

# CS 314 Lab 3

MV Karthik - 200010030

Josyula V N Taraka Abhishek - 200010021

Jan 2023

## Contents

<b>1</b>	<b>Part - I</b>	<b>1</b>
<b>2</b>	<b>Part-II</b>	<b>2</b>
2.1	Custom Mix - 1 . . . . .	3
2.2	Custom Mix - 2 . . . . .	4
2.3	Custom Mix - 3 . . . . .	5
2.4	Conclusions . . . . .	6
2.5	Runnig the mixes . . . . .	7

## List of Figures

1	Process swapped in statements are printed successfully. . . . .	2
2	Process scheduling sequence for mix1 or log messages . . . . .	4
3	Process scheduling sequence for mix2 or log messages . . . . .	5
4	Process scheduling sequence for mix3 or log messages . . . . .	6
5	Run mix1 . . . . .	7
6	Run mix2 . . . . .	7
7	Run mix3 . . . . .	8

## 1 Part - I

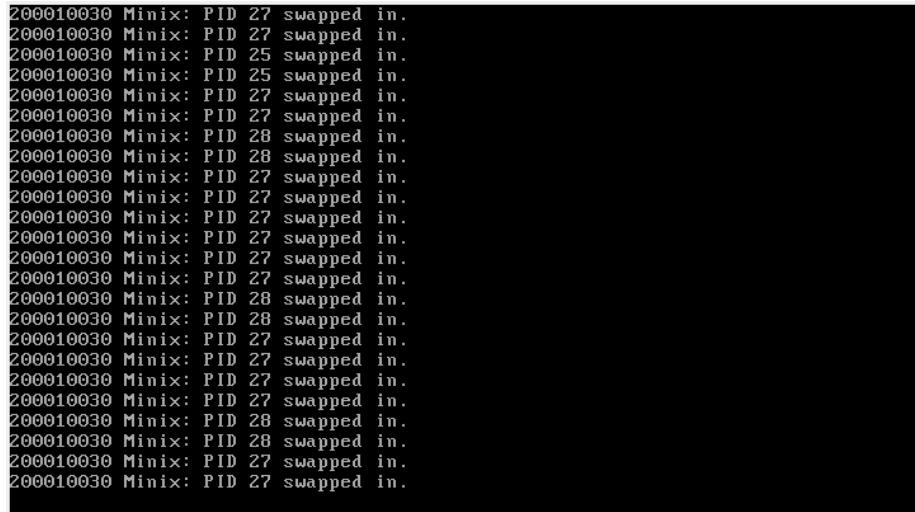
We are asked to log a statement "Minix: process PID <pid> swapped in" for each user process that is brought in by the scheduler. We observed that the **sched** server is calling kernel call to schedule a process in the function **schedule\_process.c**. Which is used by **do\_start\_schedule**, **balance\_queues**, **do\_noquantum** and **do\_nice** entry point functions. Hence we included the below if statement in the **schedule\_process** function.

```

if (err == OK && _ENDPOINT_P(rmp->endpoint)>=0)
    printf("200010030 Minix: PID %d swapped in.\n",
        _ENDPOINT_P(rmp->endpoint));
}

```

The function `_ENDPOINT_P` returns the process slot number of the process from the given endpoint, which is similar to the notion of PID. A non-negative process slot number indicates that it is a user process, else it will be a kernel process. (Described in `endpoint.h` file)



```

200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 25 swapped in.
200010030 Minix: PID 25 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 28 swapped in.
200010030 Minix: PID 28 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 28 swapped in.
200010030 Minix: PID 28 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 28 swapped in.
200010030 Minix: PID 28 swapped in.
200010030 Minix: PID 27 swapped in.
200010030 Minix: PID 27 swapped in.

```

Figure 1: Process swapped in statements are printed successfully.

## 2 Part-II

We have run the benchmarks on the minix system by various custom workload mixes to find the nature of the unix benchmarks(I/O bound or CPU bound).

1. From the analysis of the code, we see that `arithoh.sh` is CPU intensive
2. `fstime` is IO intensive
3. `syscall` has lots of blocking code (syscalls).

The following code is used to obtain log messages

```

#!/bin/sh

cat "" > /var/log/messages
bash $1
cat /var/log/messages > $1.txt

```

## 2.1 Custom Mix - 1

Custom mix 1 contains 4 concurrent runs of arithoh.sh and the messages are printed into log.txt

```
#!/bin/sh

./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
wait
```

We can see the PID's of four process swapping frequently from log messages

```

Lab3 > byte-unixbench-mod > UnixBench > workload_mix > mix1.sh.txt
1 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 228 swapped in
2 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 229 swapped in
3 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 230 swapped in
4 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 231 swapped in
5 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 232 swapped in
6 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 233 swapped in
7 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 234 swapped in
8 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 235 swapped in
9 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 236 swapped in
10 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 237 swapped in
11 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 238 swapped in
12 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 239 swapped in
13 Jan 23 03:57:16 10 kernel: 200010021 Minix: PID 240 swapped in
14 Jan 23 03:57:17 10 kernel: 200010021 Minix: PID 237 swapped in
15 Jan 23 03:57:17 10 kernel: 200010021 Minix: PID 238 swapped in
16 Jan 23 03:57:17 10 kernel: 200010021 Minix: PID 239 swapped in
17 Jan 23 03:57:17 10 kernel: 200010021 Minix: PID 240 swapped in
18 Jan 23 03:57:18 10 kernel: 200010021 Minix: PID 238 swapped in
19 Jan 23 03:57:18 10 kernel: 200010021 Minix: PID 239 swapped in
20 Jan 23 03:57:18 10 kernel: 200010021 Minix: PID 237 swapped in
21 Jan 23 03:57:18 10 kernel: 200010021 Minix: PID 240 swapped in
22 Jan 23 03:57:18 10 kernel: 200010021 Minix: PID 238 swapped in
23 Jan 23 03:57:19 10 kernel: 200010021 Minix: PID 239 swapped in
24 Jan 23 03:57:19 10 kernel: 200010021 Minix: PID 240 swapped in
25 Jan 23 03:57:19 10 kernel: 200010021 Minix: PID 237 swapped in
26 Jan 23 03:57:19 10 kernel: 200010021 Minix: PID 238 swapped in
27 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 237 swapped in
28 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 238 swapped in
29 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 239 swapped in
30 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 240 swapped in
31 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 239 swapped in
32 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 240 swapped in
33 Jan 23 03:57:20 10 kernel: 200010021 Minix: PID 237 swapped in
34 Jan 23 03:57:21 10 kernel: 200010021 Minix: PID 239 swapped in
35 Jan 23 03:57:21 10 kernel: 200010021 Minix: PID 240 swapped in
36 Jan 23 03:57:21 10 kernel: 200010021 Minix: PID 238 swapped in
37 Jan 23 03:57:21 10 kernel: 200010021 Minix: PID 237 swapped in
38 Jan 23 03:57:22 10 kernel: 200010021 Minix: PID 238 swapped in
39 Jan 23 03:57:22 10 kernel: 200010021 Minix: PID 239 swapped in
40 Jan 23 03:57:22 10 kernel: 200010021 Minix: PID 240 swapped in
41 Jan 23 03:57:22 10 kernel: 200010021 Minix: PID 237 swapped in
42 Jan 23 03:57:23 10 kernel: 200010021 Minix: PID 238 swapped in
43 Jan 23 03:57:23 10 kernel: 200010021 Minix: PID 239 swapped in
44 Jan 23 03:57:23 10 kernel: 200010021 Minix: PID 240 swapped in
45 Jan 23 03:57:23 10 kernel: 200010021 Minix: PID 237 swapped in
46 Jan 23 03:57:23 10 kernel: 200010021 Minix: PID 238 swapped in

```

Figure 2: Process scheduling sequence for mix1 or log messages

## 2.2 Custom Mix - 2

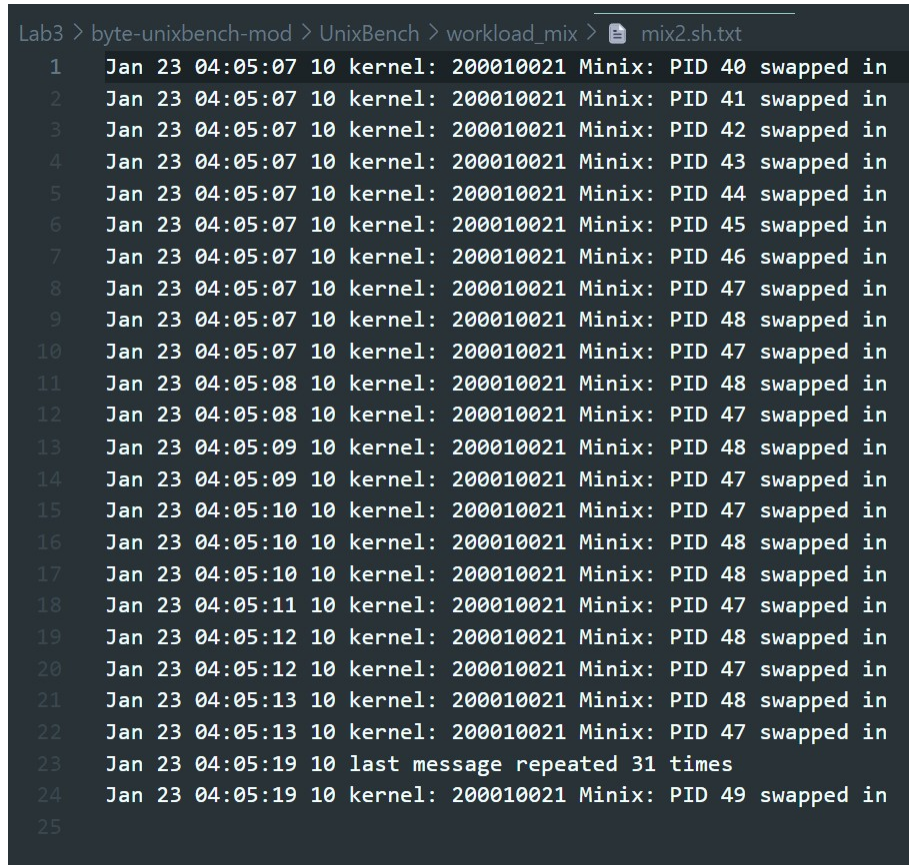
Custom mix 2 contains two benchmarks `arithoh.sh` and `syscall.sh` running in parallel.

From log message we see that PID 47 is swapped in 31 times so it is `arithoh.sh` and 48 is `syscall.sh`

```

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

```



```

Lab3 > byte-unixbench-mod > UnixBench > workload_mix > mix2.sh.txt
1 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 40 swapped in
2 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 41 swapped in
3 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 42 swapped in
4 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 43 swapped in
5 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 44 swapped in
6 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 45 swapped in
7 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 46 swapped in
8 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 47 swapped in
9 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 48 swapped in
10 Jan 23 04:05:07 10 kernel: 200010021 Minix: PID 47 swapped in
11 Jan 23 04:05:08 10 kernel: 200010021 Minix: PID 48 swapped in
12 Jan 23 04:05:08 10 kernel: 200010021 Minix: PID 47 swapped in
13 Jan 23 04:05:09 10 kernel: 200010021 Minix: PID 48 swapped in
14 Jan 23 04:05:09 10 kernel: 200010021 Minix: PID 47 swapped in
15 Jan 23 04:05:10 10 kernel: 200010021 Minix: PID 47 swapped in
16 Jan 23 04:05:10 10 kernel: 200010021 Minix: PID 48 swapped in
17 Jan 23 04:05:10 10 kernel: 200010021 Minix: PID 48 swapped in
18 Jan 23 04:05:11 10 kernel: 200010021 Minix: PID 47 swapped in
19 Jan 23 04:05:12 10 kernel: 200010021 Minix: PID 48 swapped in
20 Jan 23 04:05:12 10 kernel: 200010021 Minix: PID 47 swapped in
21 Jan 23 04:05:13 10 kernel: 200010021 Minix: PID 48 swapped in
22 Jan 23 04:05:13 10 kernel: 200010021 Minix: PID 47 swapped in
23 Jan 23 04:05:19 10 last message repeated 31 times
24 Jan 23 04:05:19 10 kernel: 200010021 Minix: PID 49 swapped in
25

```

Figure 3: Process scheduling sequence for mix2 or log messages

## 2.3 Custom Mix - 3

This workload mix contains benchmarks `arithoh.sh` and `fstime.sh` running in parallel.

From the log message, we see that PID 57 is swapped in 35 times, so it is `arithoh.sh` and 58 is `syscall.sh`.

```

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

```

```
./fstime.sh &  
wait
```

```
Lab3 > byte-unixbench-mod > UnixBench > workload_mix > mix3.sh.txt  
1 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 52 swapped in  
2 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 53 swapped in  
3 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 54 swapped in  
4 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 55 swapped in  
5 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 56 swapped in  
6 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 57 swapped in  
7 Jan 23 04:05:23 10 kernel: 200010021 Minix: PID 58 swapped in  
8 Jan 23 04:05:24 10 kernel: 200010021 Minix: PID 57 swapped in  
9 Jan 23 04:05:32 10 last message repeated 35 times  
10 Jan 23 04:05:33 10 kernel: 200010021 Minix: PID 23 swapped in  
11 Jan 23 04:05:35 10 kernel: 200010021 Minix: PID 23 swapped in  
12 Jan 23 04:05:38 10 kernel: 200010021 Minix: PID 58 swapped in  
13 Jan 23 04:05:38 10 kernel: 200010021 Minix: PID 59 swapped in  
14
```

Figure 4: Process scheduling sequence for mix3 or log messages

## 2.4 Conclusions

From mix1, we can see that the scheduler is giving the multiple processes of same `arithoh.sh` are getting scheduled alternately(Round robin way).

From mix2, We can see the scheduler is giving `arithoh.sh` less time slice, and it is scheduled more times than the more blocked process of `syscall.sh`

From mix3, We can see the scheduler is giving `arithoh.sh` less time slice, and it is scheduled more times than the more I/O blocked the process of `fstime.sh`

## 2.5 Runnig the mixes

```
# ./run.sh mix1.sh
200010021 Minix: PID 211 swapped in
200010021 Minix: PID 212 swapped in
cat: : No such file or directory
200010021 Minix: PID 213 swapped in
200010021 Minix: PID 214 swapped in
200010021 Minix: PID 215 swapped in
200010021 Minix: PID 216 swapped in
200010021 Minix: PID 217 swapped in
200010021 Minix: PID 218 swapped in
200010021 Minix: PID 219 swapped in
200010021 Minix: PID 220 swapped in
200010021 Minix: PID 221 swapped in
200010021 Minix: PID 222 swapped in
200010021 Minix: PID 223 swapped in
200010021 Minix: PID 220 swapped in
200010021 Minix: PID 224 swapped in
200010021 Minix: PID 222 swapped in
200010021 Minix: PID 225 swapped in
```

Figure 5: Run mix1

```
# ./run.sh mix2.sh
200010021 Minix: PID 228 swapped in
200010021 Minix: PID 229 swapped in
cat: : No such file or directory
200010021 Minix: PID 230 swapped in
200010021 Minix: PID 231 swapped in
200010021 Minix: PID 232 swapped in
200010021 Minix: PID 233 swapped in
200010021 Minix: PID 234 swapped in
200010021 Minix: PID 235 swapped in
200010021 Minix: PID 236 swapped in
200010021 Minix: PID 235 swapped in
200010021 Minix: PID 236 swapped in
200010021 Minix: PID 235 swapped in
200010021 Minix: PID 235 swapped in
200010021 Minix: PID 236 swapped in
```

Figure 6: Run mix2

```
# ./run.sh mix3.sh
200010021 Minix: PID 239 swapped in
200010021 Minix: PID 240 swapped in
cat: : No such file or directory
200010021 Minix: PID 241 swapped in
200010021 Minix: PID 242 swapped in
200010021 Minix: PID 243 swapped in
200010021 Minix: PID 244 swapped in
200010021 Minix: PID 245 swapped in
200010021 Minix: PID 246 swapped in
200010021 Minix: PID 247 swapped in
200010021 Minix: PID 246 swapped in
200010021 Minix: PID 246 swapped in
200010021 Minix: PID 246 swapped in
200010021 Minix: PID 246 swapped in
200010021 Minix: PID 246 swapped in
200010021 Minix: PID 246 swapped in
```

Figure 7: Run mix3