

OS LAB - 3

Report

Name: Ashish Kupsad

Roll no: 180020003

Part I:

The code is added in the function ***schedule_process()*** present in the file *minix/servers/sched/schedule.c*.

```
if(rmp->priority >= 7)
printf("Minix: PID %d swapped in\n",
_ENDPOINT_P(rmp->endpoint));
```

Part II:

1. arithoh.sh (CPU Bound)

Observations:

a. Running alone:

- When we run arithoh.sh alone, the real and user times taken are almost same. So, the kernel scheduler log indicates that the instruction to schedule Arithoh was sent 92 times consecutively.

```

Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 196 swapped in
Minix: PID 221 exited
      12.05 real      12.00 user      0.03 sys
Minix: PID 220 exited
arithoh completed
---
Minix: PID 219 exited
#

```

b. Running parallelly:

- We can observe that, it is following a round-robin kind of scheduling with kernel scheduler and our given print statements.
- Moreover, while both the processes start at the same time, the 2nd one takes around 9 seconds more time than the first one to begin its execution.

2. fstime.sh (IO Bound)

Observations:

a. Running alone:

- Since it looks like an IO Bound Benchmark process, the user time is less than sys time (moderate) which is less than the total turnaround time (which is largest).

- This occurs because, the process needs to wait for the completion of IO before executing.

```
# ./fstime.sh
Minix: PID 224 created
Minix: PID 199 swapped in
Minix: PID 225 created
Minix: PID 200 swapped in
Minix: PID 226 created
Minix: PID 201 swapped in
Write done: 1008000 in 0.6833, score 368780
COUNT:368780:0:KBps
TIME:0.7
Read done: 1000004 in 0.6500, score 384616
COUNT:384616:0:KBps
TIME:0.7
Copy done: 1000004 in 1.4833, score 168540
COUNT:168540:0:KBps
TIME:1.5
Minix: PID 226 exited
      14.28 real      0.21 user      2.63 sys
Minix: PID 225 exited
fstime completed
---
```

b. Running ./arithoh.sh and fstime.sh together:

- Arithoh.sh keeps repeating for some time when IO is scheduled.
- Clearly, arithoh.sh is trying to utilize the CPU when, fstime is waiting for IO. This increases the efficiency of scheduler.

3. pipe.sh (CPU Bound)

Observations:

a. Running Alone:

- Lot of time spent by pipe is on sys mode, because the communications between different processes (through pipe) are based on the system itself.

- Time spent in user mode is less, because there's not much user IO in it.

```
# ./pipe.sh
Minix: PID 230 created
Minix: PID 205 swapped in
Minix: PID 231 created
Minix: PID 206 swapped in
Minix: PID 232 created
Minix: PID 207 swapped in
Minix: PID 207 swapped in
Minix: PID 232 exited
          5.55 real          0.40 user          5.13 sys
Minix: PID 231 exited
pipe completed
---
```

b. Running with arithoh.sh together:

- They run similar to [fstime – arithoh] together, but the only difference is the pipe finishes earlier which is followed by consecutive scheduling of arithoh until it completes.

4. spawn.sh (CPU bound)

Observations

a. Running Alone:

- Time spent in sys mode is large (because CPU does most of the work), but total turnaround time (real time) is largest.
- Large number of swaps happening between different processes (in the queue).

```
Minix: PID 10232 created
Minix: PID 243 swapped in
Minix: PID 10232 exited
Minix: PID 10233 created
Minix: PID 244 swapped in
Minix: PID 10233 exited
Minix: PID 10234 created
Minix: PID 245 swapped in
Minix: PID 10234 exited
Minix: PID 10235 created
Minix: PID 246 swapped in
Minix: PID 10235 exited
Minix: PID 10236 created
Minix: PID 247 swapped in
Minix: PID 10236 exited
Minix: PID 10237 created
Minix: PID 248 swapped in
Minix: PID 10237 exited
Minix: PID 236 exited
      13.43 real      0.00 user      6.36 sys
Minix: PID 235 exited
spawn completed
```

b. Running with arithoh together:

- Similar to [pipe-arithoh], spawn completes early and arithoh continues its execution till it completes.

5. syscall.sh (CPU bound)

Observations:

a. Running alone:

- Again, major time spent is on sys mode whereas time spent by user is smaller, but not too small.

```
# ./syscall.sh
Minix: PID 10238 created
Minix: PID 249 swapped in
Minix: PID 10239 created
Minix: PID 250 swapped in
Minix: PID 10240 created
Minix: PID 251 swapped in
Minix: PID 251 swapped in
Minix: PID 251 swapped in
Minix: PID 251 swapped in
Minix: PID 251 swapped in
Minix: PID 10240 exited
          3.81 real          1.18 user          2.61 sys
Minix: PID 10239 exited
syscall completed
```

Running with arithoh together:

- Both follow a round-robin fashion. syscall completes its execution first and then arithoh is completed.