

# Operating Systems Lab - CS 314

## Assignment - 3

-KARTHIK.K(200010024)

-CHINTU NUNAVATH(200010033)

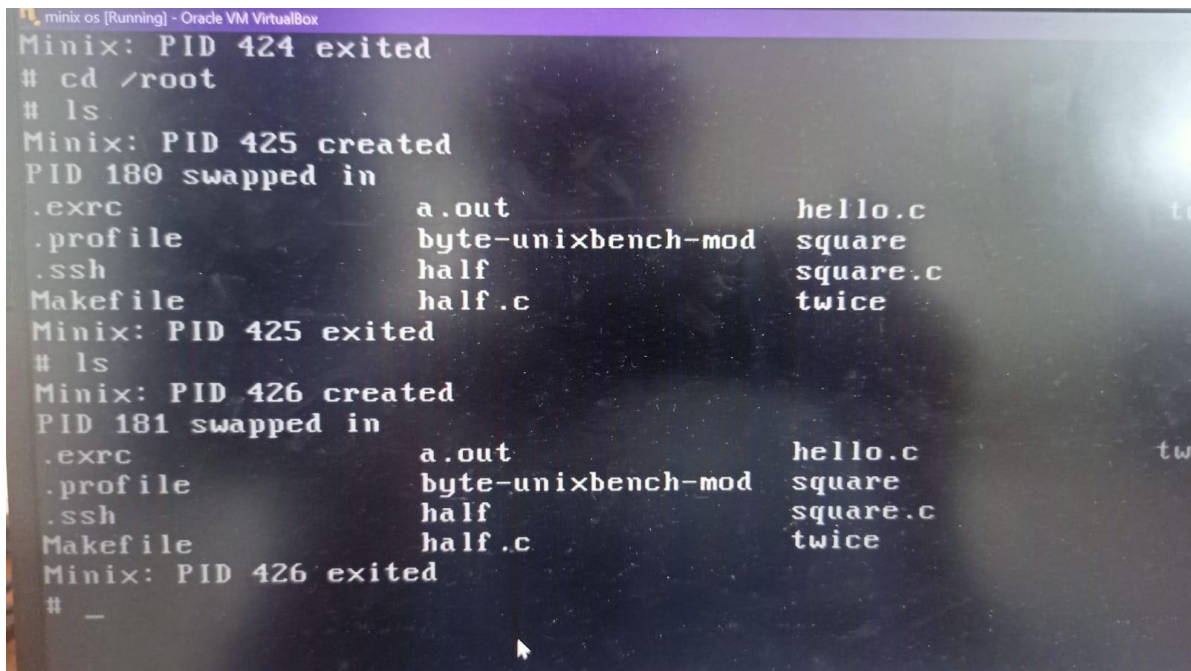
### Part-1

In Part-1, we had to modify the Minix3 source code such that the string “PID swapped in” is printed, whenever a user-level process is brought in by the scheduler. Using the help of Source-1 & Source-2, the file at location: minix/servers/sched called schedule.c. The following piece of code was added in schedule process function:

```
if(is_system_proc(rmp)==0){  
    printf("PID %d swapped in\n", _ENDPOINT_P(rmp->endpoint));  
}
```

In order to deploy the changes on Minix3 VM, a simple script run.sh was written whose main function is to copy schedule.c to correct location and build the system to see changes.

```
cp schedule.c /usr/src/minix/servers/sched/schedule.c  
cd /usr/src && make build MKUPDATE=yes >log.txt 2>log.txt
```



```
minix os [Running] - Oracle VM VirtualBox  
Minix: PID 424 exited  
# cd /root  
# ls  
Minix: PID 425 created  
PID 180 swapped in  
.exrc          a.out          hello.c  
.profile       byte-unixbench-mod square  
.ssh           half          square.c  
Makefile       half.c        twice  
Minix: PID 425 exited  
# ls  
Minix: PID 426 created  
PID 181 swapped in  
.exrc          a.out          hello.c  
.profile       byte-unixbench-mod square  
.ssh           half          square.c  
Makefile       half.c        twice  
Minix: PID 426 exited  
# _
```

After a successful build, Figure shows the effective changes done.

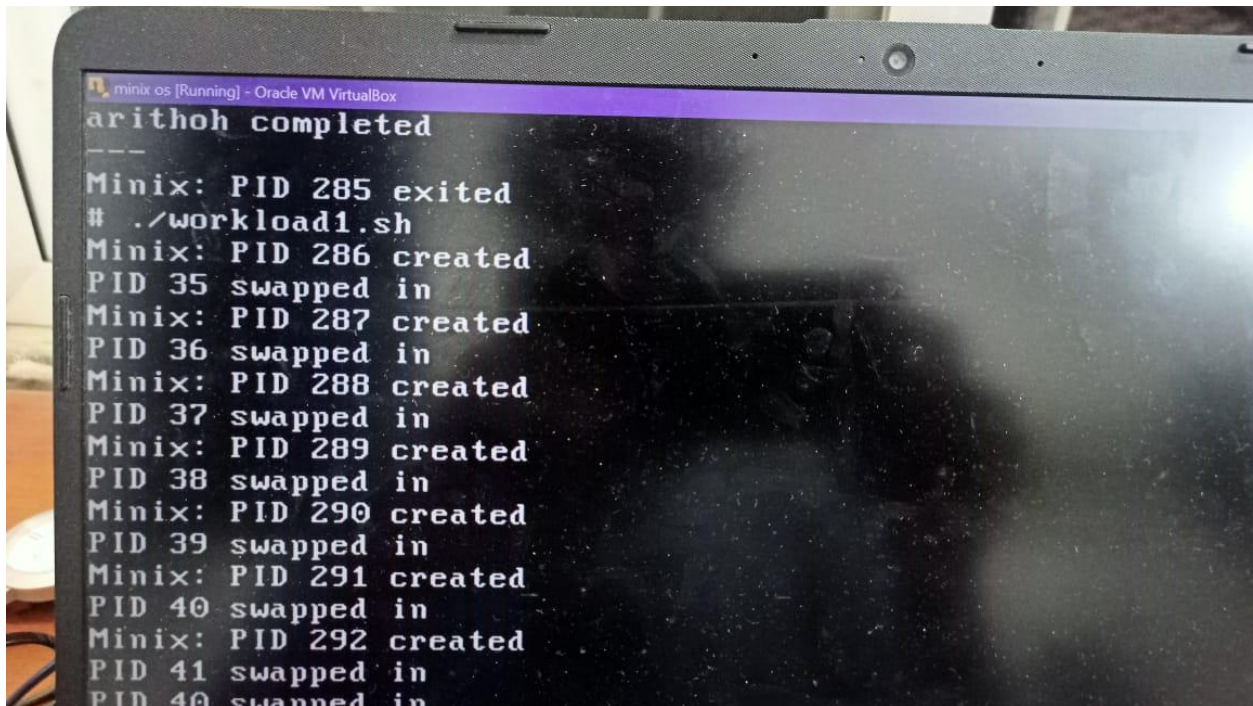
## Part-2

Using the UnixBench Benchmark Suite, some modifications in UnixBench/workload mix were done in order to study the behavior of the scheduler by seeing the sequence of PID prints when these workloads are run. In the following subsections, 4 different combinations of workload mix to understand how workloads are run.

### 2.1 workload mix1.sh

In this shell script, only script arithoh.sh is used as follows:

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



minix os [Running] - Oracle VM VirtualBox

```
PID 38 swapped in
Minix: PID 290 created
PID 39 swapped in
Minix: PID 291 created
PID 40 swapped in
Minix: PID 292 created
PID 41 swapped in
PID 40 swapped in
PID 41 swapped in
PID 41 swapped in
PID 40 swapped in
PID 41 swapped in
PID 40 swapped in
PID 40 swapped in
PID 41 swapped in
PID 40 swapped in
PID 41 swapped in
PID 40 swapped in
PID 41 swapped in
```

minix os [Running] - Oracle VM VirtualBox

PID	41	swapped	in
PID	40	swapped	in
PID	40	swapped	in
PID	41	swapped	in
PID	40	swapped	in
PID	41	swapped	in
PID	40	swapped	in
PID	41	swapped	in
PID	40	swapped	in
PID	41	swapped	in
PID	40	swapped	in
PID	41	swapped	in
PID	40	swapped	in
PID	41	swapped	in
PID	41	swapped	in
PID	40	swapped	in
PID	40	swapped	in



```

PID 40 swapped in
PID 41 swapped in
PID 40 swapped in
PID 41 swapped in
PID 40 swapped in
PID 40 swapped in
PID 41 swapped in
Minix: PID 291 exited
      22.80 real      11.01 user      0.76 sys
Minix: PID 289 exited
arithoh completed
----
Minix: PID 287 exited
PID 41 swapped in
PID 41 swapped in
PID 41 swapped in
Minix: PID 292 exited
      23.33 real      10.96 user      0.55 sys
Minix: PID 290 exited
arithoh completed
----
Minix: PID 288 exited
Minix: PID 286 exited
#

```

Here, first arithoh.sh has PID of 40, second arithoh.sh has PID 41 . Clearly, if multiple instances are run of arithoh.sh which are CPU Intensive in nature as in this workload, it can be seen that the processes are getting scheduled alternatively. The figure will show that processes with PID 40,41 correspond to 2 instances of workload in arithoh.sh that are scheduled alternatively until they are completed.

## 2.2 workload mix2.sh

In this shell script, script fstime.sh is used as follows:

```

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

```

In Figures two same workloads fstime.sh with PID's 129,130 . All the 2 are I/O processes that are waiting for input. When a process receives input, then that PID process is scheduled and finished prior. In the similar fashion, in the order of received inputs, processes are scheduled and then finished.

```
Minix: PID 369 exited
Minix: PID 367 exited
# ./workload2.sh
Minix: PID 374 created
PID 124 swapped in
Minix: PID 375 created
PID 125 swapped in
Minix: PID 376 created
PID 126 swapped in
Minix: PID 377 created
PID 127 swapped in
Minix: PID 378 created
PID 128 swapped in
Minix: PID 379 created
PID 129 swapped in
Minix: PID 380 created
PID 130 swapped in
Write done: 1008000 in 1.8500, score 136216
Write done: 1008000 in 1.8500, score 136216
COUNT:136216:0:KBps
COUNT:136216:0:KBps
TIME:1.9
TIME:1.9
```

```
Minix: PID 375 created
PID 125 swapped in
Minix: PID 376 created
PID 126 swapped in
Minix: PID 377 created
PID 127 swapped in
Minix: PID 378 created
PID 128 swapped in
Minix: PID 379 created
PID 129 swapped in
Minix: PID 380 created
PID 130 swapped in
Write done: 1008000 in 1.8500, score 136216
Write done: 1008000 in 1.8500, score 136216
COUNT:136216:0:KBps
COUNT:136216:0:KBps
TIME:1.9
TIME:1.9
Read done: 1000004 in 1.8000, score 138889
Read done: 1000004 in 1.8000, score 138889
COUNT:138889:0:KBps
COUNT:138889:0:KBps
TIME:1.8
TIME:1.8
```

```
TIME:1.8
TIME:1.8
PID 130 swapped in
PID 129 swapped in
Copy done: 1000004 in 3.2167, score 77720
COUNT:77720:0:KBps
TIME:3.2
Minix: PID 379 exited
      17.88 real      0.30 user      3.61 sys
Minix: PID 377 exited
fstime completed
---
Minix: PID 375 exited
PID 130 swapped in
Copy done: 1000004 in 3.6667, score 68182
COUNT:68182:0:KBps
TIME:3.7
Minix: PID 380 exited
      18.33 real      0.41 user      2.98 sys
Minix: PID 378 exited
fstime completed
---
Minix: PID 376 exited
Minix: PID 374 exited
```

## 2.3 workload mix3.sh

In this shell script, 2 scripts arithoh.sh, syscall.sh are used as follows:

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

Here, arithoh.sh has PID 149 and syscall.sh has PID 151. By literature, it is known that arithoh.sh and syscall.sh are different kinds of CPU Intensive tasks. Figure shows that during the execution at start, both are alternatively scheduled depending upon their intensiveness and then when syscall.sh with PID 151, which is relatively less CPU Intensive as compared to arithoh.sh is completed prior. Towards the end, arithoh.sh is scheduled until finished completely.



```
minix os [Running] - Oracle VM VirtualBox
13.76 real      8.56 user      0.16 sys
Minix: PID 388 exited
arithoh completed
---
Minix: PID 386 exited
Minix: PID 385 exited
# ./workload3.sh
Minix: PID 392 created
PID 143 swapped in
Minix: PID 393 created
PID 145 swapped in
Minix: PID 394 created
PID 146 swapped in
Minix: PID 395 created
PID 147 swapped in
Minix: PID 396 created
PID 148 swapped in
Minix: PID 397 created
PID 149 swapped in
Minix: PID 398 created
PID 151 swapped in
PID 149 swapped in
PID 151 swapped in
```

```
PID 148 swapped in
Minix: PID 397 created
PID 149 swapped in
Minix: PID 398 created
PID 151 swapped in
PID 149 swapped in
PID 151 swapped in
PID 149 swapped in
PID 151 swapped in
PID 149 swapped in
PID 151 swapped in
PID 149 swapped in
PID 151 swapped in
PID 149 swapped in
Minix: PID 398 exited
6.10 real      1.88 user      3.11 sys
Minix: PID 396 exited
syscall completed
---
Minix: PID 394 exited
PID 149 swapped in
PID 149 swapped in
```

[illegible]

```

PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
PID 149 swapped in
Minix: PID 397 exited
      13.70 real      8.61 user      0.08 sys
Minix: PID 395 exited
arithoh completed
---
Minix: PID 393 exited
Minix: PID 392 exited
#

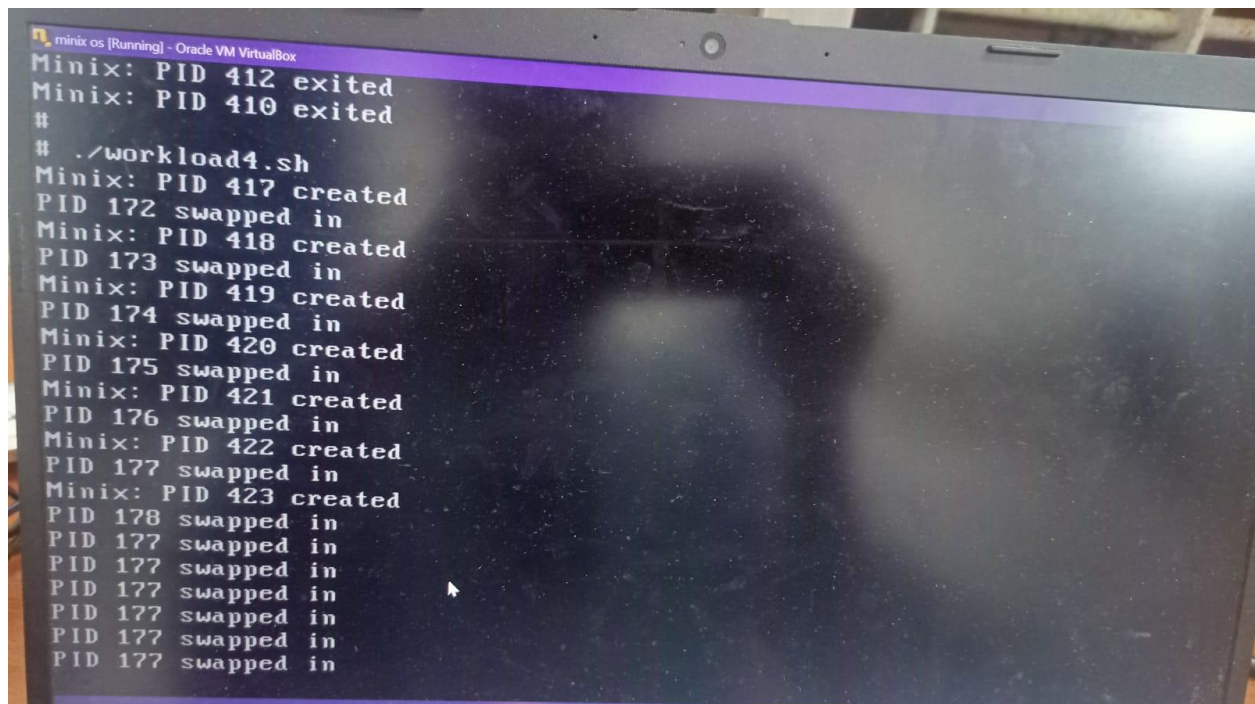
```



## 2.4 workload mix4.sh

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

In workload mix4.sh, an instance of file arithoh.sh and an instance of file fstime.sh are run. It is clear that the instructions in arithoh.sh are computationally intensive, while those in fstime.sh are I/O bound. Clearly in Figure it is seen that only the “PID %d swapped in” statements corresponding to arithoh.sh get printed as fstime.sh waits for its I/O operations to be completed. Here, arithoh.sh has PID 177 and fstime.sh has PID 178. So, PID 177 process is scheduled and PID process 178 is waiting for an input till then arithoh.sh is scheduled and is using the processor for its CPU intensive tasks. Then the process with PID 177 will complete its Execution(Arithmetic) and after the process with PID 178 will get its input and scheduled and then complete its execution.



[illegible]

```

minix os [Running] - Oracle VM VirtualBox
PID 177 swapped in
PID 177 swapped in
Read done: 1000004 in 0.8500, score 294118
COUNT:294118:0:KBps
TIME:0.8
PID 177 swapped in
PID 177 swapped in
Minix: PID 422 exited
      10.28 real      8.41 user      0.08 sys
Minix: PID 420 exited
arithoh completed
-----
Minix: PID 418 exited
PID 178 swapped in
Copy done: 1000004 in 1.8500, score 135135
COUNT:135135:0:KBps
TIME:1.9
Minix: PID 423 exited
      14.83 real      0.43 user      3.18 sys
Minix: PID 421 exited
fstime completed
-----
Minix: PID 419 exited
Minix: PID 417 exited
#

```