# CS314 Lab-3

*By: Ayush Gupta (190030007)*

## Part-I

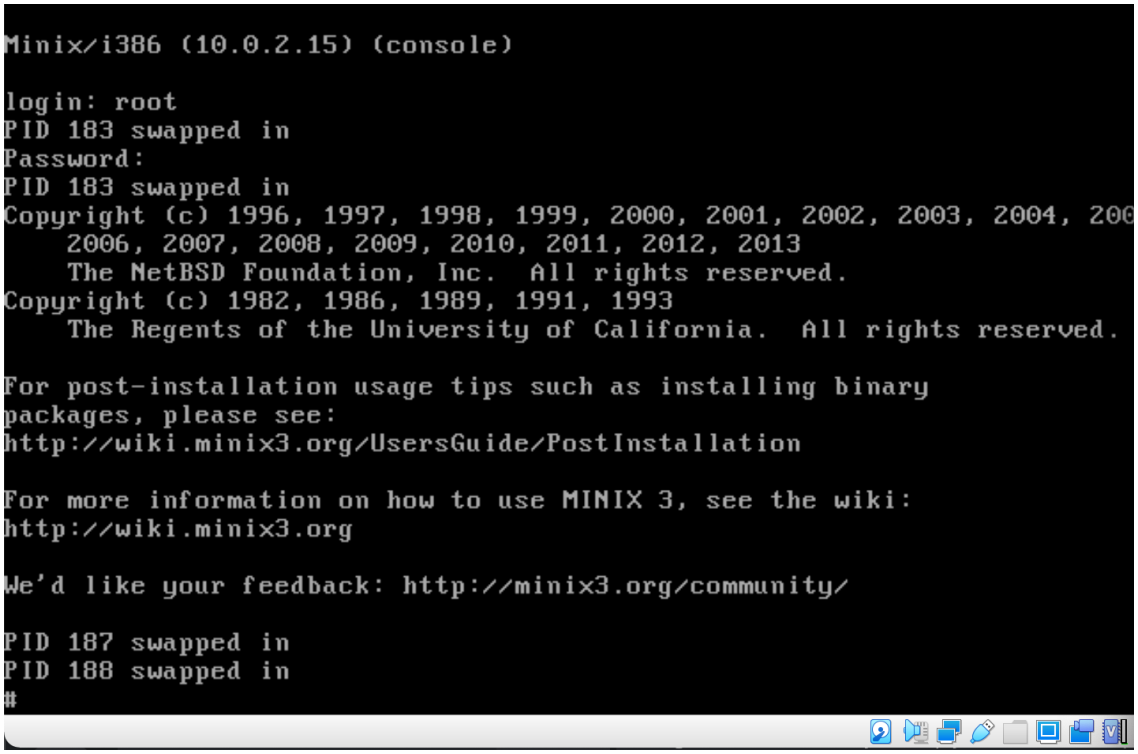
In Part-I, we have to print the string, *"PID <pid> swapped in"* when the scheduler brings a user-level process.

In order to achieve this, we have to edit the file *schedule.c* in and add the printf() statement whenever *rmp->priority >= USER_Q*

This file *scheduler.c* is present in the folder *minix/servers/sched* directory path

After this, we run the make build command.

```
Minix: PID 25057 exited
Minix: PID 25058 created
Minix: PID 25058 exited
Minix: PID 25059 created
Minix: PID 25059 exited
Minix: PID 25060 created
Minix: PID 25060 exited
Minix: PID 25061 created
Minix: PID 25061 exited
Minix: PID 25062 created
Minix: PID 25062 exited
Minix: PID 25007 exited
Minix: PID 24974 exited
Minix: PID 24897 exited
Minix: PID 24896 exited
Minix: PID 25063 created
Build started at:  Sun Jan 30 19:40:20 GMT 2022
Minix: PID 25063 exited
Minix: PID 25064 created
Build finished at: Minix: PID 25065 created
Sun Jan 30 19:44:48 GMT 2022
Minix: PID 25065 exited
Minix: PID 25064 exited
Minix: PID 396 exited
#
```




This two commands are put together in the *run.sh* file

# Part-II

UnixBench Benchmark suite was downloaded from the google drive, and then the gmake command was run.

After that, I created four *workload_mix.sh* files.

1) **workload_mix1.sh**
   In this file, I run *syscall* and *fstime*.

   *syscall* had PID 239 and it took 12.01 seconds

*fstime* had PID 238 and it took 17.91 seconds

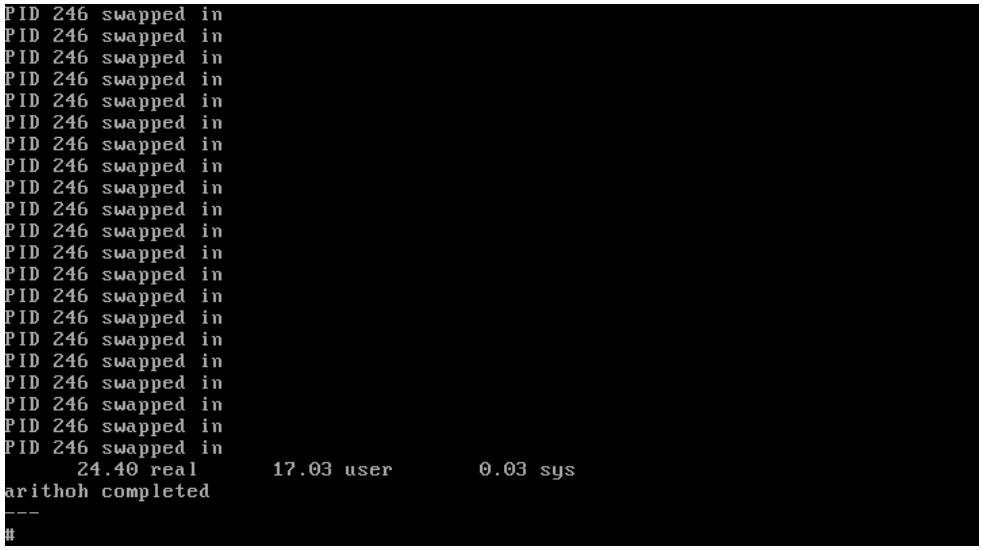
```
PID 239 swapped in
Write done: 1008000 in 1.4167, score 177882
COUNT¦177882¦0¦KBps
TIME¦1.4
PID 239 swapped in
PID 239 swapped in
PID 239 swapped in
PID 239 swapped in
PID 239 swapped in
Read done: 1000004 in 1.6000, score 156250
COUNT¦156250¦0¦KBps
TIME¦1.6
PID 239 swapped in
      12.01 real       2.90 user       6.10 sys
syscall completed
---
PID 238 swapped in
PID 238 swapped in
Copy done: 1000004 in 3.9000, score 64102
COUNT¦64102¦0¦KBps
TIME¦3.9
      17.91 real       0.61 user       6.30 sys
fstime completed
---
#
```

**2) workload_mix2.sh**

In this file, I run *fstime* and *arithoh*

*fstime* had PID 247 and it took 18.56 seconds
*arithoh* had PID 246 and it took 24.40 seconds

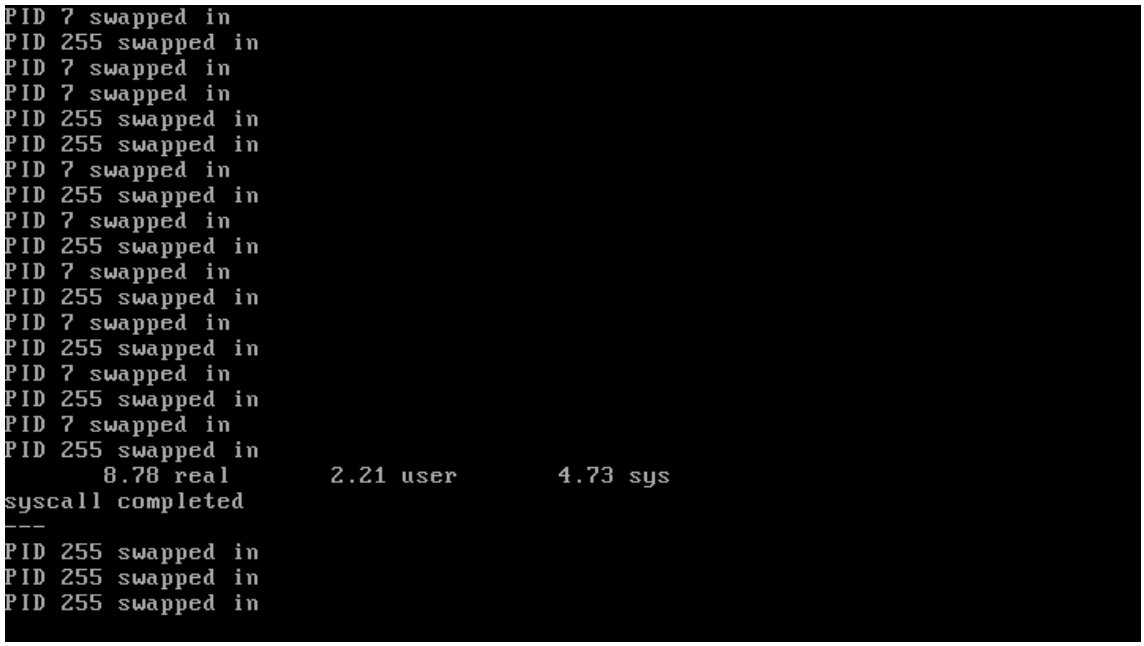
```
PID 246 swapped in
PID 247 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 247 swapped in
PID 246 swapped in
PID 247 swapped in
Copy done: 1000004 in 3.8000, score 65789
COUNT¦65789¦0¦KBps
TIME¦3.8
      18.56 real       0.68 user       6.65 sys
fstime completed
---
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
PID 246 swapped in
_
```




### 3) workload_mix3.sh

In this file, *syscall* and *arithoh*

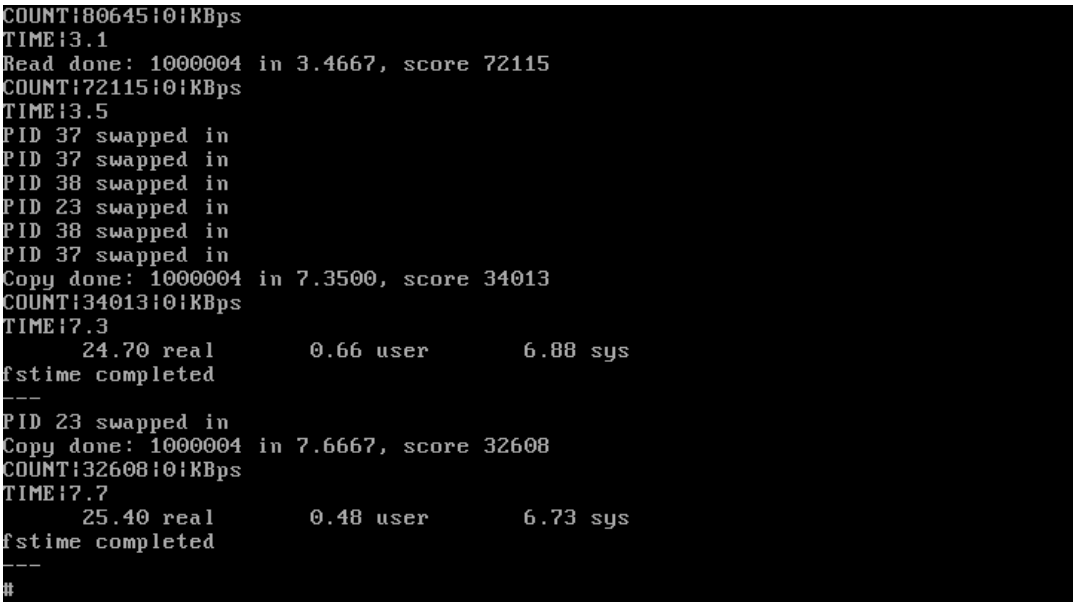
*syscall* had PID 7 and it took 8.78 seconds
*arithoh* had PID 255 and it took 21.05 seconds




**4) workload_mix4.sh**

In this file, I run *fstime* and *fstime* both times

The first *fstime* had PID 37 and it took 24.07 seconds whereas the second *fstime* had PID 23 and it took 25.40 seconds



As we can see, the time taken by each of these processes didn’t vary much every time they were run in different bash files.