[]) {
7      int n = atoi(argv[1]); // 0 <= n <= 3
8      if (n < 0 || n > 3) {
9          fprintf(stderr, "Error: Input CPU core number is
10             ↪ invalid.\n");
11             // Return value of int main() other than
12             ↪ EXIT_SUCCESS = 0 OR EXIT_FAILURE = 1
13             // in <stdlib.h> is not defined. Instructed by
14             ↪ Lab_3.pdf, we return a negative value -1.
15             return -1;
16     }
17     // sched_get_priority_min(SCHED_RR) <= rtpriority <=
18     ↪ sched_get_priority_max(SCHED_RR)
19     int rtpriority = atoi(argv[2]);
20     size_t x;
21     if (rtpriority < sched_get_priority_min(SCHED_RR) ||
22         ↪ rtpriority > sched_get_priority_max(SCHED_RR)) {
23         fprintf(stderr, "Error: Input priority is out of
24             ↪ range.\n");
25         return -1;
26     }
27 }

```

```

21     struct sched_param param;
22     param.sched_priority = rtpriority;
23     cpu_set_t set;
24     CPU_ZERO(&set);
25     CPU_SET(n, &set);
26     sched_setaffinity(0, sizeof(cpu_set_t), &set);
27     if (sched_setscheduler(0, SCHED_RR, &param)) {
28         perror("Error: Fail to set scheduler SCHED_RR\n");
29         return -1;
30     }
31     for (size_t i = 0; i <= 500000000; ++i) {
32         x = (i + 1) * 2;
33     }
34     return 0;
35 }

```

```

yihua@ECE4730J-FA2021-Group3: ~
yihua@ECE4730J-FA2021-Group3:~$ sudo ./lab3-2 0 99
[sudo] password for yihua:
sudo: ./lab3-2: command not found
yihua@ECE4730J-FA2021-Group3:~$ ls
Desktop    gcc-releases-gcc-8.3.0  Music      Public    Templates
Documents  lab3-2                  perl5      shutter   Videos
Downloads  lab3-2.c                Pictures    snap
yihua@ECE4730J-FA2021-Group3:~$ gcc -o lab3-2 lab3-2.c
yihua@ECE4730J-FA2021-Group3:~$ sudo ./lab3-2 0 99
yihua@ECE4730J-FA2021-Group3:~$ ./lab3-2 0 1
Error: Fail to set scheduler SCHED_RR
: Operation not permitted
yihua@ECE4730J-FA2021-Group3:~$

```

Figure 4. The largest number for which sched\_setscheduler() succeeds when run as root and not as root.

The largest number for which sched\_setscheduler() succeeds when run as root is 99. No matter what number is chosen, sched\_setscheduler() never succeeds when run not as root.

```

sudo trace-cmd record -e sched_switch ./lab3-2 0
kernelshark trace.dat

```

```

yihua@ECE4730J-FA2021-Group3: ~
yihua@ECE4730J-FA2021-Group3:~$ sudo trace-cmd record -e sched_switch ./lab3-2 0
1
CPU0 data recorded at offset=0x6a3000
8192 bytes in size
CPU1 data recorded at offset=0x6a5000
36864 bytes in size
CPU2 data recorded at offset=0x6ae000
24576 bytes in size
CPU3 data recorded at offset=0x6b4000
32768 bytes in size
yihua@ECE4730J-FA2021-Group3:~$ kernelshark trace.dat
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland
to run on Wayland anyway.
Loading "trace.dat"
yihua@ECE4730J-FA2021-Group3:~$ kernelshark trace.dat
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland
to run on Wayland anyway.
Loading "trace.dat"
qt.svg: /usr/share/icons/Yaru/16x16/legacy/help-about.svg:248: Could not resolve
property: #linearGradient1241
qt.svg: /usr/share/icons/Yaru/16x16/legacy/help-about.svg:248: Could not resolve
property: #linearGradient1241
qt.svg: /usr/share/icons/Yaru/16x16/legacy/help-about.svg:248: Could not resolve
property: #linearGradient1241

```

Figure 5. kernelshark trace.dat.

4. Please state whether any processes preempted your program (and if so which ones), and whether there are any meaningful differences in how these interruptions appear in this trace (kernelshark trace.dat), versus in your original (non-real-time) trace (kernelshark trace\_original.dat).



Figure 6. Processes that preempted our program.

- gmain
- nautilus

- <idle>
- tracker-miner-f
- ksoftirqd
- kworker
- brcmf\_wdog/mmc1
- migration
- tracker-store

Task lab3-2 first executed very shortly on CPU 2, then by migration->swapper->nautilus it is switched to CPU 0.

The interruptions of the trace appear much less than the original trace for all the 4 CPUs, especially for CPU 0, there are less sched\_switch in the trace than in the original trace. The price is the execution time of the real-time program is a bit longer than the non-real-time program [1].

5. Please state how many sched\_switch events were recorded on CPU core 0 where your program ran, and compare that number to the number of sched\_switch events were recorded on each of the other CPU cores.

By using Search:Column, we have the number of sched\_switch events recorded on each CPU:

CPU 0	64
CPU 1	485
CPU 2	314
CPU 3	476

Table 1. The number of sched\_switch events recorded on each CPU.

The number of sched\_switch events recorded on CPU 1, 2, and 3 are much more than those on CPU 0.

6. Please state (1) what range of real-time priorities you see being used, (2) what processes have real-time priorities, and (3) speculate why they may need to be run with real-time priority.





[migration/0]	99
[idle_inject/0]	50
[idle_inject/1]	50
[migration/1]	99
[idle_inject/2]	50
[migration/2]	99
[idle_inject/3]	50
[migration/3]	99
[watchdog]	50
[irq/69-aerdrv]	50
[irq/55-mmc0]	50
[v3d_bin]	1
[v3d_render]	1
[v3d_tfu]	1
[v3d_csd]	1
[v3d_cache_clean]	1
[irq/46-vc4_hdmi]	50
[irq/47-vc4_hdmi]	50
[irq/43-vc4_hdmi]	50
[irq/42-vc4_hdmi]	50
[irq/52-vc4_hdmi]	50
[irq/53-vc4_hdmi]	50
[irq/49-vc4_hdmi]	50
[irq/48-vc4_hdmi]	50
[card1-crtc0]	50
[card1-crtc1]	50
[card1-crtc2]	50
[card1-crtc3]	50
[card1-crtc4]	50
[card1-crtc5]	50
/usr/libexec/tracker-miner-	0

Table 2. Processes that have real-time priorities.

(3) They may need to be run with real-time priority because they are related to irq (interrupt requests), system idle, watchdog, migration, HDMI, or card. Some of them are related to drivers. System must schedule these CMDs real-time to run normally.

7. Please state (1) how many sched.switch events occur on your program's processor, (2) whether your program is ever preempted, and if so (3) when and where is it preempted?

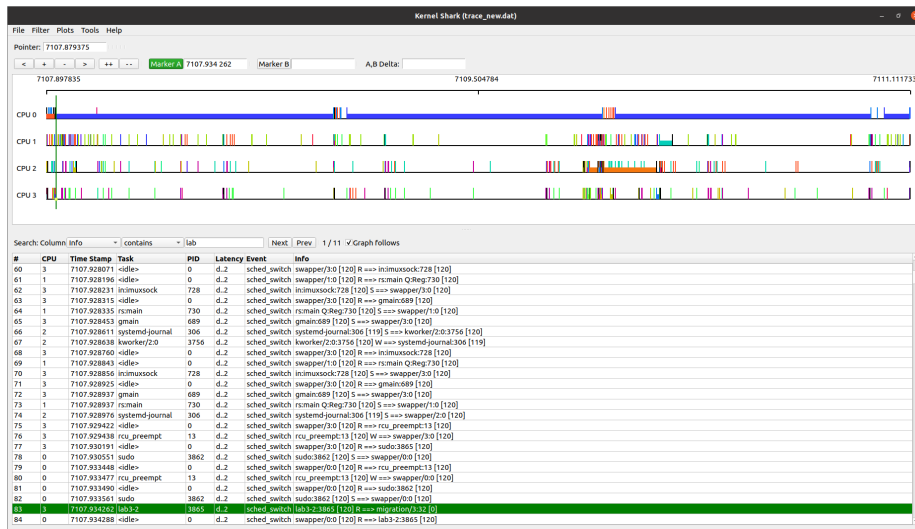


Figure 8. kernelshark new\_track.dat.

- (1) 117 sched\_switch events occur on your program's processor (CPU 0).
- (2) Our program is preempted.
- (3) It is preempted at tracker-miner-f, <idle>, nautilus, kworker, rcu\_preempt, gdbus, trace-cmd, pool-tracker-st, ksoftirqd.

## References

- [1] Paul E McKenney. "Real Time'vs.'Real Fast': How to Choose?" In: *Ottawa Linux Symposium (July 2008)*, pp. v2. 2008, pp. 57–65.