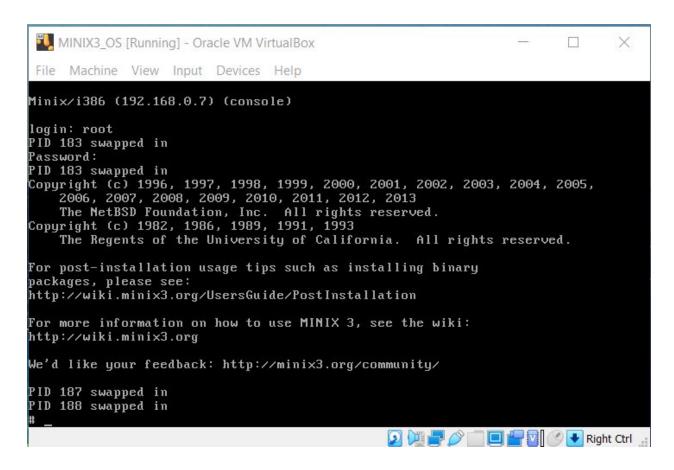
OS Lab 3

Name: Hemanth Reddy

Roll: 180010023

Part-1

The code modified in the file:- minix/servers/sched/schedule.c in the function schedule_process() for printing pid swapped whenever a user-level process is taken by the scheduler



```
MINIX3_OS [Running] - Oracle VM VirtualBox
 File Machine View Input Devices Help
 pwd
 cd /home
# ls
PID 207 swapped in
UnixBench hemanth
# cd UnixBench/
# ls
PID 208 swapped in
.cproject
               README
                               WRITING_TESTS
                                               src
                                                               workload_mix
.project
               Run
                                               testdir
                               pgms
               USAGE
Makefile
                               results
                                               tmp
# cd workload_mix/
# ls
PID 209 swapped in
               pipe.sh
arithoh.sh
                               syscall.sh
fstime.sh
                               workload_mix.sh
               spawn.sh
                                             Right Ctrl
```

Part-2

Observations:

1.Arithoh

Running an arithoh.sh instance only once, the time taken by real and the user is exactly the same. Also the sys time taken is 0.

```
PID 232 swapped in
PID 232
        swapped
                 in
PID 232
        swapped
PID 232 swapped in
      15.68 real
                       15.68 user
                                          0.00 sys
arithoh completed
```

2. Arithoh vs Arithoh

Running two instances of arithoh.sh, we see that the pid is not going sequentially rather making their movers parallely. Also we can observe though the process start at the same time but the first process ends earlier. So we can claim that it was given the most priority.

```
PID 225 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
PID 225 swapped
PID 224 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped
PID 225 swapped in
PID 225 swapped in
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
```

```
PID 224 swapped in
PID 225 swapped in
PID 224 swapped in
PID 225 swapped in
PID 225 swapped in
PID 224 swapped in
PID 225 swapped in
PID 225 swapped
PID 224 swapped
                in
PID 225 swapped in
PID 225 swapped in
PID 224 swapped in
     30.13 real
                      15.76 user
                                       0.00 sys
arithoh completed
PID 224 swapped in
     31.58 real
                      15.81 user
                                       0.00 sys
arithoh completed
```

3. Arithoh vs syscall

Running arithoh.sh and syscall.sh (both of them are cpu bound benchmarks). We can observe from screenshots that the users time taken is way less than that of real due its computational(cpu) intensive tasks. We can also see round robin way of execution before completion of syscall (240,241,240,...).

```
PID 240 swapped in
PID 241 swapped in
PID 241 swapped in
PID 240 swapped in
PID 241 swapped in
PID 240 swapped in
       6.03 real
                                       3.23 sys
                       1.71 user
syscall completed
PID 240 swapped in
```

```
PID 240 swapped in
PID 240 swapped in
PID 240 swapped in
PID 240 swapped
                i n
PID 240 swapped
                i n
PID 240 swapped
                i n
PID 240 swapped
PID 240 swapped
                i n
PID 240 swapped
                i n
PID 240 swapped
                i n
PID 240 swapped in
                       15.76 user
                                         0.00 sys
     20.71 real
arithoh completed
```

4. Arithoh vs fstime

Running arithoh.sh and fstime.sh, The later one(fstime) is a special file compared to others, that is it is an I/O bound Unix Benchmark.

We see most of arithoh.sh process running in the screenshot because the I/O process waits for its I/O and then complex processing(see 87 pid just before its completion). And this is the reason for user time to be very less compared to that of arithoh.sh

Here the fstime finishes earlier followed by consecutive scheduling of arithon until completion.

```
PID 86 swapped in
Write done: 1008000 in 0.9167, score 274909
COUNT:274909:0:KBps
TIME:0.9
PID 86 swapped in
```

```
PID 86 swapped in
Read done: 1000004 in 0.8500, score 294118
COUNT:294118:0:KBps
TIME:0.8
PID 86 swapped in
```

```
PID 86 swapped in
PID 86 swapped in
PID 86 swapped in
PID 86 swapped in
PID 86 swapped
PID 86 swapped in
PID 86 swapped in
PID 86 swapped in
PID 87 swapped in
Copy done: 1000004 in 1.8333, score 136364
COUNT:136364:0:KBps
TIME:1.8
      14.81 real
                        0.45 user
                                          3.15 sys
fstime completed
PID 86 swapped in
PID 86 swapped
PID 86 swapped in
```

```
PID 86 swapped in
PID 86 swapped
PID 86 swapped in
      19.28 real
                      15.68 user
                                        0.00 sys
arithoh completed
```

5. Arithoh vs pipe

Running arithoh.sh and pipe.sh(both of them are cpu bound benchmarks). A huge amount of time spent by Pipe is in sys mode. This is because the Inter-Process Communication protocols are highly based on the system. The user-mode time spent is very less. Here the pipe finishes earlier followed by consecutive scheduling of arithoh until completion.

```
PID 99 swapped in
PID 98 swapped in
PID 98 swapped in
PID 99 swapped in
7.61 real
                        0.70 user
                                        6.50 sys
pipe completed
PID 98 swapped in
PID 98 swapped in
PID 98 swapped in
PID 98 swapped
               in
PID 98 swapped in
PID 98 swapped
PID 98 swapped in
PID 98 swapped
               in
PID 98 swapped
               in
PID 98 swapped
PID 98 swapped
PID 98 swapped in
      22.90 real
                       15.70 user
                                        0.00 sys
arithoh completed
```

6.spawn vs arithoh

Running arithoh.sh and spawn.sh(both of them are cpu bound benchmarks). We can see there are many processes running down in the screenshot(descendants of spawn.sh). Here the time spent by the system is large compared to that of the user.

Here the pipe finishes earlier followed by consecutive scheduling of arithon until completion.

```
PID 224 swapped in
PID 225 swapped in
PID 226 swapped in
PID 227 swapped in
PID 228 swapped in
PID 229 swapped in
PID 230 swapped in
PID 231 swapped in
PID 232 swapped in
PID 233 swapped in
        5.76 real
                            0.28 user
                                               4.06 sys
spawn completed
PID 109 swapped in
```

```
PID 109 swapped in
      20.93 real
                       15.60 user
                                         0.00 sys
arithoh completed
```