

CS314

Lab 3

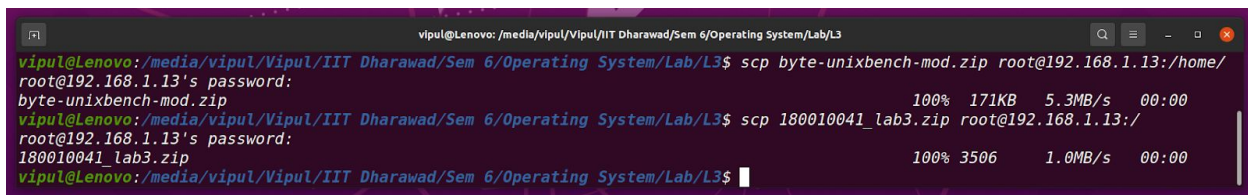
Vipul Yuvraj Nikam
180010041

Q1. Lab 3 Report?

Answer :

Part 0:

Transferring Files from Ubuntu to Minix.



```
vipul@Lenovo: /media/vipul/Vipul/IIT Dharwad/Sem 6/Operating System/Lab/L3
vipul@Lenovo: /media/vipul/Vipul/IIT Dharwad/Sem 6/Operating System/Lab/L3$ scp byte-unixbench-mod.zip root@192.168.1.13:/home/
root@192.168.1.13's password:
byte-unixbench-mod.zip                                100% 171KB  5.3MB/s  00:00
vipul@Lenovo: /media/vipul/Vipul/IIT Dharwad/Sem 6/Operating System/Lab/L3$ scp 180010041_lab3.zip root@192.168.1.13:/
root@192.168.1.13's password:
180010041_lab3.zip                                   100% 3506  1.0MB/s  00:00
vipul@Lenovo: /media/vipul/Vipul/IIT Dharwad/Sem 6/Operating System/Lab/L3$
```

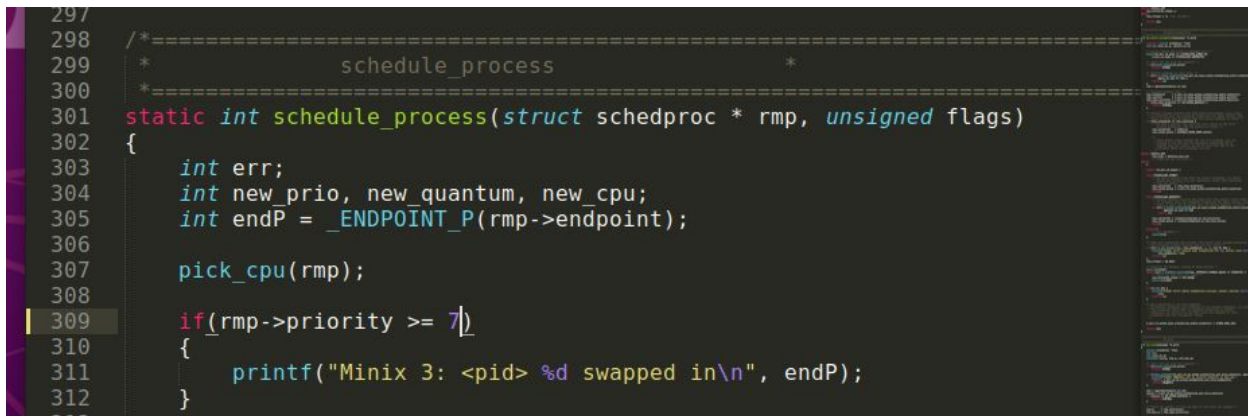
Part I:

To print when the user-level process is brought in by the scheduler, code modified in the file:

minix/servers/sched/schedule.c

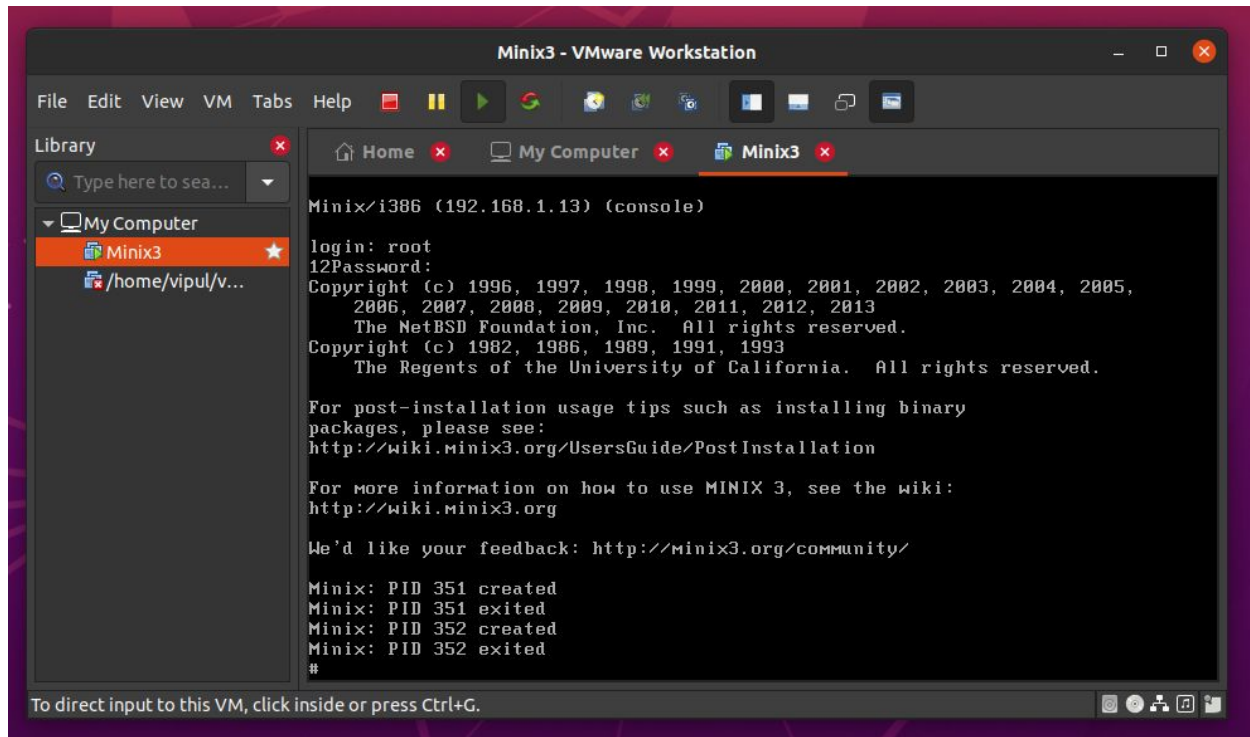
modified function

schedule_process ()



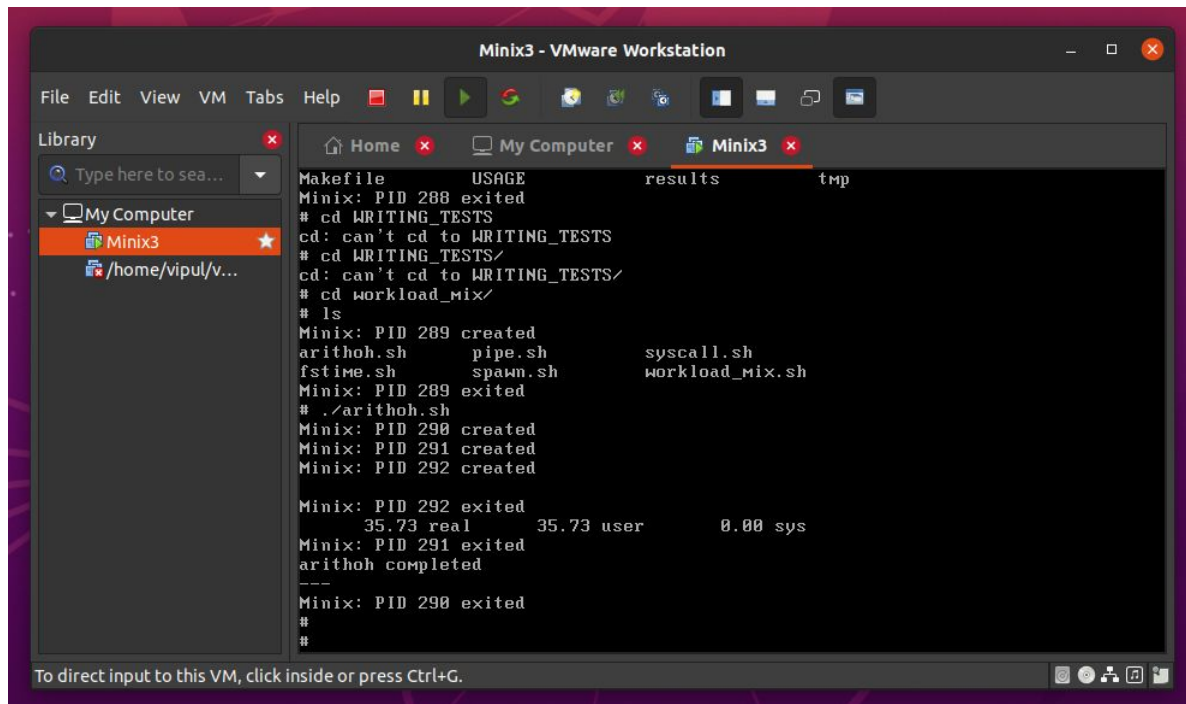
```
297
298 /*=====
299  *          schedule_process
300  *=====
301 static int schedule_process(struct schedproc * rmp, unsigned flags)
302 {
303     int err;
304     int new_prio, new_quantum, new_cpu;
305     int endP = _ENDPOINT_P(rmp->endpoint);
306
307     pick_cpu(rmp);
308
309     if(rmp->priority >= 7)
310     {
311         printf("Minix 3: <pid> %d swapped in\n", endP);
312     }
313 }
```

schedule.c



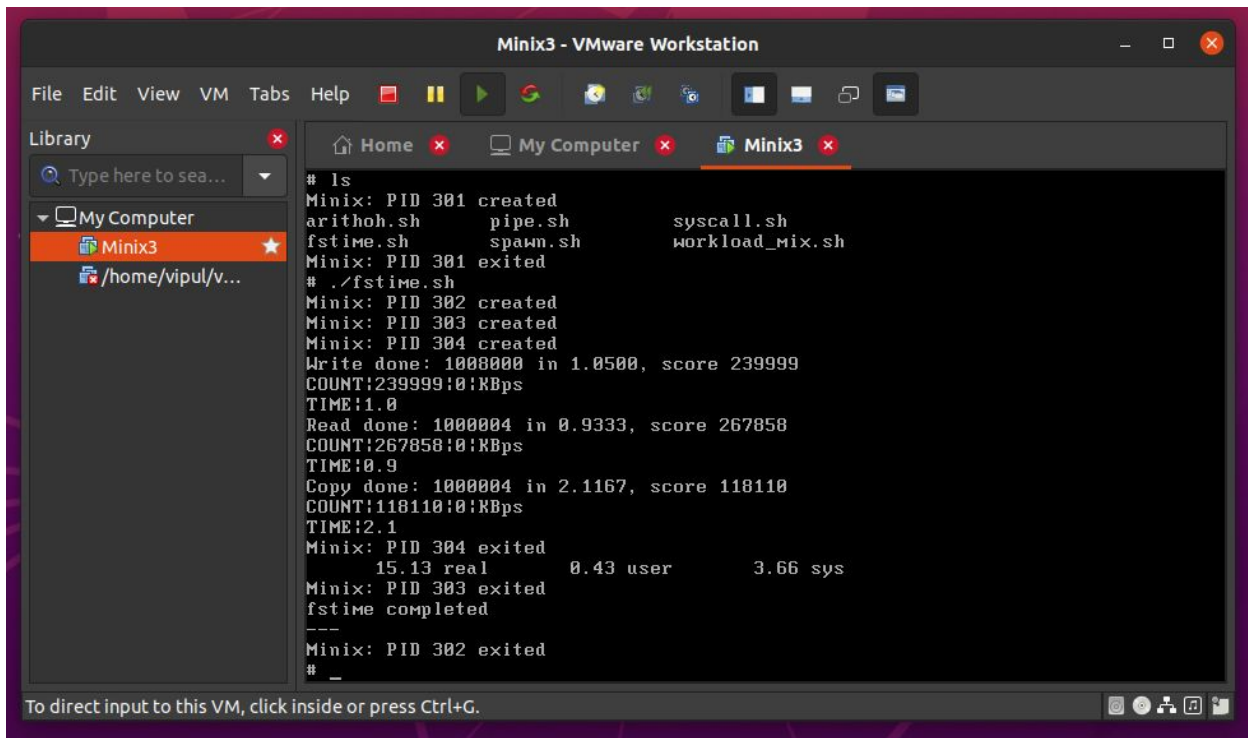
Part II:

- **Arithoh.sh:**



- CPU Bound Benchmark.
- It is observed that while running './arithoh.sh' command, the real & the user took the same time of **35.73** whereas the sys time taken is **0** as we can clearly see it in the Screenshot. The kernel scheduler log conforms with this and shows the message to schedule ./arithoh was sent 92 times consecutively.
- We can observe after running two executables of ./arithoh parallelly:
 - Almost turn-based programming in both the kernel scheduler and our print statements. However, the <pid> of one is not always followed by the other, and is sometimes changed again.
 - We also see that although both processes start at the same time, the second takes about 8 seconds longer after the first, which shows that it was given lower priority during scheduling.

● Fstime.sh:



The screenshot shows a VMware Workstation window titled "Minix3 - VMware Workstation". The interface includes a menu bar (File, Edit, View, VM, Tabs, Help), a toolbar with various icons, and a sidebar on the left labeled "Library" with a search bar and a tree view showing "My Computer" and "Minix3". The main terminal window displays the following output:

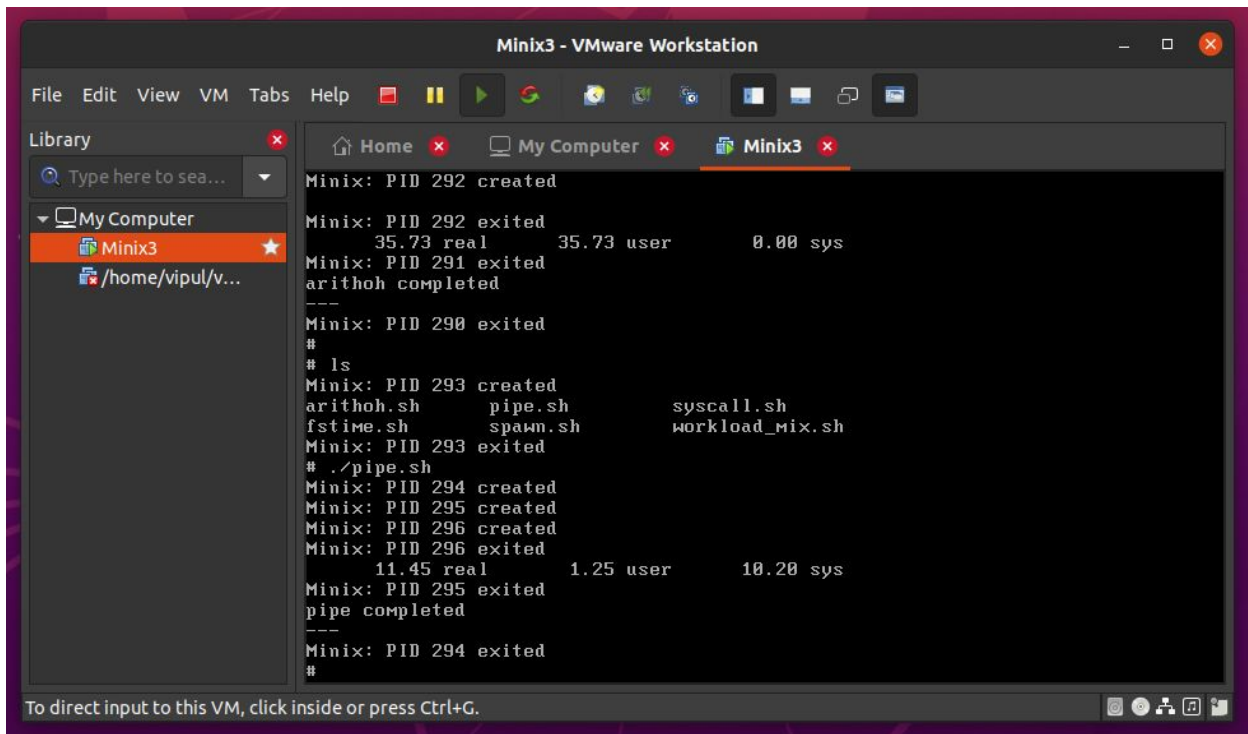
```
# ls
Minix: PID 301 created
arithoh.sh      pipe.sh      syscall.sh
fstime.sh       spawn.sh     workload_mix.sh
Minix: PID 301 exited
# ./fstime.sh
Minix: PID 302 created
Minix: PID 303 created
Minix: PID 304 created
Write done: 1000000 in 1.0500, score 239999
COUNT:239999:0:KBps
TIME:1.0
Read done: 1000004 in 0.9333, score 267858
COUNT:267858:0:KBps
TIME:0.9
Copy done: 1000004 in 2.1167, score 118110
COUNT:118110:0:KBps
TIME:2.1
Minix: PID 304 exited
      15.13 real      0.43 user      3.66 sys
Minix: PID 303 exited
fstime completed
---
Minix: PID 302 exited
# _
```

At the bottom of the window, a status bar reads: "To direct input to this VM, click inside or press Ctrl+G."

- IO Bound Benchmark.
- We can observe on running `fstime.sh` that '*total turnaround time*' is greater than '*sys time*' is greater than '*user time*' which occurs because the process needs to wait for its IO to complete before continuing.
- And if `./arithoh` and `./fstime` are executed at the same time we can see that Arithoh repeats for some time before IO is scheduled. Which demonstrates the scheduler's efficiency in utilizing the wait time of `./fstime` to schedule a CPU task like `./arithoh`.

● Pipe.sh:

- CPU Bound Benchmark. Because of Inter-Process Communication protocols, it takes a longer amount of time in sys compared to usr.
- If executed together with `./arithoh.sh`, works similar to previous case of `./fstime.sh`, where the pipe finishes earlier followed by consecutive scheduling of arithoh until Completion



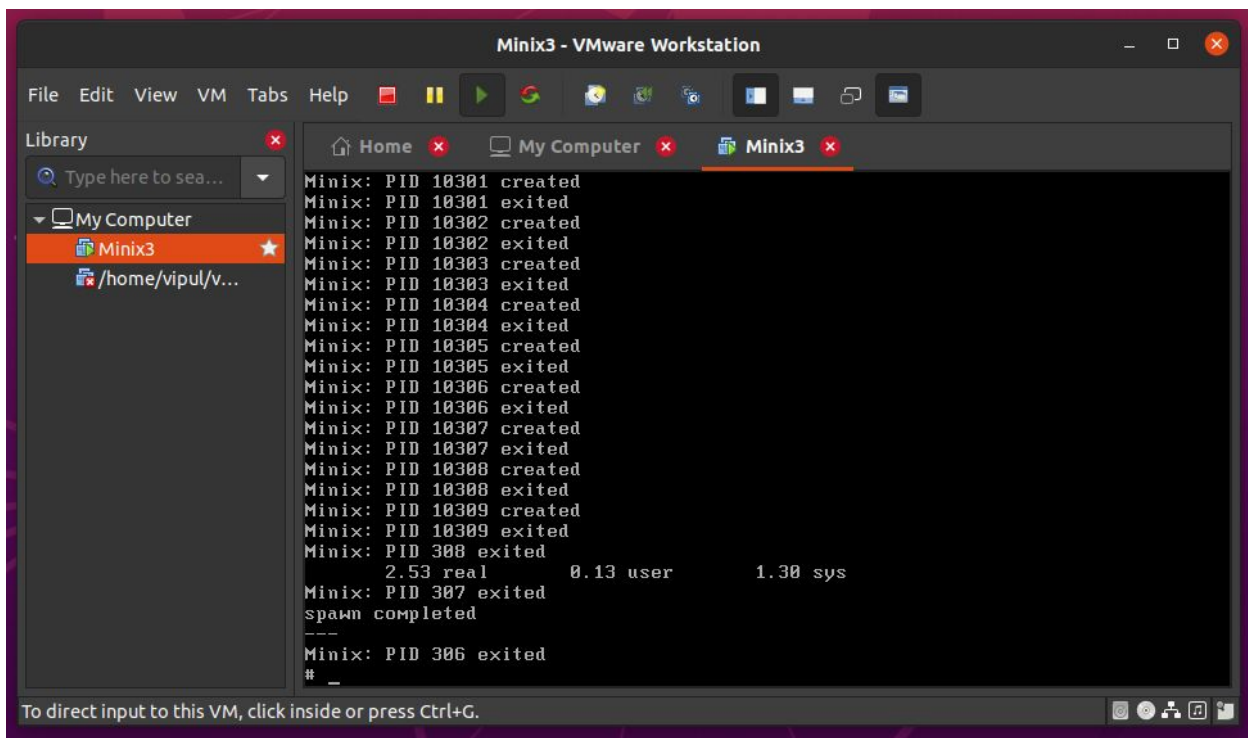
The screenshot shows a VMware Workstation window titled "Minix3 - VMware Workstation". The interface includes a menu bar (File, Edit, View, VM, Tabs, Help), a toolbar with various icons, and a sidebar on the left labeled "Library" with a search bar and a tree view showing "My Computer" and "Minix3". The main terminal window displays the following output:

```
Minix: PID 292 created
Minix: PID 292 exited      35.73 real      35.73 user      0.00 sys
Minix: PID 291 exited
arithoh completed
---
Minix: PID 290 exited
#
# ls
Minix: PID 293 created
arithoh.sh      pipe.sh      syscall.sh
fstime.sh      spawn.sh    workload_mix.sh
Minix: PID 293 exited
# ./pipe.sh
Minix: PID 294 created
Minix: PID 295 created
Minix: PID 296 created
Minix: PID 296 exited      11.45 real      1.25 user      10.20 sys
Minix: PID 295 exited
pipe completed
---
Minix: PID 294 exited
#
```

At the bottom of the window, a status bar reads: "To direct input to this VM, click inside or press Ctrl+G."

- **Spawn.sh:**

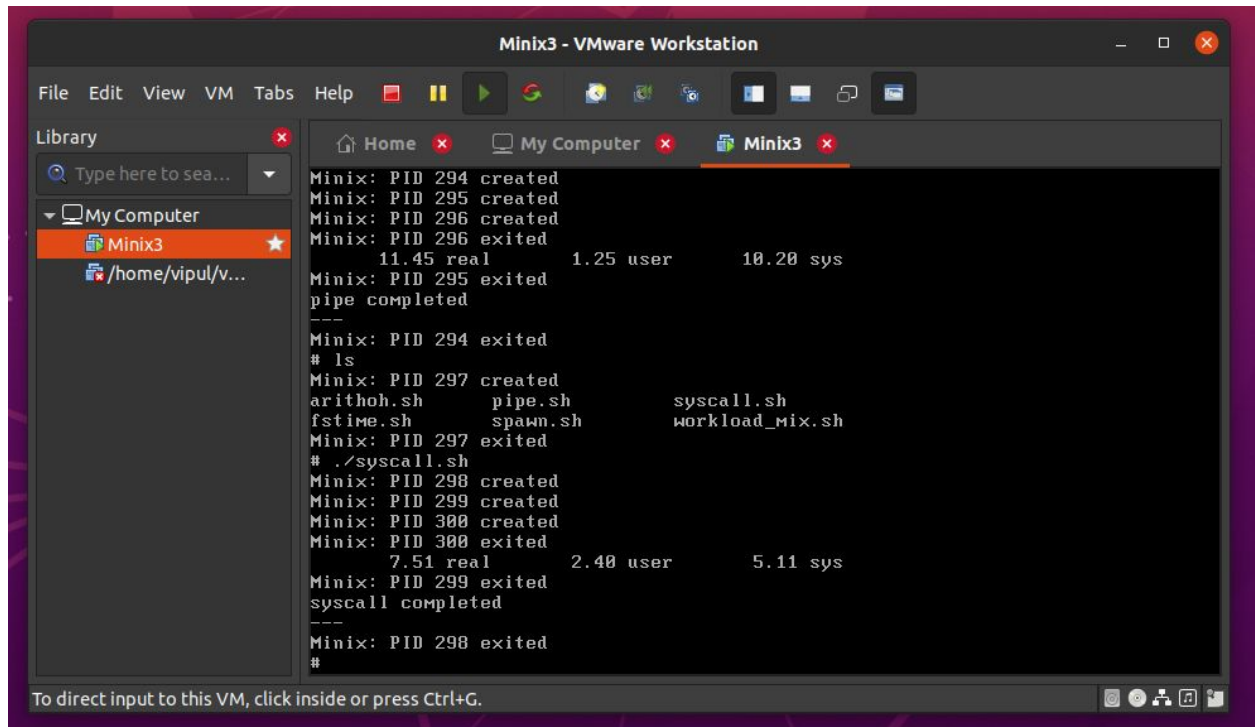
- CPU Bound Benchmark.
- Like in pipe, here too it takes a longer in sys compared to usr. We can see that a large number of processes ranging from 12 to 320 are exchanged in the queue consecutively.
- When executed together with arithoh.sh run, spawn ends earlier whereas arithoh continues to run until complete.



```
Minix: PID 10301 created
Minix: PID 10301 exited
Minix: PID 10302 created
Minix: PID 10302 exited
Minix: PID 10303 created
Minix: PID 10303 exited
Minix: PID 10304 created
Minix: PID 10304 exited
Minix: PID 10305 created
Minix: PID 10305 exited
Minix: PID 10306 created
Minix: PID 10306 exited
Minix: PID 10307 created
Minix: PID 10307 exited
Minix: PID 10308 created
Minix: PID 10308 exited
Minix: PID 10309 created
Minix: PID 10309 exited
Minix: PID 308 exited
2.53 real    0.13 user    1.30 sys
Minix: PID 307 exited
spawn completed
---
Minix: PID 306 exited
# _
```

- **Syscall.sh:**

- CPU Bound Benchmark.
- ‘*Real time*’ is greater than ‘*sys time*’ is greater than ‘*usr time*’.
- When executed together with ./arithoh.sh, runs in a round-robin manner with syscall completing first followed by arithoh.



XXX