

Operating Systems Lab

Assignment 3

180010002

Part I. In this, the task is to modify the minix3 source code to print “PID <pid> swapped in”, everytime a user-level process is brought in by the scheduler. Figure 1 shows the modified code, a few lines have been added to the file *minix/servers/sched/schedule.c* in the function *schedule_process*.

```
/*=====*
 *               schedule_process               *
 *=====*/
static int schedule_process(struct schedproc * rmp, unsigned flags)
{
    int err;
    int new_prio, new_quantum, new_cpu;

    pick_cpu(rmp);

    // to print whenever a user level process is brought in
    if(rmp->priority >= USER_Q) {
        printf("Minix 3: PID %d swapped in\n",
            _ENDPOINT_P(rmp->endpoint));
    }
}
```

Figure 1. Schedule.c

Command: ./run.sh

After the script is run and system is rebooted, the desirable output could be seen in the minix VM.

Part II. In this, the task is to study the behaviour of the scheduler by seeing the sequence of PID prints when different workload_mixes are run. The following are the different observations on running different workload_mixes:-

(i) workload_mix.sh

- In this, we run **2 arithoh.sh** parallelly and observe the output.
- We know that arithoh is computationally intensive for the CPU, that is why we see them getting scheduled alternatively in figure 3. We also run only one **arithoh.sh**(Figure 2) to see what we observe from that, we clearly see that it is computationally intensive, since it is

scheduled several times and we also see that sys time taken by this is almost 0.

```
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
Minix 3: PID 197 swapped in
20.41 real    20.40 user    0.01 sys
arithoh completed
---
#
```

Figure 2. Arithoh.sh

```
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
Minix 3: PID 209 swapped in
Minix 3: PID 208 swapped in
```

Figure 3. Two Arithoh.sh running parallelly

(ii) workload_mix1.sh

- In this, we run **arithoh.sh** and **fstime.sh** parallelly and observe the output.
- We know that **fstime.sh** is IO intensive process for the CPU, that is why we see **arithoh** getting scheduled several times in the meantime

fstime waits for its IO operations to be completed in figure 5. We can clearly see that scheduler efficiently utilizes the wait time of IO process like fstime and schedules a computationally intensive process like arithoh. We also see that the sys time in fstime is larger compared to arithoh (Figure 4).

```

./fstime.sh
Minix 3: PID 210 swapped in
Minix 3: PID 211 swapped in
Minix 3: PID 212 swapped in
Write done: 1008000 in 1.5333, score 164347
COUNT:164347:0:KBps
TIME:1.5
Read done: 1000004 in 1.3333, score 187500
COUNT:187500:0:KBps
TIME:1.3
Minix 3: PID 212 swapped in
Copy done: 1000004 in 3.2000, score 78125
COUNT:78125:0:KBps
TIME:3.2
      17.10 real          0.56 user          5.45 sys
fstime completed
---
```

Figure 4. fstime.sh

[illegible]

Figure 5. fstime.sh and arithoh.sh running parallelly

(iii) workload mix2.sh

- In this, we run **arithoh.sh** and **pipe.sh** parallelly and observe the output.
- We know that pipe.sh is CPU bound process, that is why we see them getting scheduled alternatively in figure 7 with pipe finishing first,

however pipe uses more sys time because inter process communication protocols are system bases highly. We also see that the user time in pipe is much smaller compared to arithoh (Figure 6).

```
# ./pipe.sh
Minix 3: PID 219 swapped in
Minix 3: PID 220 swapped in
Minix 3: PID 221 swapped in
Minix 3: PID 221 swapped in
Minix 3: PID 221 swapped in
Minix 3: PID 221 swapped in
      10.51 real      0.78 user      9.73 sys
pipe completed
---
```

Figure 6. pipe.sh

```
Minix 3: PID 227 swapped in
Minix 3: PID 226 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 226 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 9 swapped in
Minix 3: PID 226 swapped in
Minix 3: PID 227 swapped in
      11.71 real      0.95 user      9.90 sys
pipe completed
---
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 227 swapped in
```

Figure 7. pipe.sh and arithoh.sh running parallelly

(iv) workload_mix3.sh

- In this, we run **arithoh.sh** and **spawn.sh** parallelly and observe the output.
- We see similar observations in this as we saw in previous one (workload_mix2.sh). Here also spawn finishes first (figure 9) and sys time is higher in spawn too like in pipe. However spawn calls in a lot of processes ranging from 7 to 255 as we can see in figure 8.

```

Minix 3: PID 247 swapped in
Minix 3: PID 248 swapped in
Minix 3: PID 249 swapped in
Minix 3: PID 250 swapped in
Minix 3: PID 251 swapped in
Minix 3: PID 252 swapped in
Minix 3: PID 253 swapped in
Minix 3: PID 254 swapped in
Minix 3: PID 255 swapped in
Minix 3: PID 7 swapped in
Minix 3: PID 11 swapped in
Minix 3: PID 15 swapped in
Minix 3: PID 18 swapped in
Minix 3: PID 33 swapped in
Minix 3: PID 35 swapped in
Minix 3: PID 36 swapped in
Minix 3: PID 37 swapped in
Minix 3: PID 38 swapped in
Minix 3: PID 39 swapped in
Minix 3: PID 40 swapped in
Minix 3: PID 41 swapped in
      8.31 real      0.50 user      6.46 sys
spawn completed
---
#

```

Figure 8. spawn.sh

```

Minix 3: PID 226 swapped in
Minix 3: PID 227 swapped in
Minix 3: PID 228 swapped in
Minix 3: PID 229 swapped in
Minix 3: PID 230 swapped in
Minix 3: PID 231 swapped in
Minix 3: PID 232 swapped in
Minix 3: PID 233 swapped in
Minix 3: PID 234 swapped in
Minix 3: PID 235 swapped in
Minix 3: PID 236 swapped in
      9.13 real      0.31 user      6.98 sys
spawn completed
---
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 47 swapped in

```

Figure 9. spawn.sh and arithoh.sh running parallelly

(v) workload mix4.sh

- In this, we run **arithoh.sh** and **syscall.sh** parallelly and observe the output.
- We are calling the system sitting in a loop in syscall, that is why we see sys time higher in syscall and we also see switching between arithoh and syscall processes since syscall is also cpu bound process (figure 10 and 11). In this also arithoh completes last.

```
# ./syscall.sh
Minix 3: PID 237 swapped in
Minix 3: PID 238 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
Minix 3: PID 239 swapped in
7.60 real      2.13 user      5.43 sys
syscall completed
---
```

Figure 10. syscall.sh

```
Minix 3: PID 245 swapped in
Minix 3: PID 244 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 244 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 244 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 244 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 244 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 244 swapped in
Minix 3: PID 245 swapped in
9.43 real      2.45 user      5.23 sys
syscall completed
---
Minix 3: PID 245 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 245 swapped in
Minix 3: PID 245 swapped in
```

Figure 11. syscall.sh and arithoh.sh running parallelly

(vi) workload mix5.sh

- In this, we run **all the scripts (spawn, arithoh, fstime, pipe and syscall) 2 times** parallelly and observe the output.
- We observe the finish times of all the processes and see that the order in which the processes finish is (order of increasing finishing time): spawn < fstime < pipe < syscall < arithoh (figures 12-16).
- We observe that the processes finish in the order of user time they take, the which takes most of user time finishes last and the one which takes least user time finishes first (can be seen in figures 12-16).

```

Minix 3: PID 189 swapped in
Minix 3: PID 190 swapped in
Minix 3: PID 191 swapped in
Minix 3: PID 192 swapped in
Minix 3: PID 193 swapped in
      21.61 real      0.30 user      6.25 sys
spawn completed
---
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Write done: 1008000 in 22.0667, score 11419
COUNT:11419:0:KBps
TIME:22.1
Write done: 1008000 in 22.0667, score 11419
COUNT:11419:0:KBps
TIME:22.1
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 52 swapped in

```

Figure 12.

```

Minix 3: PID 47 swapped in
Minix 3: PID 44 swapped in
Minix 3: PID 50 swapped in
Minix 3: PID 23 swapped in
Copy done: 1000004 in 10.5667, score 23659
COUNT:23659:0:KBps
TIME:10.6
      47.75 real      0.63 user      5.48 sys
fstime completed
---
Minix 3: PID 41 swapped in
Minix 3: PID 44 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 50 swapped in
Minix 3: PID 52 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Copy done: 1000004 in 8.9833, score 27829
COUNT:27829:0:KBps
TIME:9.0
      50.98 real      0.51 user      5.80 sys
fstime completed
---

```

Figure 13.

```

Minix 3: PID 41 swapped in
Minix 3: PID 44 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 50 swapped in
Minix 3: PID 52 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 9 swapped in
1:03.03 real      1.11 user      10.85 sys
pipe completed
---
1:03.18 real      1.13 user      10.50 sys
pipe completed
---
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 47 swapped in
Minix 3: PID 52 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 52 swapped in

```

Figure 14.

```

Minix 3: PID 52 swapped in
Minix 3: PID 52 swapped in
1:15.23 real      2.65 user      5.51 sys
syscall completed
---
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
1:15.60 real      2.40 user      5.40 sys
syscall completed
---
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in

```

Figure 15.


```

Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 48 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 48 swapped in
      1:47.35 real      20.43 user      0.01 sys
arithoh completed
---
Minix 3: PID 41 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 41 swapped in
Minix 3: PID 41 swapped in
      1:48.26 real      20.40 user      0.00 sys
arithoh completed
---
#

```

Figure 16.