Operating Systems Laboratory

Lab 3

Part 1

The code was changed in `minix/servers/sched/schedule.c`. The changes made were at line 328.

```
   if(rmp->priority >= USER_Q)
       printf("(210020047) Minix: PID %d swapped in\n",
_ENDPOINT_P(rmp->endpoint));
```

Please run make to run the script, which will copy the modified source file and rebuild the minix system.

Part 2

We ran the Unix Benchmark Suite on Minix to analyze its scheduler's behaviour. We adjusted the benchmark mix to create custom sets and observe how the scheduler reacts to different workloads.
Some observations:

- **arithoh.sh**: It runs arith.c, which involves arithmetic operations, which means it is CPU bound, not IO bound.
- **fstime.sh**: It runs fstime.c, which involves frequent IO operations to some files or buffers. It is IO-bound.
- **spawn.sh**: It runs spawn.c, it involves creating many children using the fork command. It is a system-level CPU-bound task. According to Minix documentation, these types of system calls are given the highest priority.

These scripts together are sufficient to analyze how the scheduler behaves for the user-level processes.

Shubh Agarwal

Test 1

Let us throw two CPU-bound processes together at the scheduler.

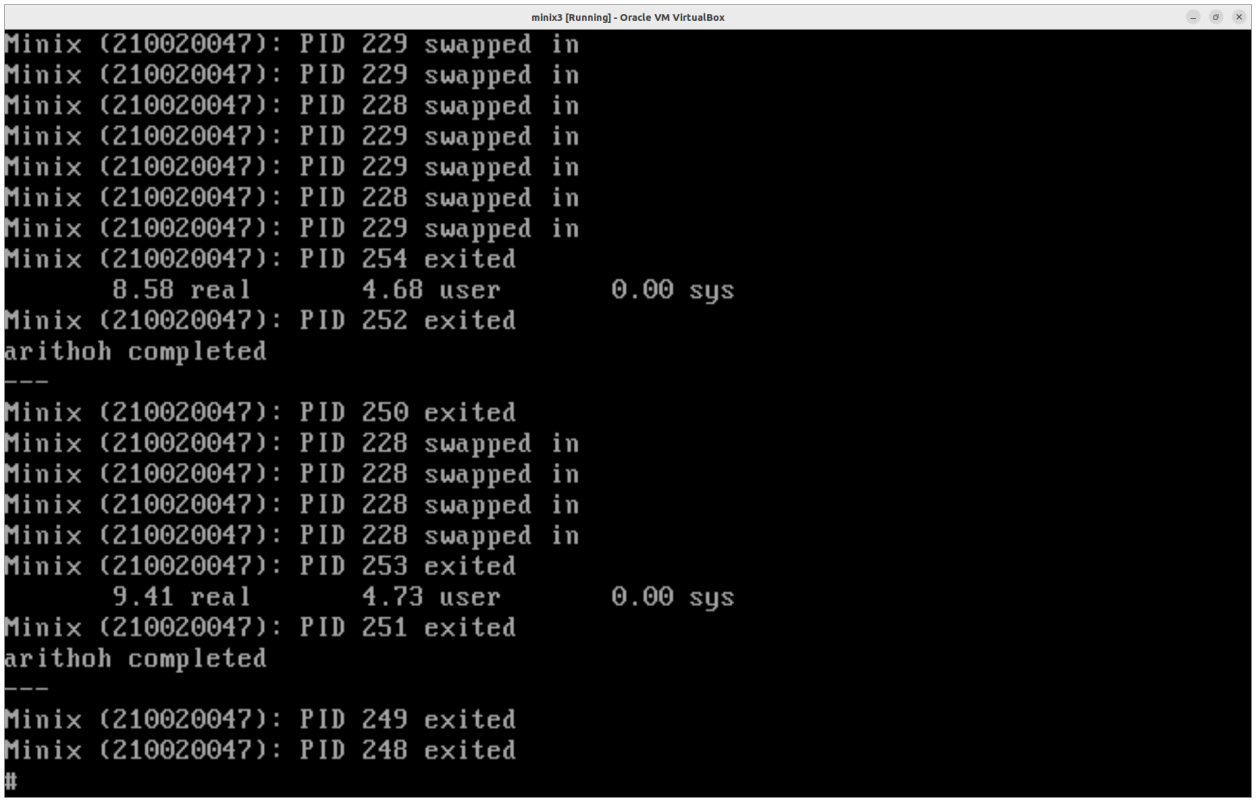
Script: workload_mix1.sh

```
./arithoh.sh &
./arithoh.sh &
wait
```

arithoh is CPU-bound, and there are two instances of it in our test. They are getting because they are getting Out-of-Quantum Messages after their quantum (or time slice) is used up. They both have equal priority, so they are allotted alternatively by the scheduler.

```
Minix (210020047): PID 248 created
Minix (210020047): PID 223 swapped in
Minix (210020047): PID 249 created
Minix (210020047): PID 224 swapped in
Minix (210020047): PID 250 created
Minix (210020047): PID 225 swapped in
Minix (210020047): PID 251 created
Minix (210020047): PID 226 swapped in
Minix (210020047): PID 252 created
Minix (210020047): PID 227 swapped in
Minix (210020047): PID 253 created
Minix (210020047): PID 228 swapped in
Minix (210020047): PID 254 created
Minix (210020047): PID 229 swapped in
Minix (210020047): PID 228 swapped in
Minix (210020047): PID 229 swapped in
Minix (210020047): PID 229 swapped in
Minix (210020047): PID 228 swapped in
Minix (210020047): PID 228 swapped in
Minix (210020047): PID 229 swapped in
Minix (210020047): PID 228 swapped in
Minix (210020047): PID 229 swapped in
_
```

The PID of the first arithoh is 228, and the PID of the second one is 229.



Test 2

Let us throw both IO-bound and CPU-bound processes together at the scheduler.

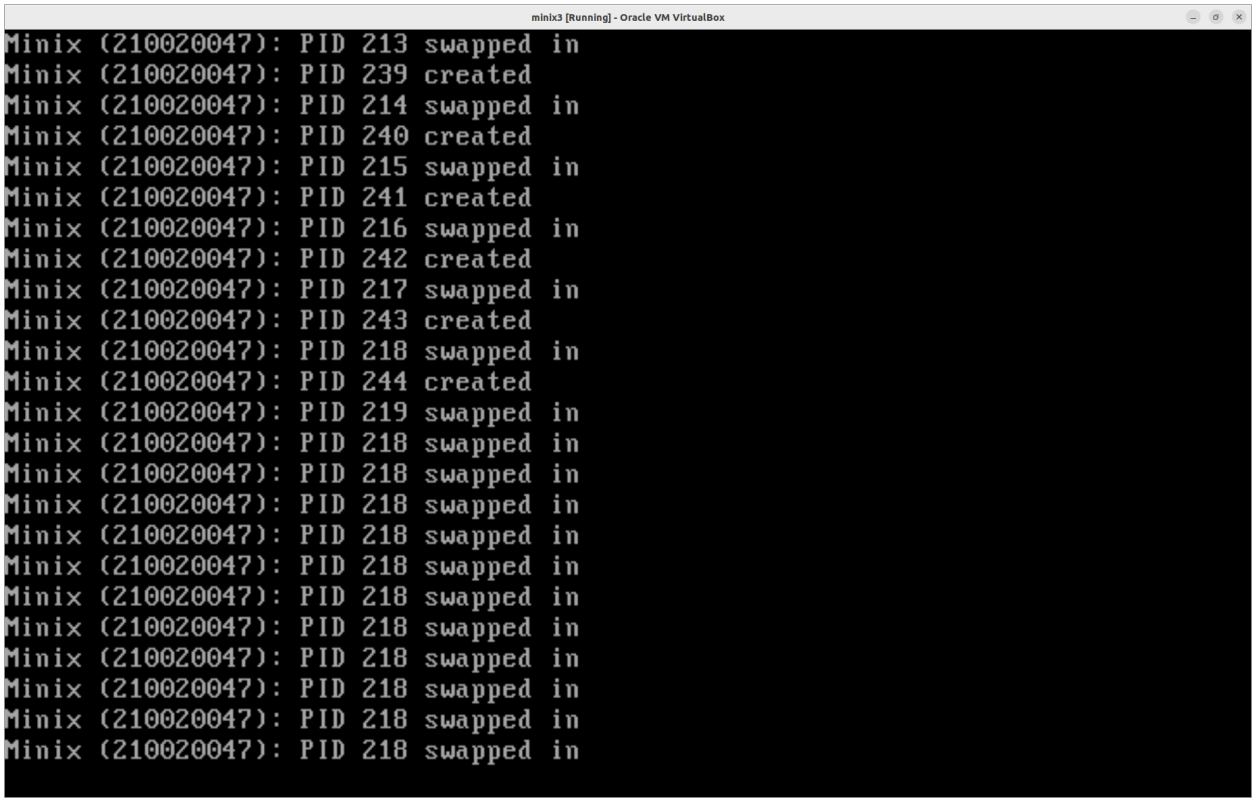
Script: workload_mix2.sh

```
./arithoh.sh &
./fstime.sh &
wait
```

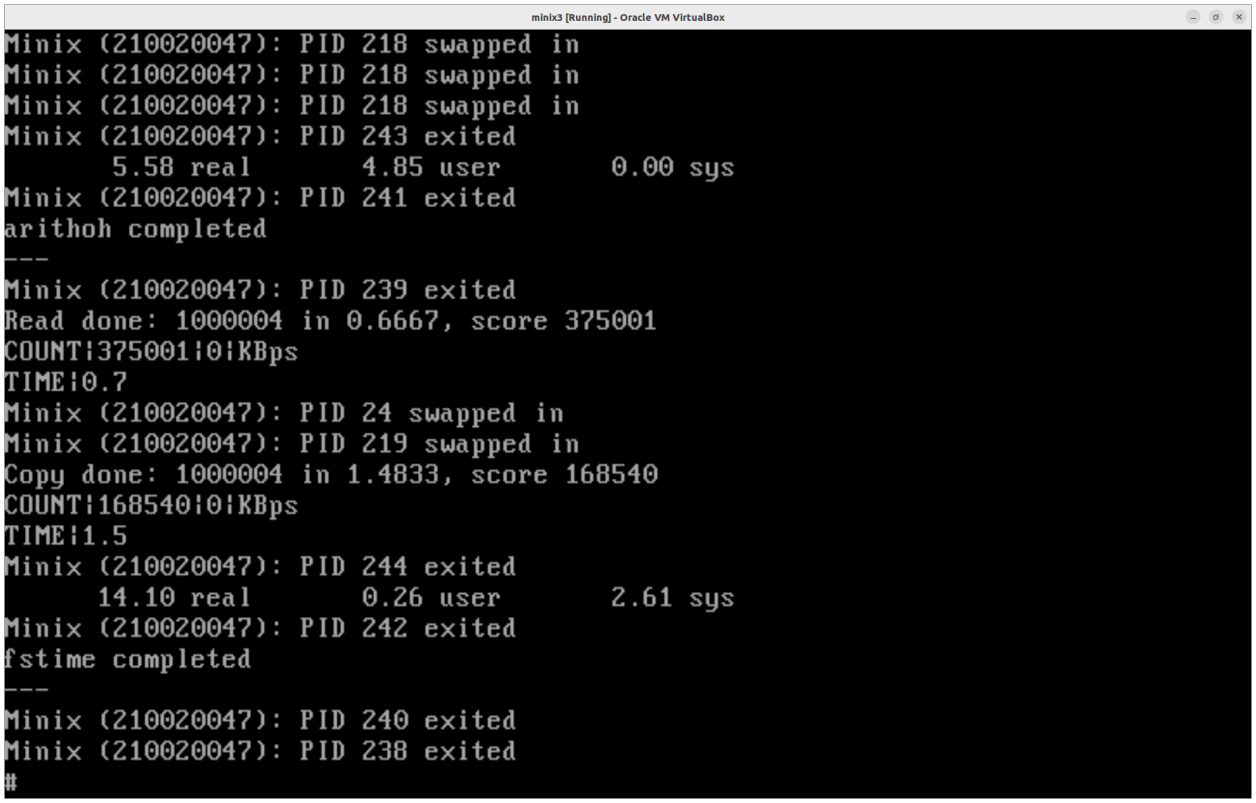
arithoh is CPU-bound, and it is not getting blocked because of IO, but it is getting terminated because of finishing its quantum (or time slice). Since it is not waiting for IO, it is again getting a chance to run just after it has terminated.

After waiting for IO, **fstime** gets back in the queue to be executed. Once it receives the request, it starts getting executed, as by then arithoh was completed.

The PIF of arithoh is 218 and the PID of fstime is 219.



We can see in this screenshot that the first 218 was scheduled, then 219 was scheduled, but then 219 got blocked. After that, only 218 was getting scheduled.

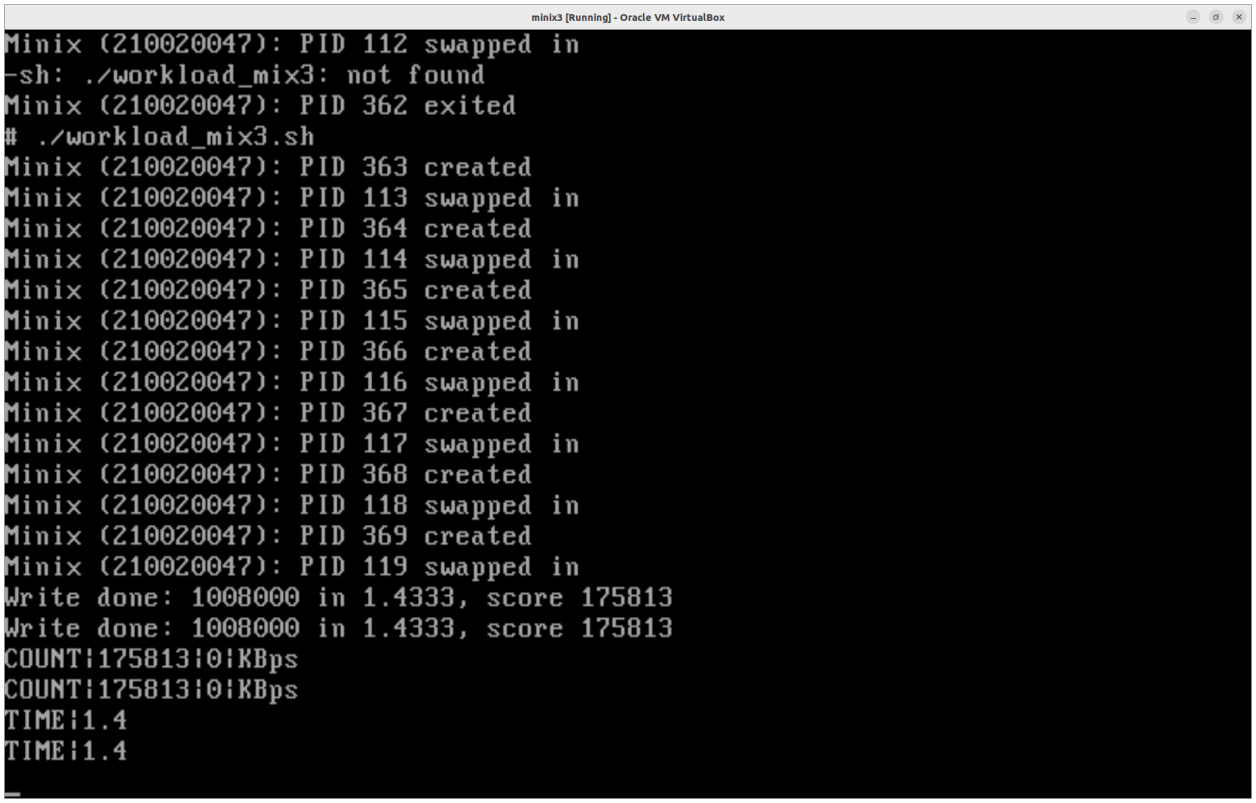


Test 3

Let us throw two IO-bound processes together at the scheduler.

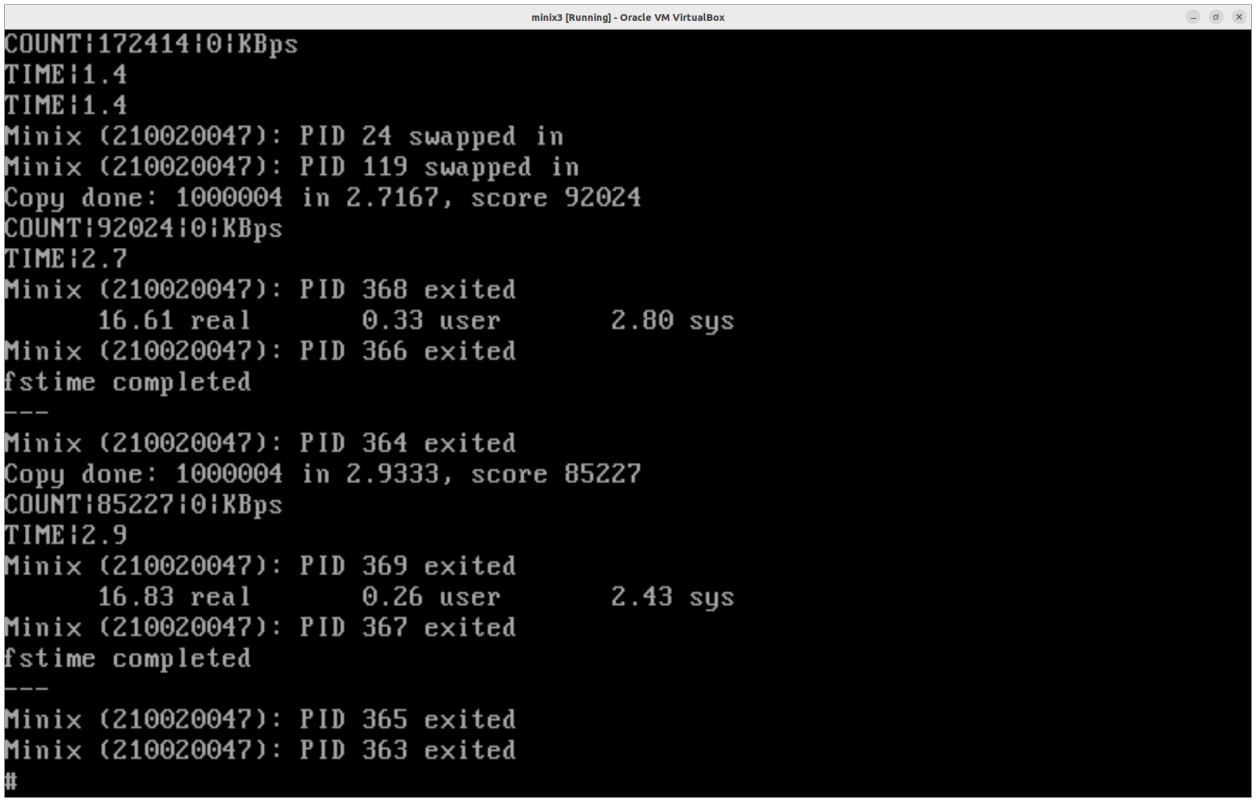
Script: workload_mix3.sh

```
./fstime.sh &
./fstime.sh &
wait
```



fstime is an IO-bound process, we are running them parallelly. They both are waiting for the IO events. Both processes exit one after the other as their respective I/O operations finish. This is shown in the attached screenshot.

The PIF of fstime is 118 and the PID of fstime is 119.



Test 4

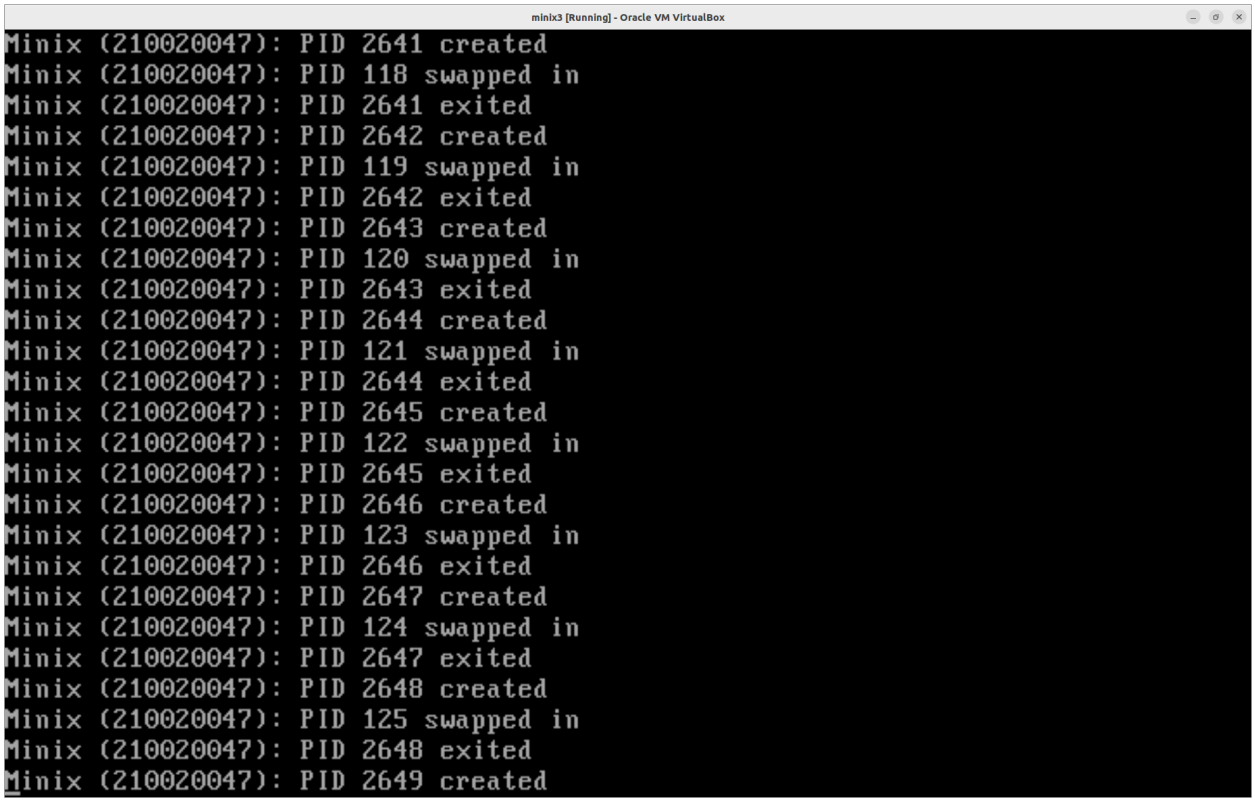
Let us throw two CPU-bound processes, but of different priority, together at the scheduler.

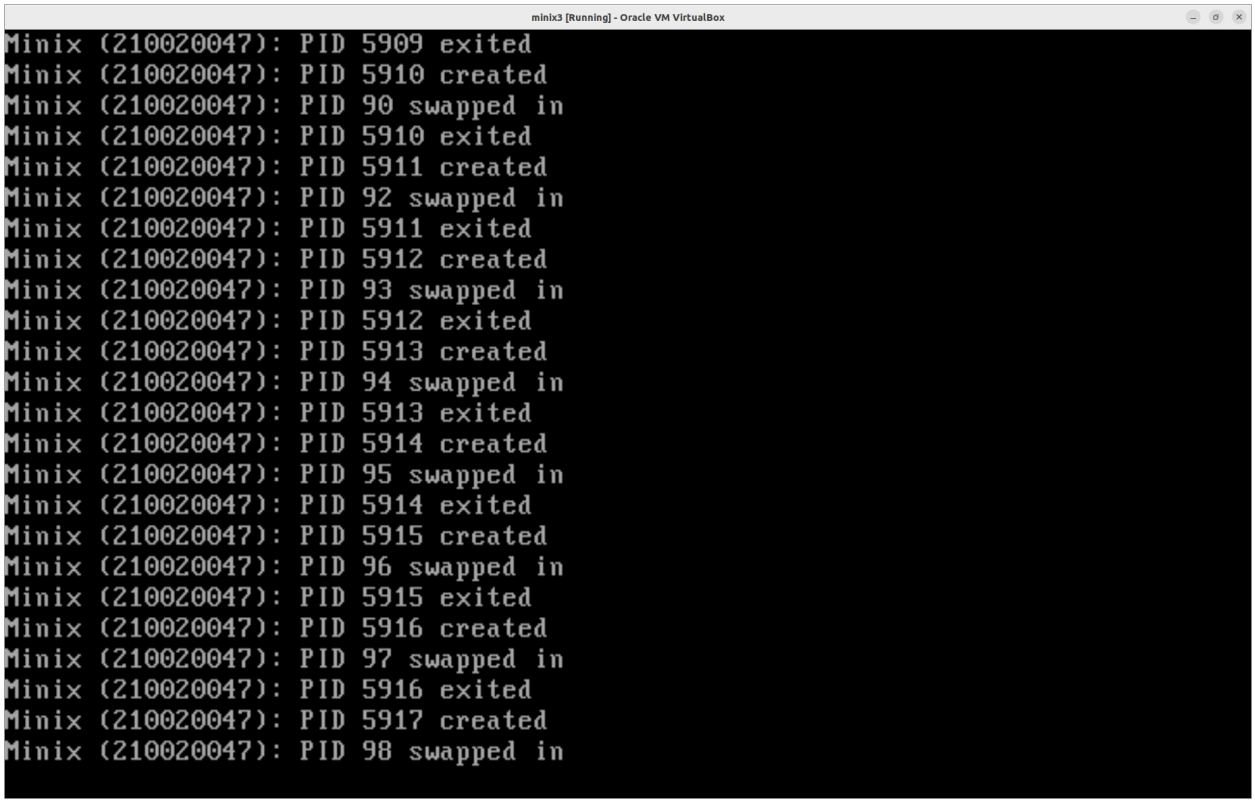
Script: workload_mix4.sh

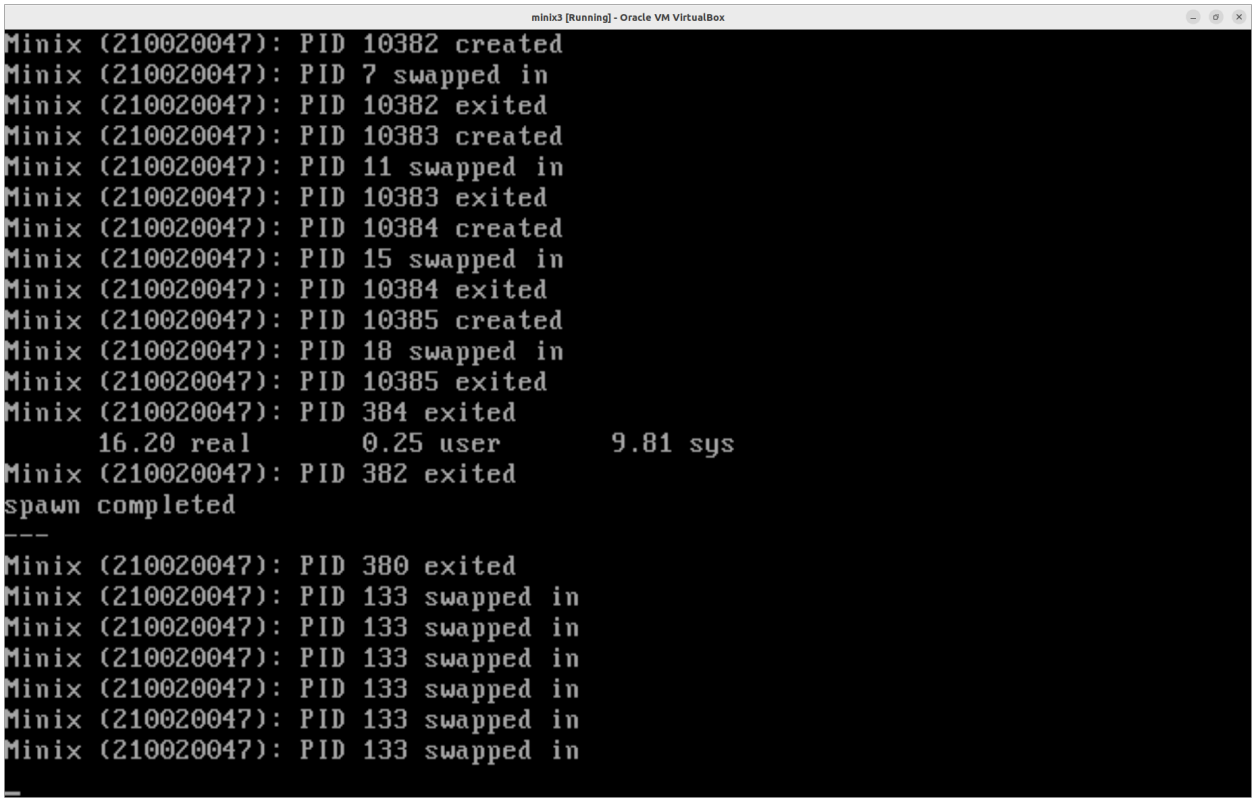
```
./arithoh.sh &
./spawn.sh &
wait
```

Both the processes are CPU-bound. The PID of **spawn** lies in a big range as it repeatedly creates various child processes, while the PID of **arithoh** is 133. Initially, arithoh was scheduled, then it got terminated because it ran for its whole quantum, and then spawn, which was far higher in priority, was scheduled. Since spawn is also CPU-bound, it never got blocked because of IO events, and it kept getting schedules repeatedly until it was over.

After spawn was over, the execution of arithoh started, and there was no other process in the queue, it kept getting rescheduled and finished in the end.


minix3 [Running] - Oracle VM VirtualBox
Minix (210020047): PID 2641 created
Minix (210020047): PID 118 swapped in
Minix (210020047): PID 2641 exited
Minix (210020047): PID 2642 created
Minix (210020047): PID 119 swapped in
Minix (210020047): PID 2642 exited
Minix (210020047): PID 2643 created
Minix (210020047): PID 120 swapped in
Minix (210020047): PID 2643 exited
Minix (210020047): PID 2644 created
Minix (210020047): PID 121 swapped in
Minix (210020047): PID 2644 exited
Minix (210020047): PID 2645 created
Minix (210020047): PID 122 swapped in
Minix (210020047): PID 2645 exited
Minix (210020047): PID 2646 created
Minix (210020047): PID 123 swapped in
Minix (210020047): PID 2646 exited
Minix (210020047): PID 2647 created
Minix (210020047): PID 124 swapped in
Minix (210020047): PID 2647 exited
Minix (210020047): PID 2648 created
Minix (210020047): PID 125 swapped in
Minix (210020047): PID 2648 exited
Minix (210020047): PID 2649 created


minix3 [Running] - Oracle VM VirtualBox
Minix (210020047): PID 5909 exited
Minix (210020047): PID 5910 created
Minix (210020047): PID 90 swapped in
Minix (210020047): PID 5910 exited
Minix (210020047): PID 5911 created
Minix (210020047): PID 92 swapped in
Minix (210020047): PID 5911 exited
Minix (210020047): PID 5912 created
Minix (210020047): PID 93 swapped in
Minix (210020047): PID 5912 exited
Minix (210020047): PID 5913 created
Minix (210020047): PID 94 swapped in
Minix (210020047): PID 5913 exited
Minix (210020047): PID 5914 created
Minix (210020047): PID 95 swapped in
Minix (210020047): PID 5914 exited
Minix (210020047): PID 5915 created
Minix (210020047): PID 96 swapped in
Minix (210020047): PID 5915 exited
Minix (210020047): PID 5916 created
Minix (210020047): PID 97 swapped in
Minix (210020047): PID 5916 exited
Minix (210020047): PID 5917 created
Minix (210020047): PID 98 swapped in


minix3 [Running] - Oracle VM VirtualBox
Minix (210020047): PID 10382 created
Minix (210020047): PID 7 swapped in
Minix (210020047): PID 10382 exited
Minix (210020047): PID 10383 created
Minix (210020047): PID 11 swapped in
Minix (210020047): PID 10383 exited
Minix (210020047): PID 10384 created
Minix (210020047): PID 15 swapped in
Minix (210020047): PID 10384 exited
Minix (210020047): PID 10385 created
Minix (210020047): PID 18 swapped in
Minix (210020047): PID 10385 exited
Minix (210020047): PID 384 exited
16.20 real 0.25 user 9.81 sys
Minix (210020047): PID 382 exited
spawn completed

Minix (210020047): PID 380 exited
Minix (210020047): PID 133 swapped in
Minix (210020047): PID 133 swapped in
Minix (210020047): PID 133 swapped in
Minix (210020047): PID 133 swapped in
Minix (210020047): PID 133 swapped in
Minix (210020047): PID 133 swapped in