Operating Systems Laboratory

Lab 4

All the screenshots are available here:
https://drive.google.com/drive/folders/1vJGCLrT1kFVM444FKDodfmhuZBqsoEg1?usp=sharing

Part 1

The behaviour of the minix scheduler was studied using some workload minix created using UnicBench Benchmarks. The benchmarks range from Completely CPU intensive to completely IO intensive. I also added some print statements in system.c (inside minix/kernel/) to get some idea about how much quanta the process is using. I only used a combination of arithoh and fstime, as they were enough to get to know about the nuances of the minix scheduler.

Script 1:

```
#!/bin/sh
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
Wait
```

Observation:
This script contains arithoh, which is a very CPU-bound task. Here we observed that the scheduler was scheduling the processes in a round-robin fashion. The processes were scheduled one after another and all had the same priority, as we can see repeated scheduling of all the processes with the same quanta one after another.

We can also observe the evidence of round-robin scheduling when we see the time taken to complete the task.

Shubh Agarwal

```
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 231 created
Minix (210020047): PID 206 Quantum: 200 200)
Minix (210020047): PID 232 created
Minix (210020047): PID 207 Quantum: 200 200)
Minix (210020047): PID 233 created
Minix (210020047): PID 208 Quantum: 200 200)
Minix (210020047): PID 234 created
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 235 created
Minix (210020047): PID 210 Quantum: 200 200)
Minix (210020047): PID 236 created
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 237 created
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 238 created
Minix (210020047): PID 213 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 210 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 213 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 210 Quantum: 200 200)
```

```
http://wiki.minix3.org/UsersGuide/PostInstallation
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically
For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org

We'd like your feedback: http://minix3.org/community/

# cd /home/shubh/OSLab/Lab03/part2/UnixBench/workload_mix/
# ./lab5_workload_mix1.sh
        21.08 real          4.40 user          0.00 sys
arithoh completed
---
        21.36 real          4.38 user          0.00 sys
arithoh completed
---
        21.48 real          4.45 user          0.00 sys
arithoh completed
---
        21.78 real          4.40 user          0.00 sys
arithoh completed
---
        22.08 real          4.41 user          0.00 sys
arithoh completed
---
#
```

3

Script 2:

```
#!/bin/sh
./arithoh.sh &
./arithoh.sh &
./fstime.sh &
./arithoh.sh &
./arithoh.sh &
wait
```

Observation: This process contains one IO-Bound process that is fstime. Rest all are CPU-Bound i.e. arithoh. All the processes were initially assigned the same priority and were running in a round-robin fashion. But soon, the scheduler figured out that fstime is an IO-bound process and gave it a higher priority with a bigger quantum of 500 ms. Since fstime was given a higher priority, it ended up executing first. Some runs of fstime were not even utilizing the complete quanta of 500 ms. The remaining arithoh processes were running in the regular round-robin manner after fstime got executed.

```
Minix (210020047): PID 273 exited
Minix (210020047): PID 35 Quantum: 200 200)
Minix (210020047): PID 33 Quantum: 200 200)
Minix (210020047): PID 286 exited
     20.61 real         4.60 user         0.00 sys
Minix (210020047): PID 281 exited
arithoh completed
---
Minix (210020047): PID 276 exited
Minix (210020047): PID 285 exited
     20.76 real         4.56 user         0.00 sys
Minix (210020047): PID 280 exited
arithoh completed
---
Minix (210020047): PID 275 exited
Minix (210020047): PID 11 Quantum: 200 200)
Minix (210020047): PID 11 Quantum: 200 200)
Minix (210020047): PID 282 exited
     21.15 real         4.56 user         0.01 sys
Minix (210020047): PID 277 exited
arithoh completed
---
Minix (210020047): PID 272 exited
Minix (210020047): PID 271 exited
#
```

Shubh Agarwal

```
Write done: 1008000 in 0.7167, score 351628
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically
TIME:0.7
Read done: 1000004 in 0.6500, score 384616
COUNT:384616:0:KBps
TIME:0.6
Copy done: 1000004 in 1.4500, score 172414
COUNT:172414:0:KBps
TIME:1.5
        14.70 real        0.30 user        2.51 sys
fstime completed
---
        20.16 real        4.43 user        0.00 sys
arithoh completed
---
        20.48 real        4.43 user        0.00 sys
arithoh completed
---
        20.55 real        4.48 user        0.03 sys
arithoh completed
---
        20.65 real        4.43 user        0.00 sys
arithoh completed
---
#
```

Script 3:

```
#!/bin/sh
./arithoh.sh &
./fstime.sh &
./arithoh.sh &
./fstime.sh &
./arithoh.sh &
wait
```

Observation: Here, the number of IO-bound processes was increased. Again, results
similar to upper scripts were observed. The IO-bound fstime was given more priority
when compared to CPU-bound arithoh with a higher time slice. However, since there
were two IO-bound processes here, a round-robin policy was observed among them.
Where there was some time left between these processes, the CPU-bound process
arithoh was getting scheduled. Finally, the fstime got finished first, then the
CPU-bound arithoh was finished.

```
File  Machine  View  Input  Devices  Help
Minix (210020047): PID 51 Quantum: 200 200)
Minix (210020047): PID 49 Quantum: 200 200)
Minix (210020047): PID 51 Quantum: 200 200)
Minix (210020047): PID 47 Quantum: 200 200)
Minix (210020047): PID 51 Quantum: 200 200)
Minix (210020047): PID 47 Quantum: 200 200)
Minix (210020047): PID 49 Quantum: 200 200)
Minix (210020047): PID 47 Quantum: 200 200)
Minix (210020047): PID 49 Quantum: 200 200)
Minix (210020047): PID 51 Quantum: 200 200)
Minix (210020047): PID 47 Quantum: 17 200)
Minix (210020047): PID 49 Quantum: 0 200)
Minix (210020047): PID 51 Quantum: 0 200)
Minix (210020047): PID 24 Quantum: 500 500)
Read done: 1000004 in 1.4333, score 174419
Read done: 1000004 in 1.4333, score 174419
COUNT¦174419¦0¦KBps
COUNT¦174419¦0¦KBps
TIME¦1.4
TIME¦1.4
Minix (210020047): PID 49 Quantum: 200 200)
Minix (210020047): PID 51 Quantum: 200 200)
Minix (210020047): PID 47 Quantum: 200 200)
Minix (210020047): PID 51 Quantum: 200 200)
```
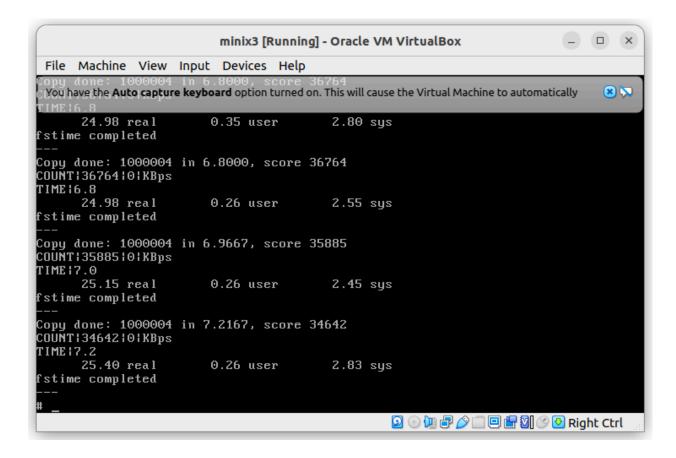
```
COUNT¦174419¦0¦KBps
  You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically
TIME¦1.4
Copy done: 1000004 in 2.6833, score 93168
COUNT¦93168¦0¦KBps
TIME¦2.7
      17.20 real         0.33 user         2.60 sys
fstime completed
---
Copy done: 1000004 in 2.8667, score 87209
COUNT¦87209¦0¦KBps
TIME¦2.9
      17.38 real         0.35 user         2.45 sys
fstime completed
---
      17.63 real         4.45 user         0.00 sys
arithoh completed
---
      18.48 real         4.53 user         0.00 sys
arithoh completed
---
      19.21 real         4.50 user         0.00 sys
arithoh completed
---
#
```

Script 4:

```sh
#!/bin/sh
./fstime.sh &
./fstime.sh &
./fstime.sh &
```

Shubh Agarwal

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

Observation: All the processes in this script are fstime, which means completely
IO-bound. All the processes started with 200 quanta and then were promoted in the
MLQ with a higher time slice. They are then executed in a round-robin fashion.

Part 2

Changes made in schedule.c are:

Added new_prio = 1; in schedule_process()
Commented the following lines:

```
// if (rmp->priority < MIN_USER_Q) {
//   rmp->priority += 1; /* lower priority */
// }
```

in do_noquantum(), and:

```
// rmp->priority -= 1; /* increase priority */
```

in balance_queues().

These were the changes made in schedule.c to implement the pseudo-FIFO policy in Minix.

How is it replicating FIFO..?
When the process gets forked, the new process gets the base priority of its parent process (near 7). Once it completes its quanta, it again gets rescheduled with the priority of 1. Now a new process comes into the picture, it will have a priority somewhere near 7, so it will not get the chance to execute as there is already a process with priority 1. This way, the process that is coming first is getting scheduled first.

Also, I disabled the balancing of queues as the process has already been assigned the priority of 1. Decreasing it should not be necessary.
Also, I disabled the priority lowering in do_noquantum(), as we wanted to closely replicate FIFO. This kind of optimisation is done in MLQ, which is not required.

Shubh Agarwal

Script 1:

```
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 211 Quantum: 200 200)
Minix (210020047): PID 10274 created with endpoint 211
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
Minix (210020047): PID 205 Quantum: 200 200)
```

```
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 209 Quantum: 200 200)
Minix (210020047): PID 10272 created with endpoint 209
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
Minix (210020047): PID 212 Quantum: 200 200)
```

210020047

```
clang -o pgms/fstime -Wall -pedantic -O0 -ffast-math -I ./src -DTIME src/fstime.
c
clang -o pgms/whetstone-double -Wall -pedantic -O0 -ffast-math -I ./src -DTIME -
DDP -DGTODay -DUNIXBENCH src/whets.c -lm
gmake[1]: Leaving directory '/home/shubh/OSLab/Lab03/part2/UnixBench'
# cd workload_mix/
# ./lab5_workload_mix1.sh
        4.41 real        17.68 real        22.11 real        22.11 real        22.11 real
        4.41 user         4.43 user         4.41 user         4.41 user         4.43 user
        0.00 sys
arithoh completed
        0.00---
 sys
        0.00arithoh completed
 sys
        0.00---
arithoh completed
 sys
---
arithoh completed
---
        0.00 sys
arithoh completed
---
# _
```

Here the processes were executed in a sequential order i.e. one after another. Unlike with the unmodified scheduler, where a round-robin policy was being used. This is the case where our scheduler most closely resembles the FIFO scheduler.

Script 2:

```
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 225 Quantum: 200 200)
Minix (210020047): PID 10288 created with endpoint 225
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
```

Shubh Agarwal

```
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 227 Quantum: 200 200)
Minix (210020047): PID 10290 created with endpoint 227
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
Minix (210020047): PID 221 Quantum: 200 200)
```

```
       4.40 user        4.41 user        4.38 user        0.00 sys
arithoh completed
       0.00---
 sys
       0.00arithoh completed
 sys
---
arithoh completed
---
       0.00 sys
arithoh completed
---
Write done: 1008000 in 0.7833, score 321701
COUNT:321701:0:KBps
TIME:0.8
Read done: 1000004 in 0.7167, score 348838
COUNT:348838:0:KBps
TIME:0.7
Copy done: 1000004 in 1.5667, score 159575
COUNT:159575:0:KBps
TIME:1.6
      31.78 real        0.20 user        2.86 sys
fstime completed
---
#
```

Here, the results were much different from the previous scheduler. The arithoh processes were finished first. Then the fstime was scheduled. Actually, it was once scheduled in the middle, but then it was terminated because it was waiting for the IO. Then the FIFO property of our scheduler took over and all the arithoh were completed first. Hence our scheduler is pseudo FIFO.

Script 3:

```
Minix (210020047): PID 10303 created with endpoint 240
Minix (210020047): PID 240 Quantum: 200 200)
Minix (210020047): PID 10304 created with endpoint 241
Minix (210020047): PID 241 Quantum: 200 200)
Minix (210020047): PID 10305 created with endpoint 242
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 10306 created with endpoint 243
Minix (210020047): PID 243 Quantum: 200 200)
Minix (210020047): PID 10307 created with endpoint 244
Minix (210020047): PID 244 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
Minix (210020047): PID 242 Quantum: 200 200)
```

```
---
# ./lab5_workload_mix3.sh
      13.23 real        13.23 real        13.23 real           4.41 user        4.41 user
       4.40 user         0.00 sys
arithoh completed
---
       0.00 sys
arithoh completed
---
       0.00 sys
arithoh completed
---
Write done: 1008000 in 1.7000, score 148235
COUNT:148235:0:KBps
Write done: 1008000 in 1.7000, score 148235
TIME:1.7
COUNT:148235:0:KBps
TIME:1.7
Read done: 1000004 in 1.5667, score 159575
COUNT:159575:0:KBps
Read done: 1000004 in 1.5667, score 159575
TIME:1.6
COUNT:159575:0:KBps
TIME:1.6
```

Shubh Agarwal

```
Write done: 1008000 in 1.7000, score 148235
COUNT|148235|0|KBps
Write done: 1008000 in 1.7000, score 148235
TIME|1.7
COUNT|148235|0|KBps
TIME|1.7
Read done: 1000004 in 1.5667, score 159575
COUNT|159575|0|KBps
Read done: 1000004 in 1.5667, score 159575
TIME|1.6
COUNT|159575|0|KBps
TIME|1.6
Copy done: 1000004 in 3.2833, score 76142
COUNT|76142|0|KBps
Copy done: 1000004 in 3.2833, score 76142
TIME|3.3
COUNT|76142|0|KBps
TIME|3.3
        30.80 real        30.80 real        0.30 user        0.41 user        2.76 sys
fstime completed
---
        3.06 sys
fstime completed
---
#
```

Here, the results are similar to the last script on this scheduler. All the arithohs processes were completed first, and then fstime got executed. It is completely different from what we observed in the MLQ scheduler.

Script 4:

```
Minix (210020047): PID 10312 created with endpoint 249
Minix (210020047): PID 249 Quantum: 200 200)
Minix (210020047): PID 10313 created with endpoint 250
Minix (210020047): PID 250 Quantum: 200 200)
Minix (210020047): PID 10314 created with endpoint 251
Minix (210020047): PID 251 Quantum: 200 200)
Minix (210020047): PID 10315 created with endpoint 252
Minix (210020047): PID 252 Quantum: 200 200)
Minix (210020047): PID 10316 created with endpoint 253
Minix (210020047): PID 253 Quantum: 200 200)
Minix (210020047): PID 10317 created with endpoint 254
Minix (210020047): PID 254 Quantum: 200 200)
Minix (210020047): PID 10318 created with endpoint 255
Minix (210020047): PID 255 Quantum: 200 200)
Minix (210020047): PID 10319 created with endpoint 7
Minix (210020047): PID 7 Quantum: 200 200)
Minix (210020047): PID 10320 created with endpoint 11
Minix (210020047): PID 11 Quantum: 200 200)
Minix (210020047): PID 10321 created with endpoint 15
Minix (210020047): PID 15 Quantum: 200 200)
Minix (210020047): PID 10322 created with endpoint 18
Minix (210020047): PID 18 Quantum: 200 200)
Minix (210020047): PID 10323 created with endpoint 33
Minix (210020047): PID 33 Quantum: 200 200)
```

210020047

```
COUNT:42954:0:KBps
Write done: 1008000 in 5.8667, score 42954
TIME:5.9
COUNT:42954:0:KBps
Write done: 1008000 in 5.8667, score 42954
TIME:5.9
COUNT:42954:0:KBps
TIME:5.9
Minix (210020047): PID 24 Quantum: 500 500)
Read done: 1000004 in 5.2333, score 47770
COUNT:47770:0:KBps
Read done: 1000004 in 5.2333, score 47770
TIME:5.2
COUNT:47770:0:KBps
Read done: 1000004 in 5.2333, score 47770
TIME:5.2
COUNT:47770:0:KBps
Read done: 1000004 in 5.2333, score 47770
TIME:5.2
COUNT:47770:0:KBps
Read done: 1000004 in 5.2333, score 47770
TIME:5.2
COUNT:47770:0:KBps
TIME:5.2
```

```
COUNT:30426:0:KBps
Copy done: 1000004 in 8.2167, score 30426
TIME:8.2
COUNT:30426:0:KBps
Copy done: 1000004 in 8.2167, score 30426
TIME:8.2
COUNT:30426:0:KBps
TIME:8.2
      27.43 real        27.43 real        27.43 real        27.43 real        0.26 user
      27.43 real         0.30 user         0.36 user         0.20 user        2.85 sys
        0.35fstime completed
 user---
      2.95 sys
fstime completed
---
      3.36 sys
fstime completed
      2.78---
 sys
fstime completed
      3.00---
 sys
fstime completed
---
#
```

In this benchmark, the FIFO nature of the scheduler was of very little use. The processes were being stopped because they were waiting for the IO. Once they were waiting, other processes in the queue were getting spawned. As a result, after some time, all the processes had the same priority. The fashion observer was FIFO at the start, which was eventually converted to round robin.

Shubh Agarwal