CS 314: Operating Systems Lab - 3

Chalumuri Sai Yeswanth (200010006), Pochimireddy Nikhileswar Reddy (200010040)

January 22, 2023

1 PART 1

In this part we should Modify the Minix3 source code such that the string "PID $< pid >$ swapped in" is printed, whenever a user-level process is brought in by the scheduler.
We have gone through the code in schedule.c file in minix/servers/sched directory and have added the following lines in schedule.c file after understanding the working of scheduler.

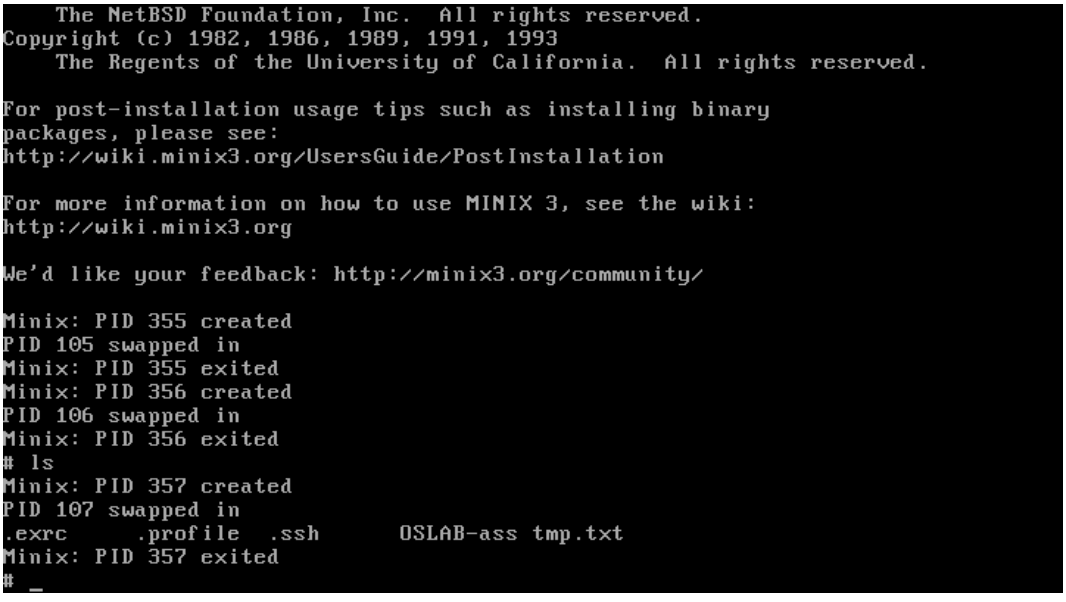
```
if (rmp->priority >= USER_Q)
{
    printf("200010040_200010006 PID %d swapped in ) \n", _ENDPOINT_P(rmp->endpoint));
}
```

We have also made a shell script runme.sh which have the following commands

```
cp schedule.c /usr/src/minix/servers/sched/schedule.c
cd --
cd /usr/src
make build MKUPDATE=yes
```

which on running copies the modified schedule.c file to /usr/src/minix/servers/sched/schedule.c and goes to /usr/src to build the system.

After building and rebooting the changes appeared are as follows along with our roll numbers.



2 PART 2

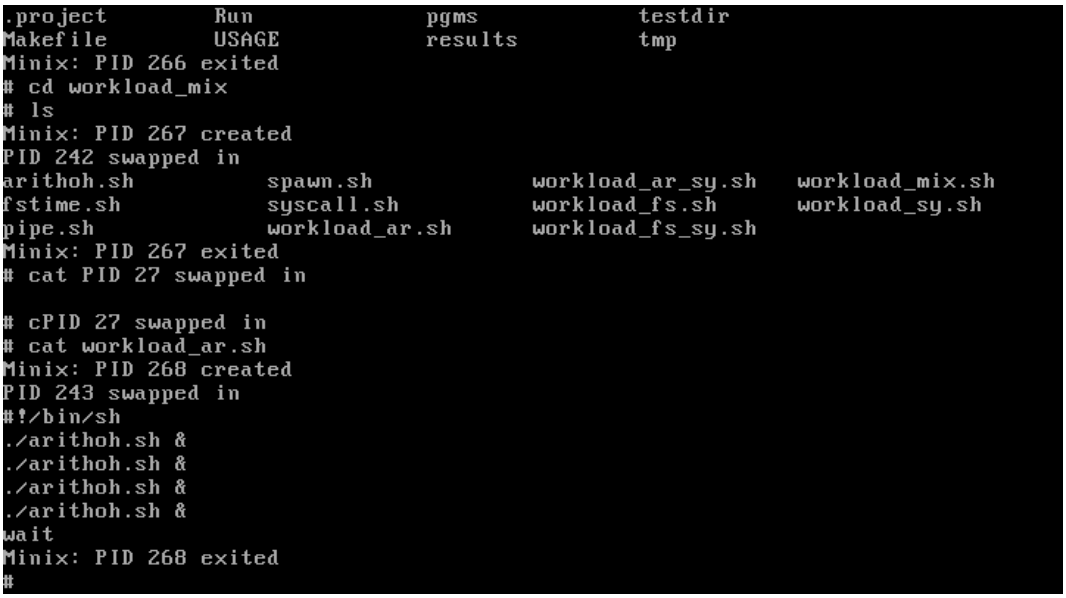
We have downloaded the source code and extracted it in the host machine. Now we have added five workloads of our own namely

- workload_ar.sh
- workload_fs.sh
- workload_ar_sy.sh
- workload_fs_sy.sh
- workload_sy.sh

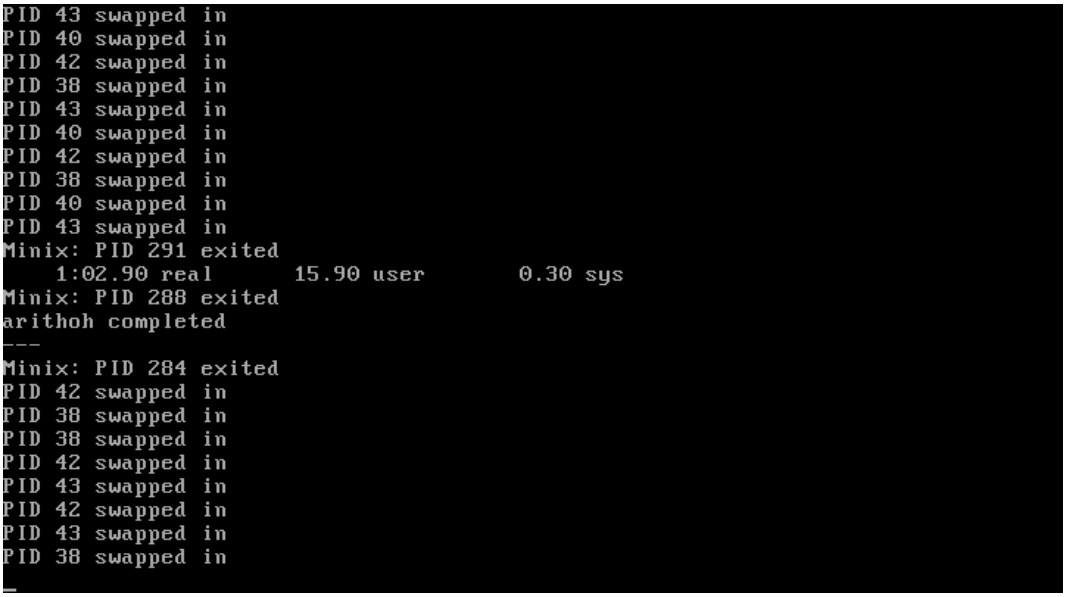
Now we have copied this folder into minix3 VM We have studied the nature of the benchmarks in the UnixBench suite by analyzing the behavior of the scheduler by seeing the sequence of "PID" prints when the workloads are run.

2.a workload_ar.sh

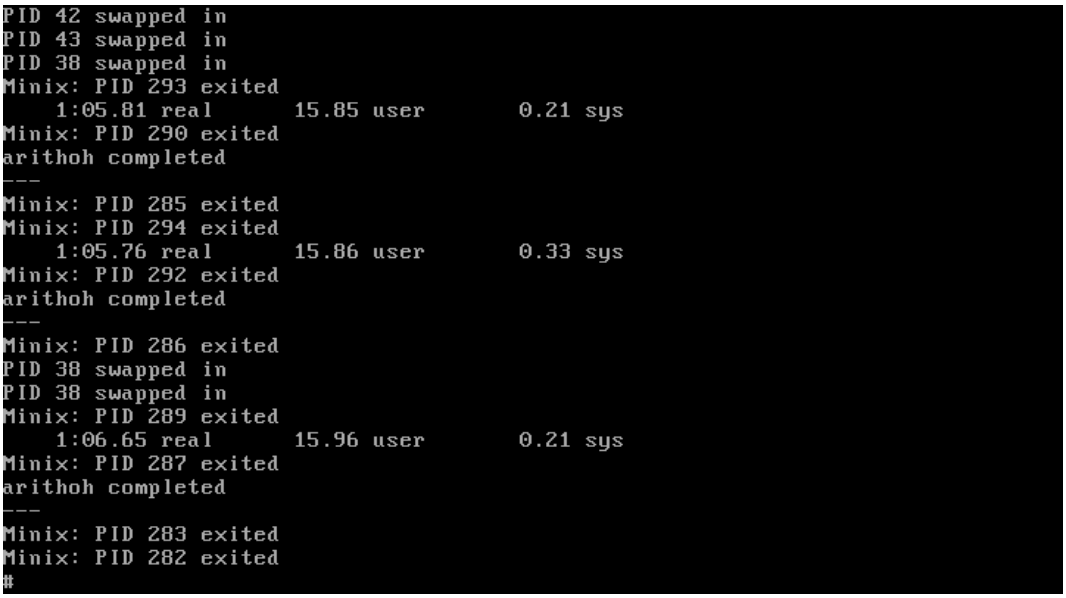
```
#!/bin/sh
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
wait
```




PID 42 swapped in
PID 38 swapped in
PID 43 swapped in
PID 40 swapped in
PID 42 swapped in
PID 38 swapped in
PID 43 swapped in
PID 40 swapped in
PID 42 swapped in
PID 38 swapped in
PID 43 swapped in
PID 40 swapped in
PID 42 swapped in
PID 38 swapped in
PID 43 swapped in
PID 40 swapped in
PID 42 swapped in
PID 38 swapped in
PID 43 swapped in
PID 40 swapped in
PID 38 swapped in
PID 40 swapped in
PID 42 swapped in
PID 43 swapped in


PID 43 swapped in
PID 40 swapped in
PID 42 swapped in
PID 38 swapped in
PID 43 swapped in
PID 40 swapped in
PID 42 swapped in
PID 38 swapped in
PID 40 swapped in
PID 43 swapped in
Minix: PID 291 exited
1:02.90 real 15.90 user 0.30 sys
Minix: PID 288 exited
arithoh completed

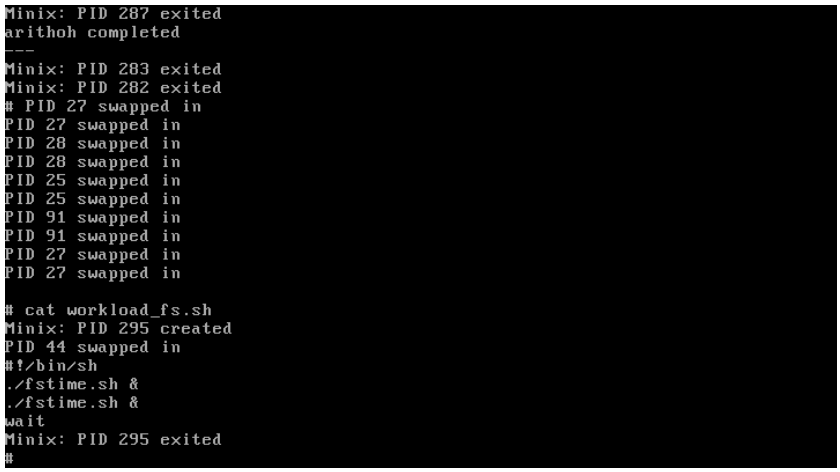
Minix: PID 284 exited
PID 42 swapped in
PID 38 swapped in
PID 38 swapped in
PID 42 swapped in
PID 43 swapped in
PID 42 swapped in
PID 43 swapped in
PID 38 swapped in



In this workload first arithoh.sh has PID of 38, second arithoh.sh has PID of 40, third arithoh.sh has PID of 42 and last arithoh.sh has PID of 43. As we know that arithoh.sh is CPU Intensive in nature and here multiple arithoh.sh are being runned.Each process run until the time-slice of that process is completed, then the scheduler will swap the current process to next process alternatively.

2.b workload_fs.sh

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



```
TIME¦1.9
PID 59 swapped in
PID 58 swapped in
PID 58 swapped in
PID 59 swapped in
Copy done: 1000004 in 3.5333, score 70755
COUNT¦70755¦0¦KBps
TIME¦3.5
Minix: PID 309 exited
      18.31 real       0.36 user       3.68 sys
Minix: PID 307 exited
fstime completed
---
Minix: PID 305 exited
Copy done: 1000004 in 3.8333, score 65217
COUNT¦65217¦0¦KBps
TIME¦3.8
Minix: PID 310 exited
      18.61 real       0.21 user       3.35 sys
Minix: PID 308 exited
fstime completed
---
Minix: PID 306 exited
Minix: PID 304 exited
#
```

In this workload the PID's of two fstime.sh are 58 and 59. The two processes are waiting for input and as soon the input is received, the PID processes are scheduled and finished. The process which received input first is scheduled and finished.

2.c workload_ar_sy.sh

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

```
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
PID 80 swapped in
Minix: PID 331 exited
      20.88 real      15.58 user       0.26 sys
Minix: PID 329 exited
arithoh completed
---
Minix: PID 327 exited
Minix: PID 326 exited
# cat workload_ar_sy.sh
Minix: PID 333 created
PID 82 swapped in
#!/bin/sh
./arithoh.sh &
./syscall.sh &
wait
Minix: PID 333 exited
# _
```

```
PID 89 swapped in
PID 88 swapped in
PID 89 swapped in
PID 88 swapped in
PID 89 swapped in
PID 88 swapped in
PID 89 swapped in
PID 88 swapped in
PID 88 swapped in
PID 89 swapped in
PID 89 swapped in
PID 88 swapped in
PID 89 swapped in
PID 88 swapped in
Minix: PID 340 exited
       6.51 real       1.65 user       3.55 sys
Minix: PID 338 exited
syscall completed
---
Minix: PID 336 exited
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
_
```

```
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
PID 88 swapped in
Minix: PID 339 exited
      21.46 real      16.01 user       0.25 sys
Minix: PID 337 exited
arithoh completed
---
Minix: PID 335 exited
Minix: PID 334 exited
# _
```

In this workload the PID of arithoh.sh is 88 and PID of syscall.sh is 89. Both arithoh.sh and syscall.sh are CPU intensive.Syscall.sh is completed prior to arithoh.sh as it is light process compared to arithoh.sh. After completion of syscall.sh , arithoh.sh is scheduled until it is completed.

2.d workload_fs_sy.sh

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

```
Minix: PID 309 exited
      18.31 real       0.36 user       3.68 sys
Minix: PID 307 exited
fstime completed
---
Minix: PID 305 exited
Copy done: 1000004 in 3.8333, score 65217
COUNT¦65217¦0¦KBps
TIME¦3.8
Minix: PID 310 exited
      18.61 real       0.21 user       3.35 sys
Minix: PID 308 exited
fstime completed
---
Minix: PID 306 exited
Minix: PID 304 exited
# cat workload_fs_sy.sh
Minix: PID 311 created
PID 60 swapped in
#!/bin/sh
./fstime.sh &
./syscall.sh &
wait
Minix: PID 311 exited
# _
```

```
PID 93 swapped in
Minix: PID 344 created
PID 94 swapped in
Minix: PID 345 created
PID 95 swapped in
Minix: PID 346 created
PID 96 swapped in
Minix: PID 347 created
PID 97 swapped in
PID 97 swapped in
PID 97 swapped in
PID 97 swapped in
PID 97 swapped in
PID 97 swapped in
Write done: 1008000 in 1.0000, score 252000
COUNT¦252000¦0¦KBps
TIME¦1.0
PID 97 swapped in
Minix: PID 347 exited
       6.30 real       1.81 user       3.46 sys
Minix: PID 345 exited
syscall completed
---
Minix: PID 343 exited
```

```
Write done: 1008000 in 1.0000, score 252000
COUNT|252000|0|KBps
TIME|1.0
PID 97 swapped in
Minix: PID 347 exited
       6.30 real       1.81 user       3.46 sys
Minix: PID 345 exited
syscall completed
---
Minix: PID 343 exited
Read done: 1000004 in 0.9667, score 258621
COUNT|258621|0|KBps
TIME|1.0
PID 96 swapped in
Copy done: 1000004 in 1.9167, score 130435
COUNT|130435|0|KBps
TIME|1.9
Minix: PID 346 exited
      14.88 real       0.18 user       3.70 sys
Minix: PID 344 exited
fstime completed
---
Minix: PID 342 exited
Minix: PID 341 exited
#
```

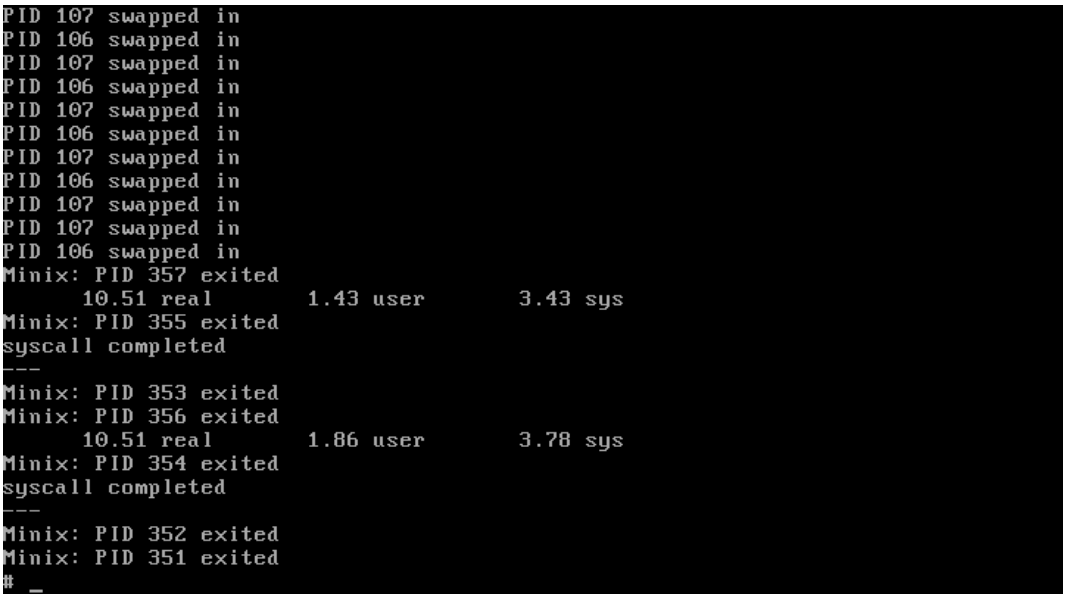
In this workload the PID of fstime.sh is 96 and syscall.sh is 97. The fstime.sh will be waiting for the input for it to be scheduled and in the mean time syscall.sh is scheduled until it's completion.After the input is received fstime.sh will be scheduled and completed.

2.e workload_sy.sh

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

```
PID 98 swapped in
arithoh.sh          spawn.sh            workload_ar_sy.sh   workload_mix.sh
fstime.sh           syscall.sh          workload_fs.sh      workload_sy.sh
pipe.sh             workload_ar.sh      workload_fs_sy.sh
Minix: PID 348 exited
# cat workload_sy.sh
Minix: PID 349 created
PID 99 swapped in
#!/bin/sh
./syscall.sh &
./syscall.sh &
wait
Minix: PID 349 exited
# PID 27 swapped in
PID 27 swapped in

# cat workload_sy.sh
Minix: PID 350 created
PID 100 swapped in
#!/bin/sh
./syscall.sh &
./syscall.sh &
wait
Minix: PID 350 exited
# _
```



In this workload the PID of first syscall.sh is 106 and second syscall.sh is 107. Here the both processes are scheduled alternatively until their completion.