CS 251 / EE 255 REAL-TIME EMBEDDED SYSTEMS
HW Project #2: Task Scheduling and Monitoring
Group16

Group members

Hengshuo Zhang
hzhan402@ucr.edu

Zhaoze Sun
zsun114@ucr.edu

Writeup

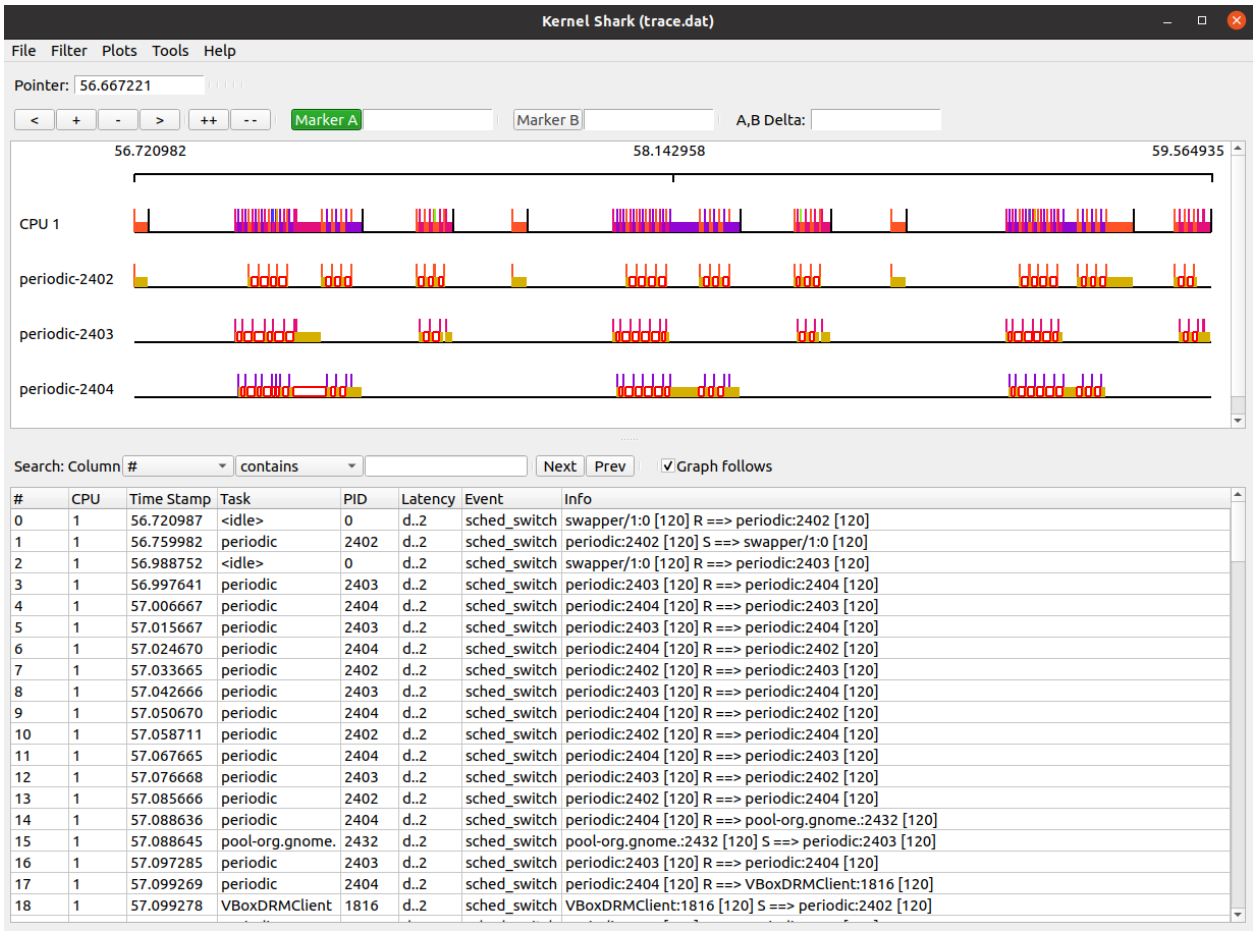
1. Problem 1

Steps for set up:

```
rex@rex:~/rtes/proj2/apps/periodic$ ./periodic 40 250 1 & ./periodic 60 500 1 & ./periodic 200 1000 1 &
[1] 2402
[2] 2403
[3] 2404
rex@rex:~/rtes/proj2/apps/periodic$ PID: 2402, C: 40, T: 250, CPUID: 1
PID: 2403, C: 60, T: 500, CPUID: 1
PID: 2404, C: 200, T: 1000, CPUID: 1
```

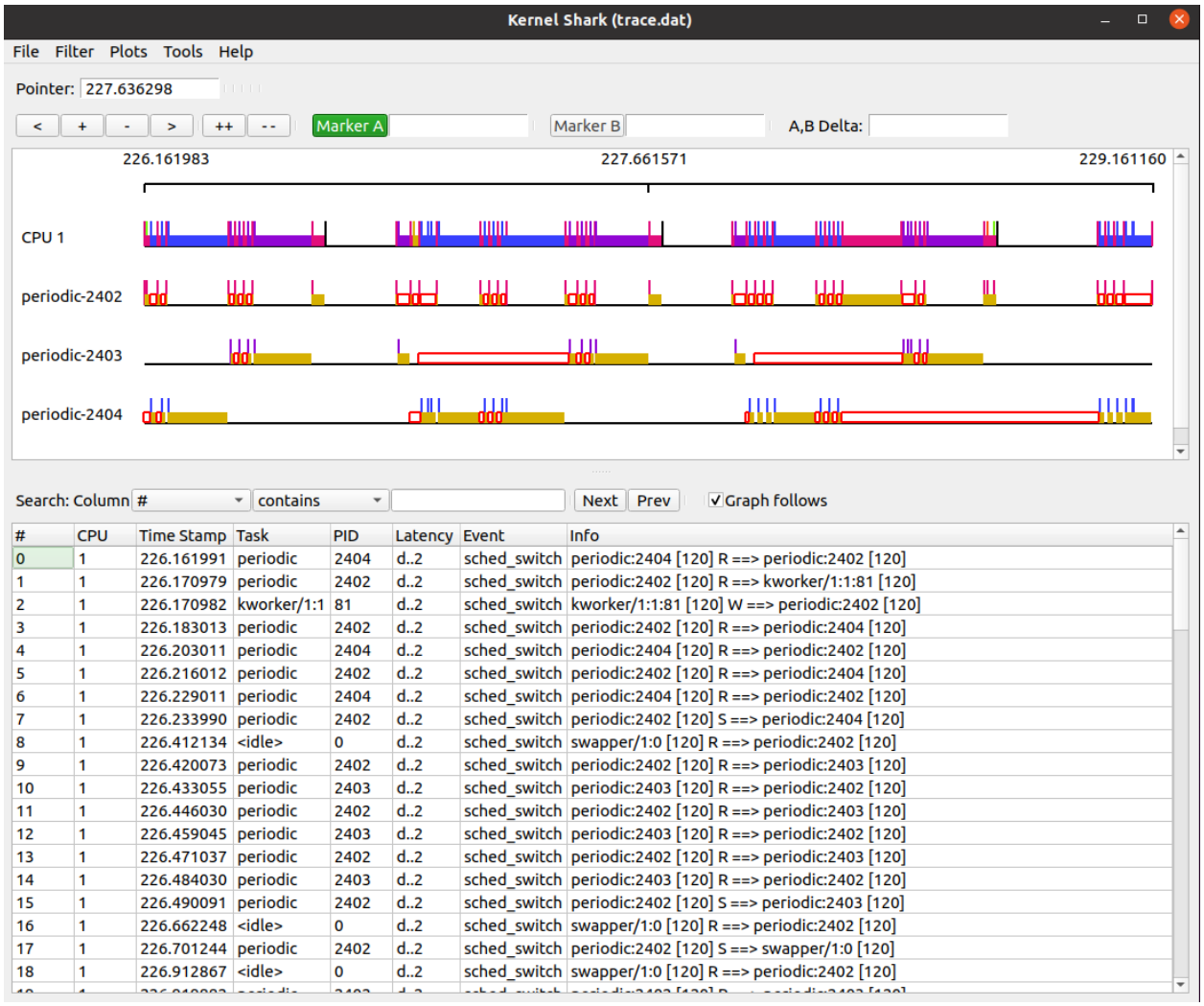
```
// Run three instances of the periodic program
> ./periodic 40 250 1 & ./periodic 60 500 1 & ./periodic 200 1000 1 &
```

The first instance has PID 2402, with a period of 40ms, a computation time of 250ms.
The second instance has PID 2403, with a period of 60ms, a computation time of 500ms.
The third instance has PID 2404, with a period of 200ms, a computation time of 1000ms.



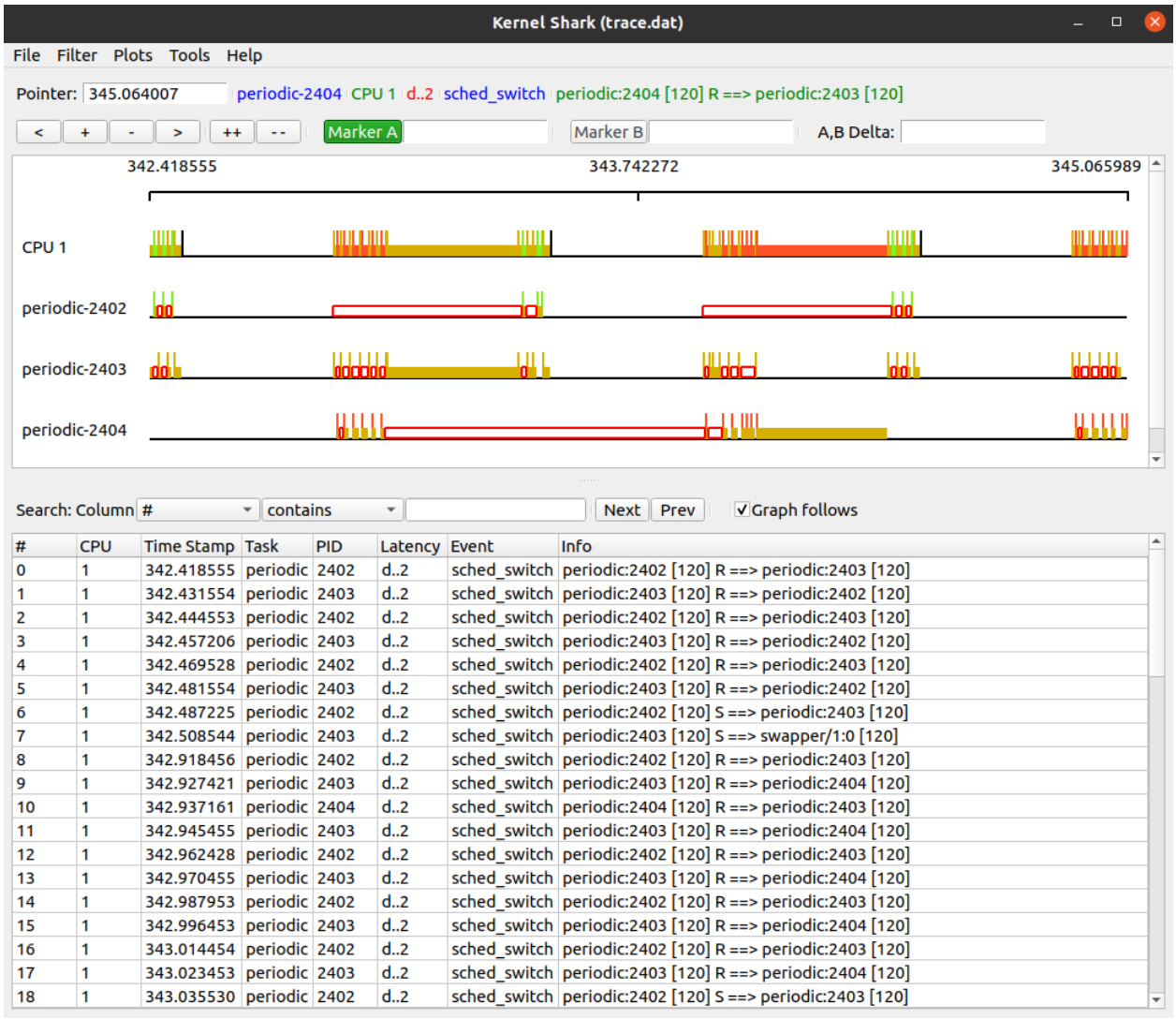
#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	56.720987	<idle>	0	d..2	sched_switch	swapper/1:0 [120] R ==> periodic:2402 [120]
1	1	56.759982	periodic	2402	d..2	sched_switch	periodic:2402 [120] S ==> swapper/1:0 [120]
2	1	56.988752	<idle>	0	d..2	sched_switch	swapper/1:0 [120] R ==> periodic:2403 [120]
3	1	56.997641	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
4	1	57.006667	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2403 [120]
5	1	57.015667	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
6	1	57.024670	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
7	1	57.033665	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
8	1	57.042666	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
9	1	57.050670	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
10	1	57.058711	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]
11	1	57.067665	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2403 [120]
12	1	57.076668	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
13	1	57.085666	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]
14	1	57.088636	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> pool-org.gnome.:2432 [120]
15	1	57.088645	pool-org.gnome.	2432	d..2	sched_switch	pool-org.gnome.:2432 [120] S ==> periodic:2403 [120]
16	1	57.097285	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
17	1	57.099269	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> VBoxDRMClient:1816 [120]
18	1	57.099278	VBoxDRMClient	1816	d..2	sched_switch	VBoxDRMClient:1816 [120] S ==> periodic:2402 [120]

```
// record all tasks
> sudo trace-cmd record -e sched_switch -P 2402 -P 2403 -P 2404 sleep 3
// displays all events recorded
> kernelshark
```



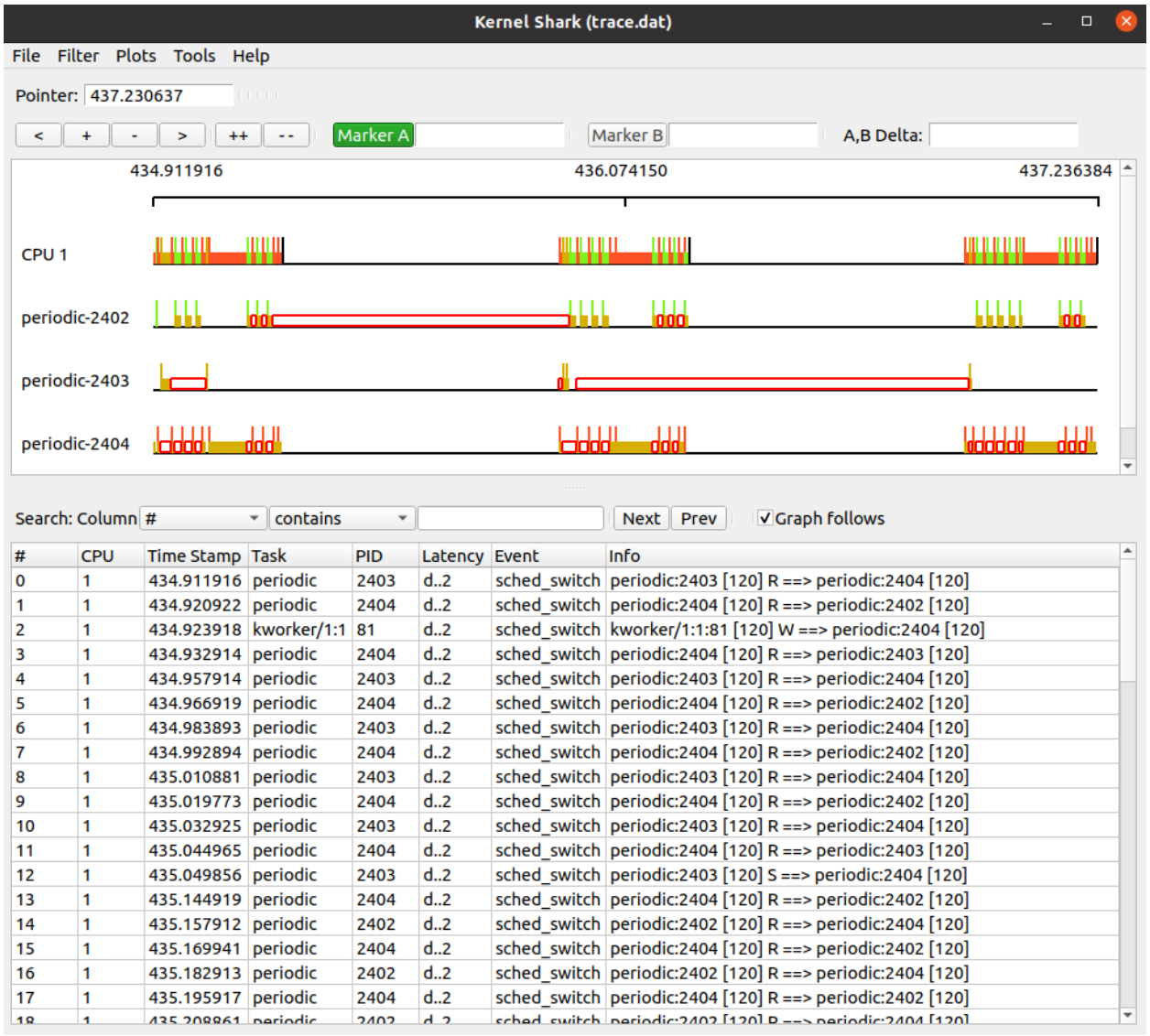
#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	226.161991	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
1	1	226.170979	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> kworker/1:1:81 [120]
2	1	226.170982	kworker/1:1	81	d..2	sched_switch	kworker/1:1:81 [120] W ==> periodic:2402 [120]
3	1	226.183013	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]
4	1	226.203011	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
5	1	226.216012	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]
6	1	226.229011	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
7	1	226.233990	periodic	2402	d..2	sched_switch	periodic:2402 [120] S ==> periodic:2404 [120]
8	1	226.412134	<idle>	0	d..2	sched_switch	swapper/1:0 [120] R ==> periodic:2402 [120]
9	1	226.420073	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
10	1	226.433055	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
11	1	226.446030	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
12	1	226.459045	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
13	1	226.471037	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
14	1	226.484030	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
15	1	226.490091	periodic	2402	d..2	sched_switch	periodic:2402 [120] S ==> periodic:2403 [120]
16	1	226.662248	<idle>	0	d..2	sched_switch	swapper/1:0 [120] R ==> periodic:2402 [120]
17	1	226.701244	periodic	2402	d..2	sched_switch	periodic:2402 [120] S ==> swapper/1:0 [120]
18	1	226.912867	<idle>	0	d..2	sched_switch	swapper/1:0 [120] R ==> periodic:2402 [120]

```
// Record scheduling events of 2402 for 3s
> sudo trace-cmd record -e sched_switch -P 2402 sleep 3
// displays all events recorded
> kernelshark
```



#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	342.418555	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
1	1	342.431554	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
2	1	342.444553	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
3	1	342.457206	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
4	1	342.469528	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
5	1	342.481554	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2402 [120]
6	1	342.487225	periodic	2402	d..2	sched_switch	periodic:2402 [120] S ==> periodic:2403 [120]
7	1	342.508544	periodic	2403	d..2	sched_switch	periodic:2403 [120] S ==> swapper/1:0 [120]
8	1	342.918456	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
9	1	342.927421	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
10	1	342.937161	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2403 [120]
11	1	342.945455	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
12	1	342.962428	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
13	1	342.970455	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
14	1	342.987953	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
15	1	342.996453	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
16	1	343.014454	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2403 [120]
17	1	343.023453	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
18	1	343.035530	periodic	2402	d..2	sched_switch	periodic:2402 [120] S ==> periodic:2403 [120]

```
// Record scheduling events of 2403 for 3s
> sudo trace-cmd record -e sched_switch -P 2403 sleep 3
// displays all events recorded
> kernelshark
```



#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	434.911916	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
1	1	434.920922	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
2	1	434.923918	kworker/1:1	81	d..2	sched_switch	kworker/1:1:81 [120] W ==> periodic:2404 [120]
3	1	434.932914	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2403 [120]
4	1	434.957914	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
5	1	434.966919	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
6	1	434.983893	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
7	1	434.992894	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
8	1	435.010881	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
9	1	435.019773	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
10	1	435.032925	periodic	2403	d..2	sched_switch	periodic:2403 [120] R ==> periodic:2404 [120]
11	1	435.044965	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2403 [120]
12	1	435.049856	periodic	2403	d..2	sched_switch	periodic:2403 [120] S ==> periodic:2404 [120]
13	1	435.144919	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
14	1	435.157912	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]
15	1	435.169941	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
16	1	435.182913	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]
17	1	435.195917	periodic	2404	d..2	sched_switch	periodic:2404 [120] R ==> periodic:2402 [120]
18	1	435.208861	periodic	2402	d..2	sched_switch	periodic:2402 [120] R ==> periodic:2404 [120]

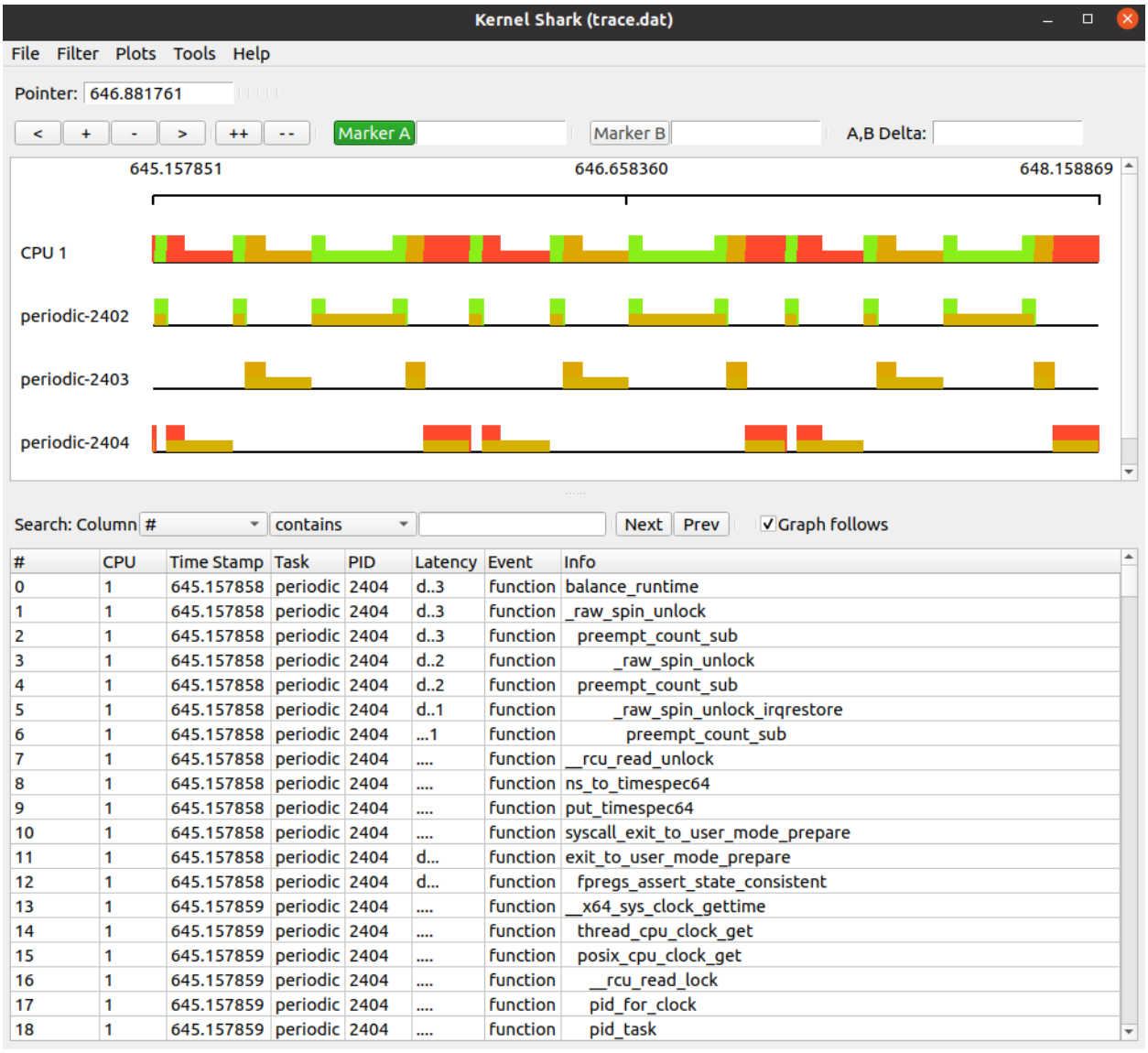
```
// Record scheduling events of 2404 for 3s
> sudo trace-cmd record -e sched_switch -P 2404 sleep 3
// displays all events recorded
> kernelshark
```

2. Problem 2

Steps for set up:

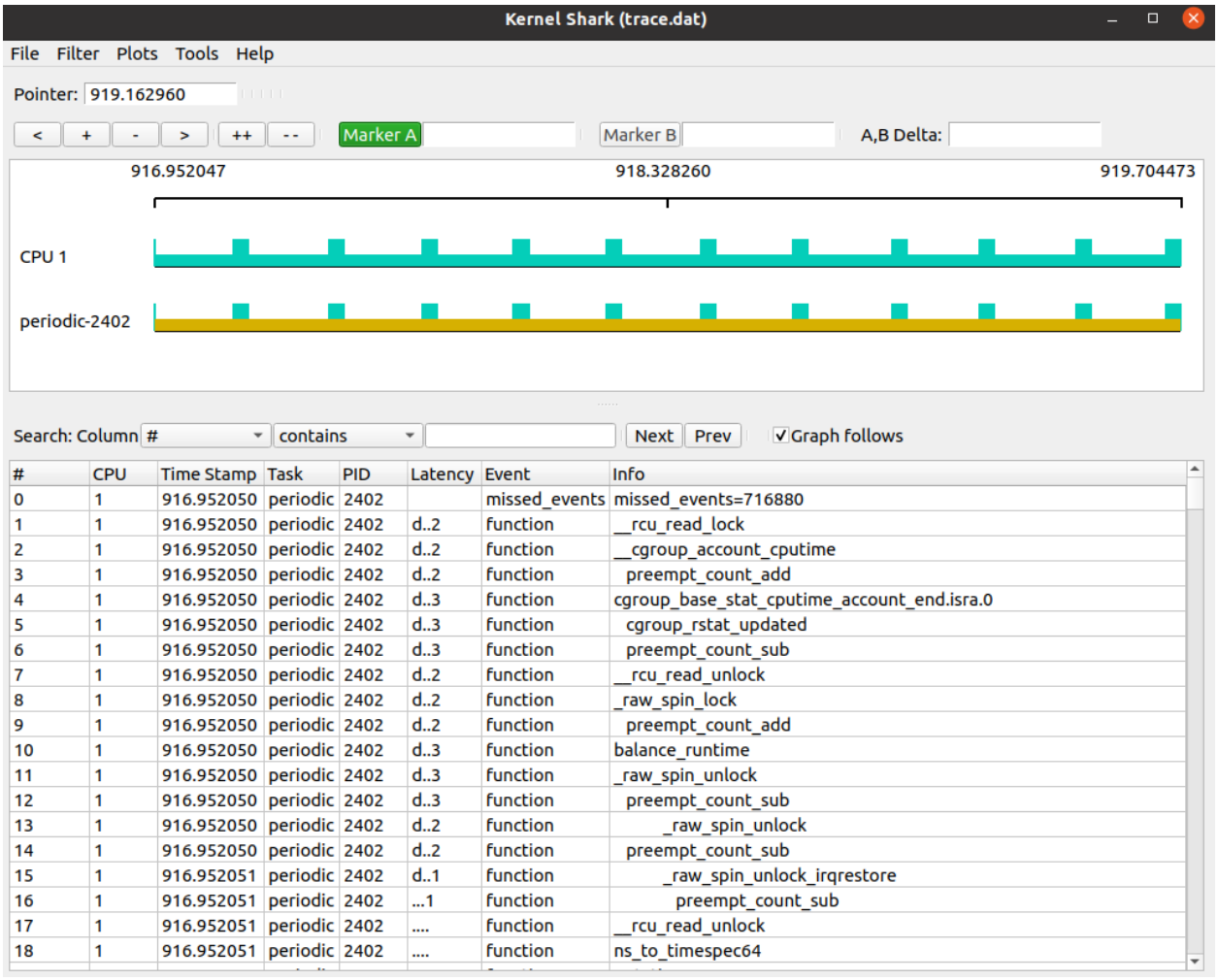
```
> sudo chrt -f -p 80 2402
> sudo chrt -f -p 79 2403
> sudo chrt -f -p 78 2404
```

High Priority: 2402 (80)
Set the real-time priority of 2402 to 80 and implement the SCHED_FIFO scheduling policy.
Mid Priority: 2403 (79)
Set the real-time priority of 2403 to 79 and implement the SCHED_FIFO scheduling policy.
Low Priority: 2404 (78)
Set the real-time priority of 2404 to 78 and implement the SCHED_FIFO scheduling policy.



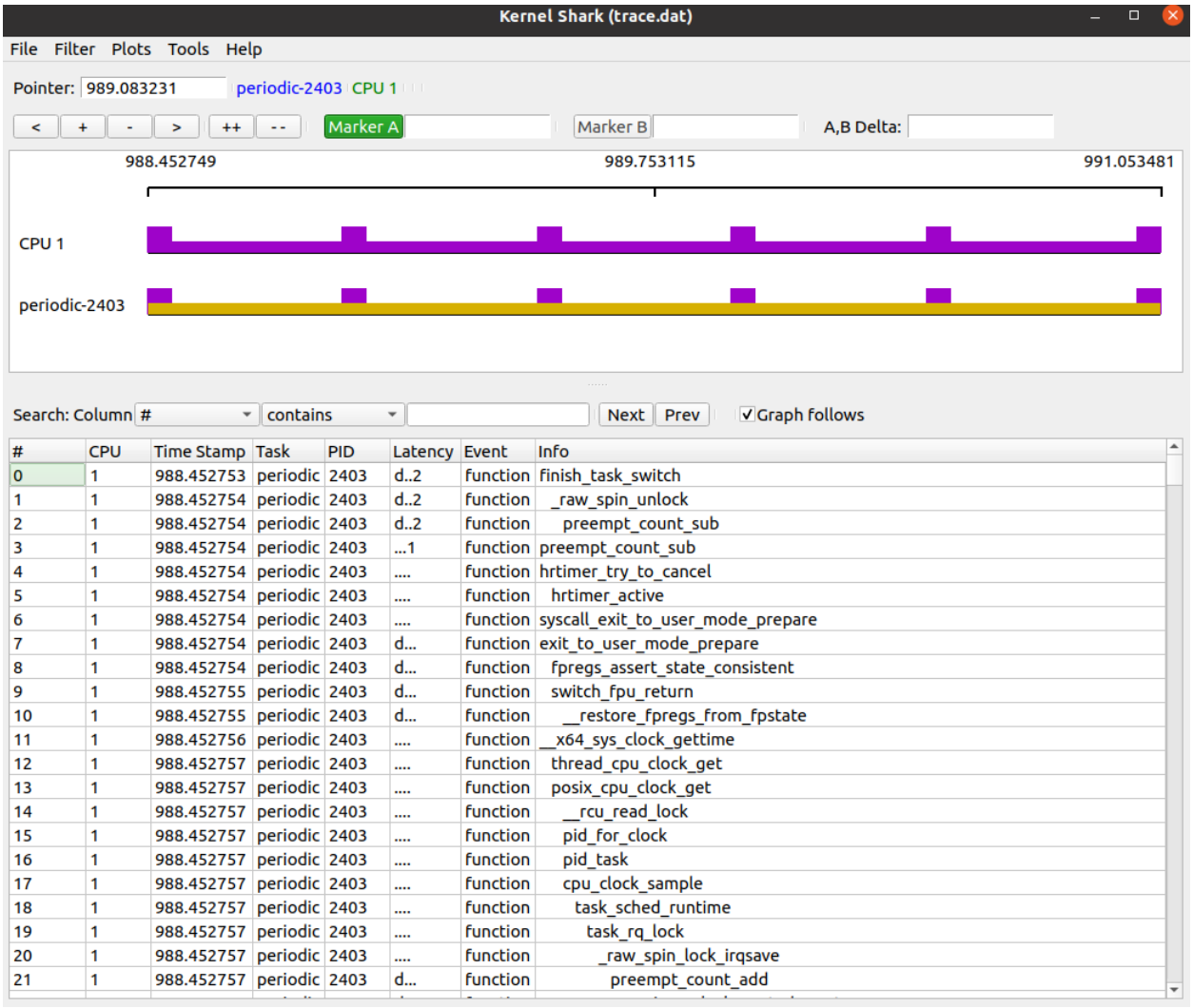
#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	645.157858	periodic	2404	d..3	function	balance_runtime
1	1	645.157858	periodic	2404	d..3	function	_raw_spin_unlock
2	1	645.157858	periodic	2404	d..3	function	preempt_count_sub
3	1	645.157858	periodic	2404	d..2	function	_raw_spin_unlock
4	1	645.157858	periodic	2404	d..2	function	preempt_count_sub
5	1	645.157858	periodic	2404	d..1	function	_raw_spin_unlock_irqrestore
6	1	645.157858	periodic	2404	...1	function	preempt_count_sub
7	1	645.157858	periodic	2404	....	function	__rcu_read_unlock
8	1	645.157858	periodic	2404	....	function	ns_to_timespec64
9	1	645.157858	periodic	2404	....	function	put_timespec64
10	1	645.157858	periodic	2404	....	function	syscall_exit_to_user_mode_prepare
11	1	645.157858	periodic	2404	d...	function	exit_to_user_mode_prepare
12	1	645.157858	periodic	2404	d...	function	fpregs_assert_state_consistent
13	1	645.157859	periodic	2404	....	function	__x64_sys_clock_gettime
14	1	645.157859	periodic	2404	....	function	thread_cpu_clock_get
15	1	645.157859	periodic	2404	....	function	posix_cpu_clock_get
16	1	645.157859	periodic	2404	....	function	__rcu_read_lock
17	1	645.157859	periodic	2404	....	function	pid_for_clock
18	1	645.157859	periodic	2404	....	function	pid_task

```
> sudo trace-cmd record -p function -P 2402 -P 2403 -P 2404
sleep 3
> kernelshark
```



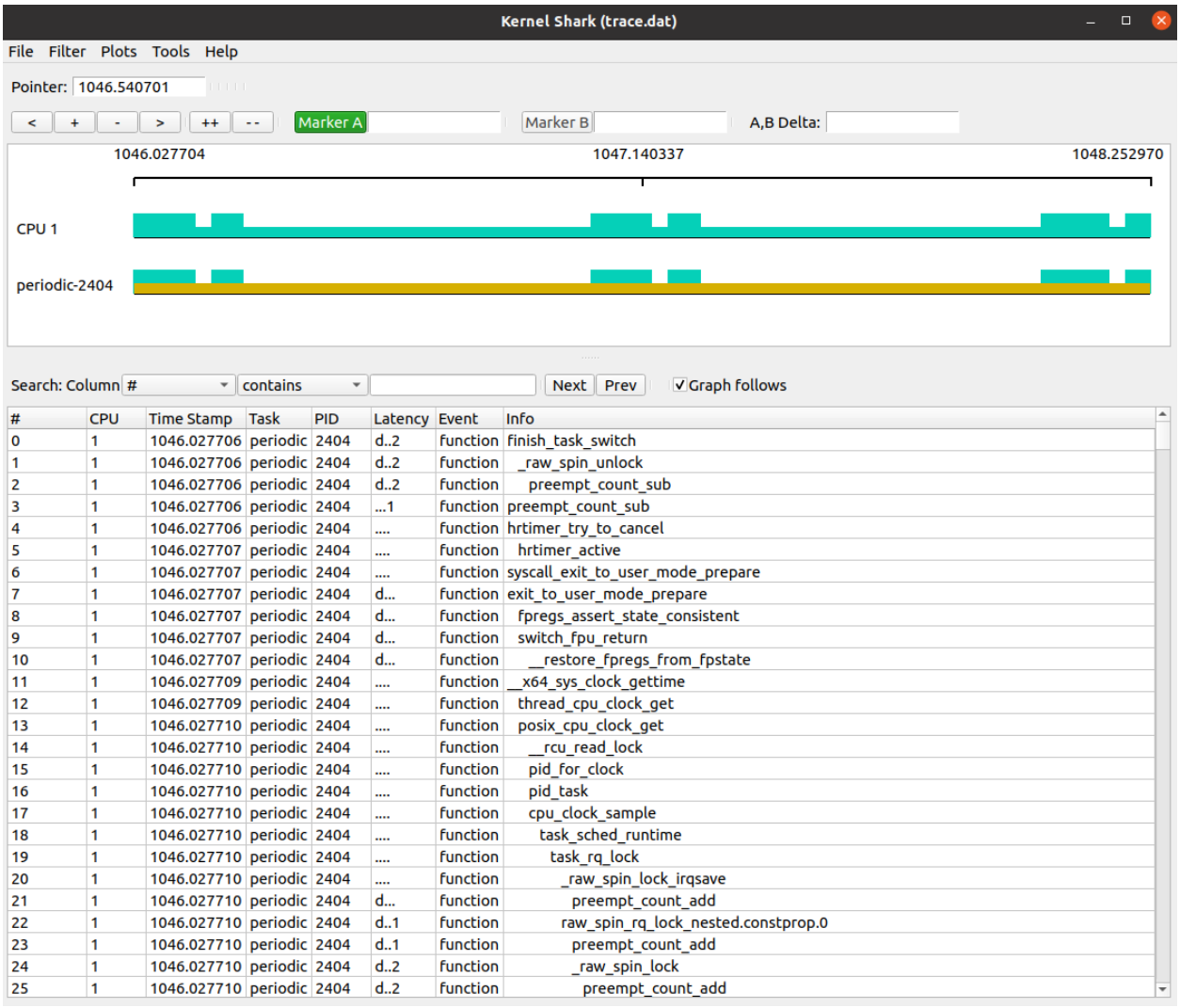
#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	916.952050	periodic	2402		missed_events	missed_events=716880
1	1	916.952050	periodic	2402	d..2	function	__rcu_read_lock
2	1	916.952050	periodic	2402	d..2	function	__cgroup_account_cputime
3	1	916.952050	periodic	2402	d..2	function	preempt_count_add
4	1	916.952050	periodic	2402	d..3	function	cgroup_base_stat_cputime_account_end.isra.0
5	1	916.952050	periodic	2402	d..3	function	cgroup_rstat_updated
6	1	916.952050	periodic	2402	d..3	function	preempt_count_sub
7	1	916.952050	periodic	2402	d..2	function	__rcu_read_unlock
8	1	916.952050	periodic	2402	d..2	function	_raw_spin_lock
9	1	916.952050	periodic	2402	d..2	function	preempt_count_add
10	1	916.952050	periodic	2402	d..3	function	balance_runtime
11	1	916.952050	periodic	2402	d..3	function	_raw_spin_unlock
12	1	916.952050	periodic	2402	d..3	function	preempt_count_sub
13	1	916.952050	periodic	2402	d..2	function	_raw_spin_unlock
14	1	916.952050	periodic	2402	d..2	function	preempt_count_sub
15	1	916.952051	periodic	2402	d..1	function	_raw_spin_unlock_irqrestore
16	1	916.952051	periodic	2402	...1	function	preempt_count_sub
17	1	916.952051	periodic	2402	....	function	__rcu_read_unlock
18	1	916.952051	periodic	2402	....	function	ns_to_timespec64

```
> sudo trace-cmd record -p function -P 2402 sleep 3
> kernelshark
```



#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	988.452753	periodic	2403	d..2	function	finish_task_switch
1	1	988.452754	periodic	2403	d..2	function	_raw_spin_unlock
2	1	988.452754	periodic	2403	d..2	function	preempt_count_sub
3	1	988.452754	periodic	2403	...1	function	preempt_count_sub
4	1	988.452754	periodic	2403	....	function	hrtimer_try_to_cancel
5	1	988.452754	periodic	2403	....	function	hrtimer_active
6	1	988.452754	periodic	2403	....	function	syscall_exit_to_user_mode_prepare
7	1	988.452754	periodic	2403	d...	function	exit_to_user_mode_prepare
8	1	988.452754	periodic	2403	d...	function	fpregs_assert_state_consistent
9	1	988.452755	periodic	2403	d...	function	switch_fpu_return
10	1	988.452755	periodic	2403	d...	function	__restore_fpregs_from_fpstate
11	1	988.452756	periodic	2403	....	function	__x64_sys_clock_gettime
12	1	988.452757	periodic	2403	....	function	thread_cpu_clock_get
13	1	988.452757	periodic	2403	....	function	posix_cpu_clock_get
14	1	988.452757	periodic	2403	....	function	__rcu_read_lock
15	1	988.452757	periodic	2403	....	function	pid_for_clock
16	1	988.452757	periodic	2403	....	function	pid_task
17	1	988.452757	periodic	2403	....	function	cpu_clock_sample
18	1	988.452757	periodic	2403	....	function	task_sched_runtime
19	1	988.452757	periodic	2403	....	function	task_rq_lock
20	1	988.452757	periodic	2403	....	function	_raw_spin_lock_irqsave
21	1	988.452757	periodic	2403	d...	function	preempt_count_add

```
> sudo trace-cmd record -p function -P 2403 sleep 3
> kernelshark
```



#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	1	1046.027706	periodic	2404	d..2	function	finish_task_switch
1	1	1046.027706	periodic	2404	d..2	function	_raw_spin_unlock
2	1	1046.027706	periodic	2404	d..2	function	preempt_count_sub
3	1	1046.027706	periodic	2404	...1	function	preempt_count_sub
4	1	1046.027706	periodic	2404	....	function	hrtimer_try_to_cancel
5	1	1046.027707	periodic	2404	....	function	hrtimer_active
6	1	1046.027707	periodic	2404	....	function	syscall_exit_to_user_mode_prepare
7	1	1046.027707	periodic	2404	d...	function	exit_to_user_mode_prepare
8	1	1046.027707	periodic	2404	d...	function	fpregs_assert_state_consistent
9	1	1046.027707	periodic	2404	d...	function	switch_fpu_return
10	1	1046.027707	periodic	2404	d...	function	__restore_fpregs_from_fpstate
11	1	1046.027709	periodic	2404	....	function	__x64_sys_clock_gettime
12	1	1046.027709	periodic	2404	....	function	thread_cpu_clock_get
13	1	1046.027710	periodic	2404	....	function	posix_cpu_clock_get
14	1	1046.027710	periodic	2404	....	function	__rcu_read_lock
15	1	1046.027710	periodic	2404	....	function	pid_for_clock
16	1	1046.027710	periodic	2404	....	function	pid_task
17	1	1046.027710	periodic	2404	....	function	cpu_clock_sample
18	1	1046.027710	periodic	2404	....	function	task_sched_runtime
19	1	1046.027710	periodic	2404	....	function	task_rq_lock
20	1	1046.027710	periodic	2404	....	function	_raw_spin_lock_irqsave
21	1	1046.027710	periodic	2404	d...	function	preempt_count_add
22	1	1046.027710	periodic	2404	d..1	function	raw_spin_rq_lock_nested.constprop.0
23	1	1046.027710	periodic	2404	d..1	function	preempt_count_add
24	1	1046.027710	periodic	2404	d..2	function	_raw_spin_lock
25	1	1046.027710	periodic	2404	d..2	function	preempt_count_add

```
> sudo trace-cmd record -p function -P 2404 sleep 3
> kernelshark
```

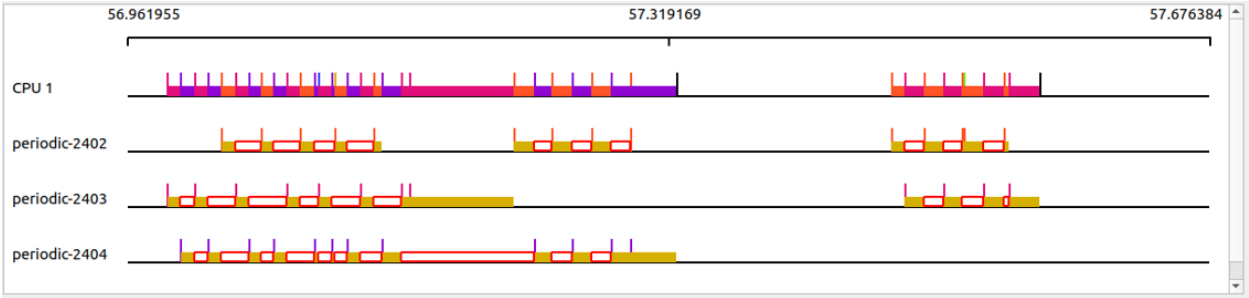
3. Problem 3

In Task 1, each task is executed using the regular scheduling policy. The execution trace will show that each task is executed at a given time period with a delay equal to or less than the system jitter.

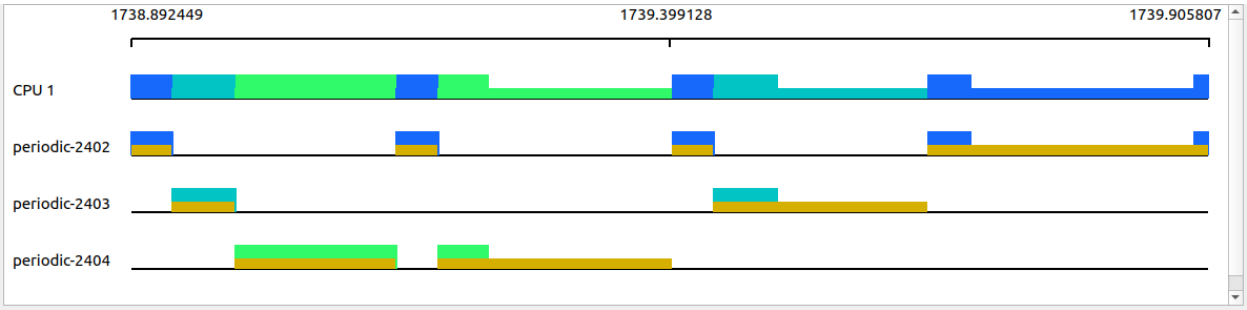
The periodic program creates a thread for each instance and uses the clock_nanosleep system call to put the thread to sleep for a period of time. When the thread wakes up, it executes the task, which is a simple busy-wait cycle that consumes CPU time for a specified duration C. After the task completes, the thread returns to the sleep state for the remainder of the time until the next cycle T.

In Task 2, each task is assigned a real-time priority via SCHED_FIFO. The execution trace will show that the higher priority task is executed before the lower priority task. Tasks with lower priority may have delays beyond their cycle time because the system always prioritizes tasks with higher priority. However, tasks with higher priority may face longer blocking times from the operating system or other tasks.

Real-time priority is a value that determines the order in which tasks are executed by the kernel scheduler. A higher priority value indicates a more critical task that needs to be executed before a lower priority task. With the SCHED_FIFO policy, this means that tasks with higher priority will run to completion before any lower priority tasks run. The priority values range from 1 to 99, where 99 is the highest priority.



Regular Scheduling



SCHED_FIFO

In a regular scheduling diagram, there is no real-time priority assignment in the cycle, and we are not able to find exactly the same two execution traces. For the analysis of the execution traces, we are not able to visualize whether the task has completed in the cycle or has been delayed or lost.

For SCHED_FIFO, the periodicity of execution traces will be scheduled with real-time priority. We can intuitively understand the execution of tasks under the same CPU by a single cycle, and with the influence of priority, we will have a more intuitive understanding of the priority relationship between tasks.

Here is summary of using different strategies:

Method	Pros	Cons
Regular Scheduling Policy	Simple and easy to implement	May not be suitable for real-time systems or applications that require timely responses to external events
	Tasks execute in a predictable and deterministic manner	Tasks may experience jitter or delay if the system is overloaded or if other high-priority tasks are executing at the same time
		Cannot prioritize critical tasks over less important tasks
Real-Time Priority Assignments with SCHED_FIFO	Provides a way to prioritize critical tasks over less important tasks	Requires knowledge and understanding of the kernel scheduler and real-time priorities
	Enables real-time systems or applications to respond quickly to external events	Assigning high priorities to tasks can lead to priority inversion, where a low-priority task blocks a higher-priority task
	Tasks are executed in a predictable and deterministic manner	Assigning high priorities to tasks can also lead to starvation, where a low-priority task is not executed at all

4. Summary of The Contributions

Overall:

Hengshuo Zhang	50%
Zhaoze Sun	50%

Breakdown to each Assignment

4.2	Setting Task's Timing Parameters	Hengshuo/Zhaoze
4.3	Printing Task's Timing Parameters	Hengshuo/Zhaoze
4.4	Admission Control	Hengshuo/Zhaoze
4.5	Computation Time Tracking	Hengshuo/Zhaoze
4.6	Periodic Task Support	Hengshuo/Zhaoze
4.8	Writeup	Hengshuo/Zhaoze

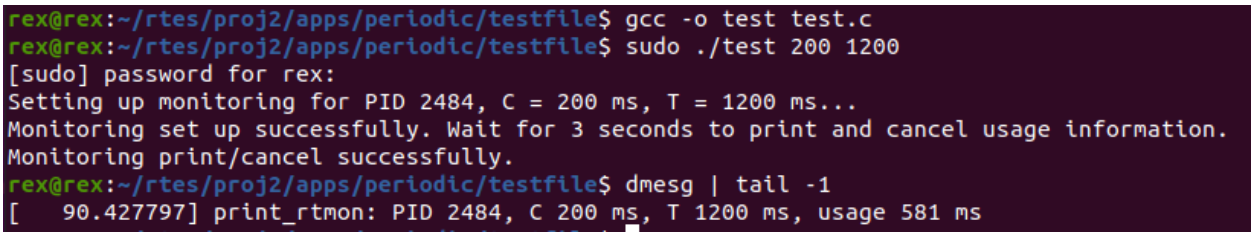
Readme:

The (set_rtmon/print_rtmon/cancel_rtmon) system calls in 4.2 to 4.5 are saved in the directory path (... rtes/proj2/kernel/rtmon.c).

For testing purposes, the test.c file in the apps folder will be available for testing the three system calls. The path to the test file is (... rtes/proj2/apps/periodic/testfile/test.c).

Demo (set_rtmon/print_rtmon/cancel_rtmon) system calls:

```
> gcc -o test test.c
> sudo ./test 200 1200
> dmesg
```



We used a separate .c file for the system calls (wait_until_next_period) in 4.6, which will show the logic more clearly and reduce pointer problems at runtime. The file path is (... rtes/proj2/kernel).
Based on the test structure mentioned in 4.6, we wrote a separate test file to test the system call. The path of the test file is (... rtes/proj2/apps/periodic/testfile/demo.c).

Demo (wait_until_next_period) system calls:

```
> gcc -o demo demo.c
> sudo ./demo
```