Hemanth Surya S(180010018)

Operating Systems Laboratory

Assignment-3

REPORT

Part-1:

By modifying the schedule_process() function in minix/servers/sched/schedule.c file, we get the user-level process brought in by scheduler. This can be achieved by printing endpoint of the process being handled with highest priority.

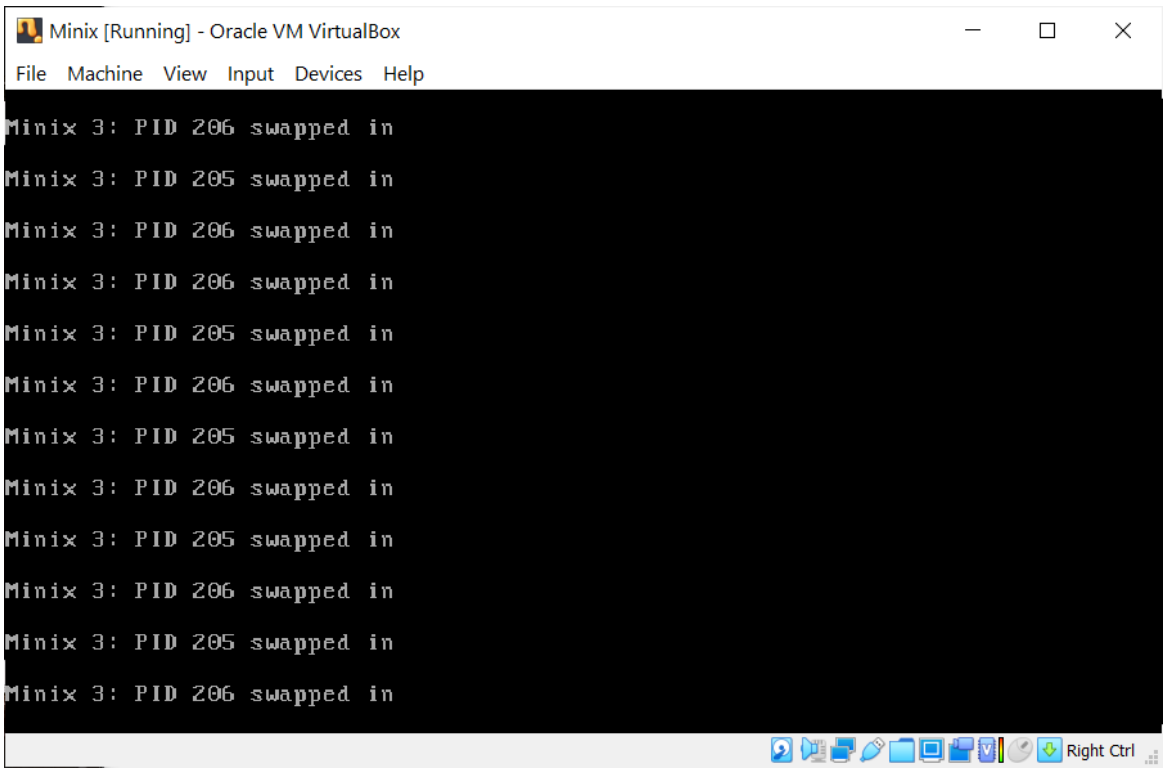
```
/*===========================================================================*
 *                schedule_process                                           *
 *===========================================================================*/
static int schedule_process(struct schedproc * rmp, unsigned flags)
{
    int err;
    int new_prio, new_quantum, new_cpu;

    pick_cpu(rmp);

    if(rmp->priority >= 7){
        printf("\nMinix 3: PID %d swapped in\n", _ENDPOINT_P(rmp->endpoint));
    }
```

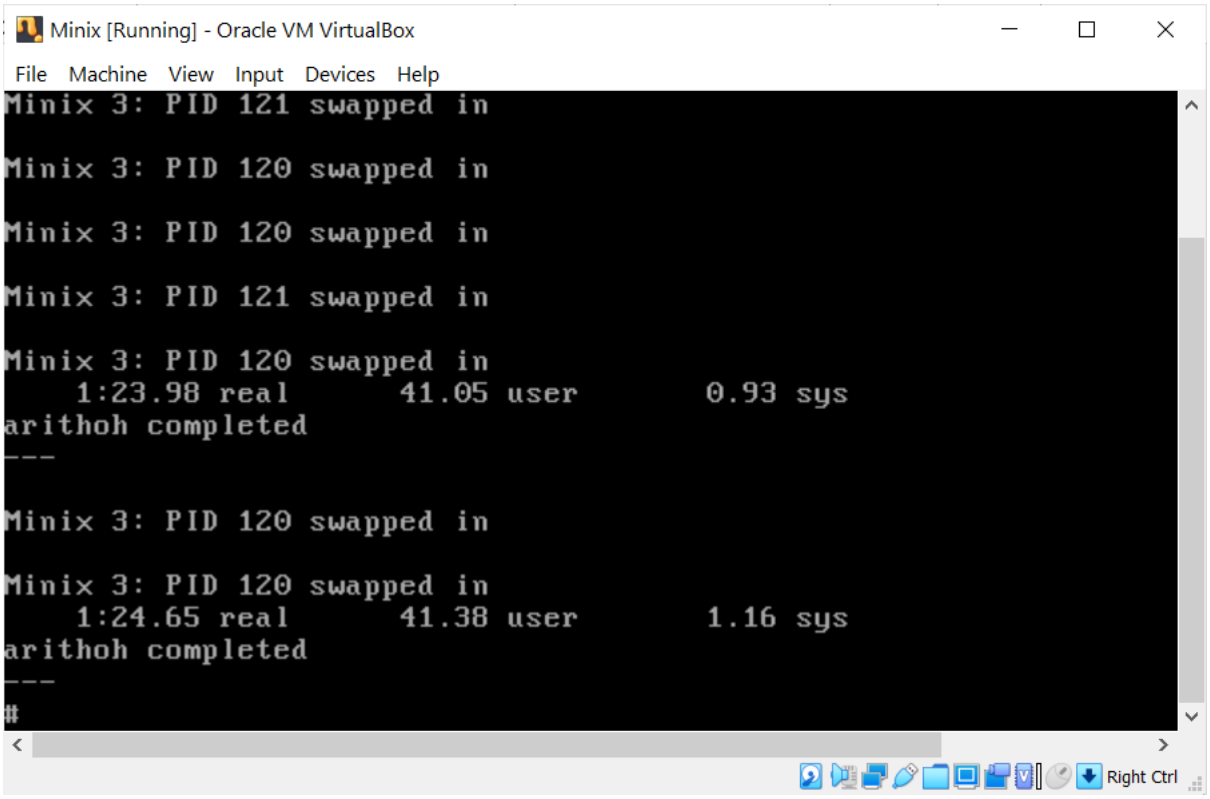
Part-2: UnixBench benchmark analysis

Observations:

The following tasks are carried in the Unixbench benchmark suite:

1. **Arithoh-CPU bound intensive task**
   To see how scheduler handles two CPU intensive tasks run simultaneously.
   Run two instances of arithoh.sh which replicates CPU intensive task and doesn't wait for any I/O operation.



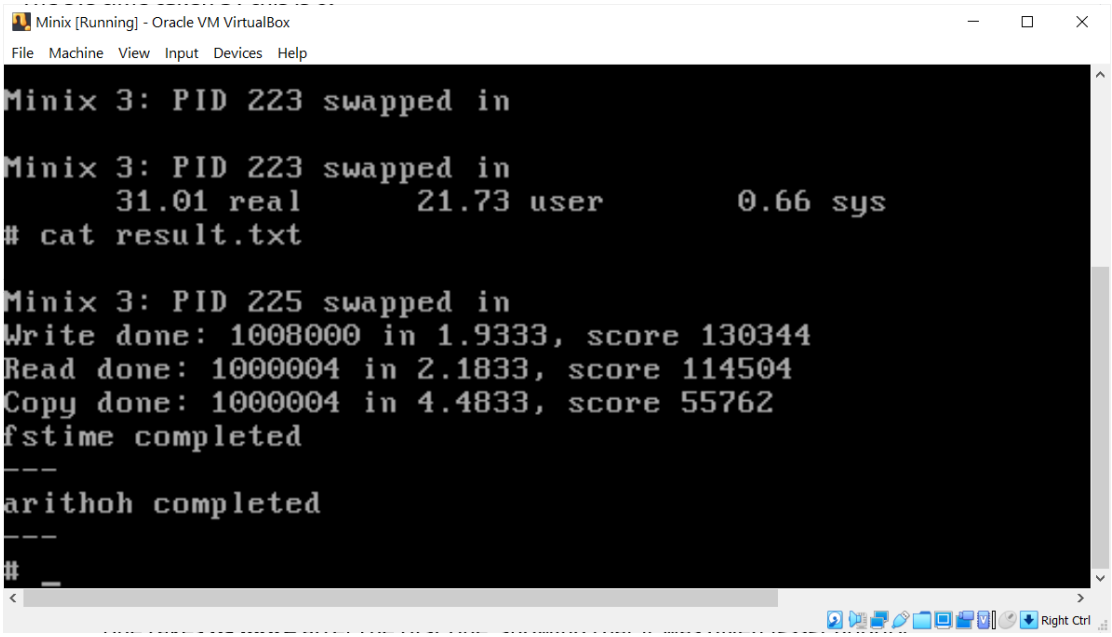


Above two images show the PID swapping between two numbers indicating PM switching between two processes as both have equal priorities. Finally, first process ends quite earlier than the second one this is because parent process is given higher priority than child process while scheduling. Later, to be allotted same priority. That explains the time difference between the two.

2. **Fstime - I/O bound task**

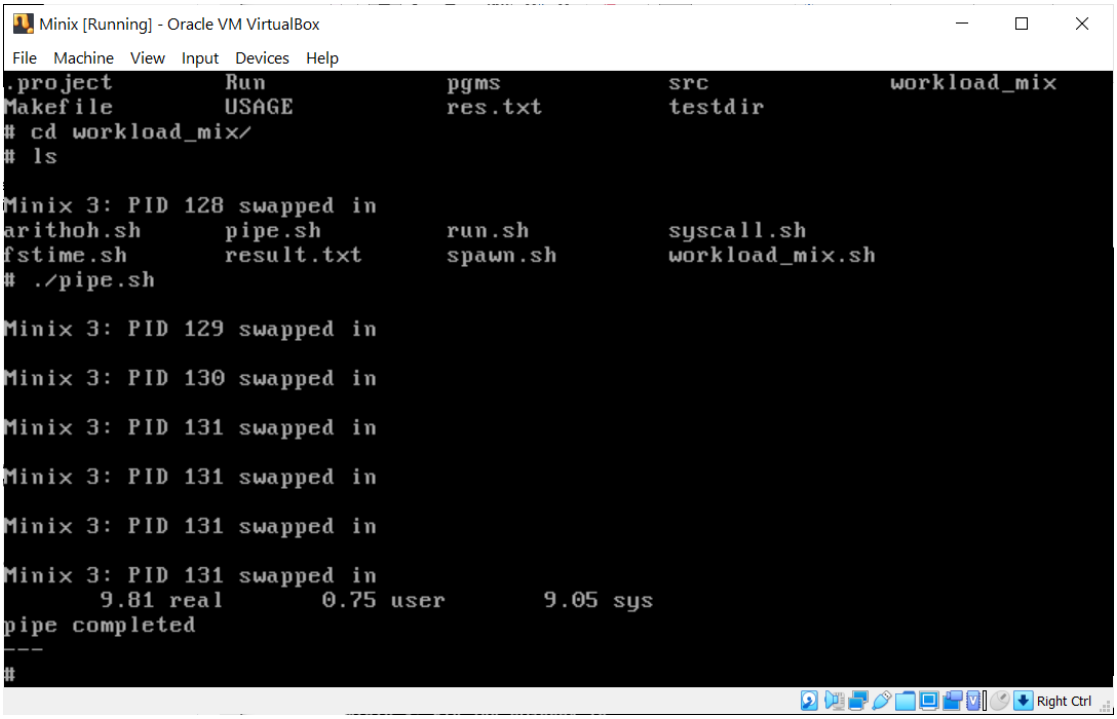
   Fstime vs arithoh

   Run arithoh and fstimecall processes simultaneously. As I/O processes are given higher priority, fstime seems to complete first followed by arithoh



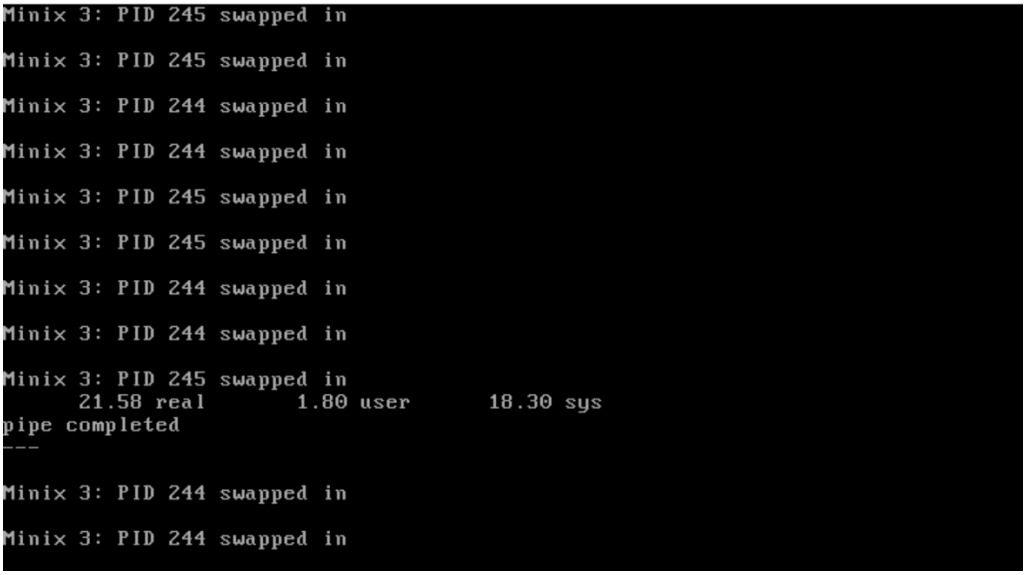
3. **Pipe- CPU bound task**

System time is very high compared to user time. This is because the of the Inter Process Communication protocols which are system based protocols.



Pipe vs arithoh:

Pipe's system time is more than user time and completes ahead of arithoh when run simultaneously.



```
Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in

Minix 3: PID 244 swapped in
      43.80 real      22.83 user       0.86 sys
arithoh completed
---
# _
```

4. **Spawn**

Spawn does the work of creating n number of process iteratively. The system time of the task is more than that of the user time. From this it can be inferred that initiated process work is less than that of the time it takes to initiate the process.

Spawn vs arithoh

Here, it is because of that, the user time of the arithoh is more than that of spawn because arithoh is CPU intensive task unlike processes created by spawn.

```
Minix 3: PID 43 swapped in

Minix 3: PID 44 swapped in

Minix 3: PID 45 swapped in

Minix 3: PID 46 swapped in

Minix 3: PID 47 swapped in

Minix 3: PID 48 swapped in
      28.86 real       0.71 user      23.33 sys
spawn completed
---

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in
```

```
Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in

Minix 3: PID 7 swapped in
      50.26 real       22.13 user       0.86 sys
arithoh completed
---
#
```

5. **Syscall- CPU bound benchmark**

   Most of the time spent is in system mode(5.18) and user time is very small(1.90)

```
Minix [Running] - Oracle VM VirtualBox
File  Machine  View  Input  Devices  Help
Minix 3: PID 135 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in

Minix 3: PID 136 swapped in
       7.11 real        1.90 user        5.18 sys
syscall completed
---
#
Minix 3: PID 91 swapped in

Minix 3: PID 91 swapped in
_
Right Ctrl
```

Syscall vs arithoh

When syscall and arithoh are run simultaneously, the scheduler handles it the same as fstime vs arithoh and uses round robin protocol. syscall completes task earlier than arithoh.