CS 314: Operating Systems Laboratory
Lab 3

Nampally Pranav
190010026

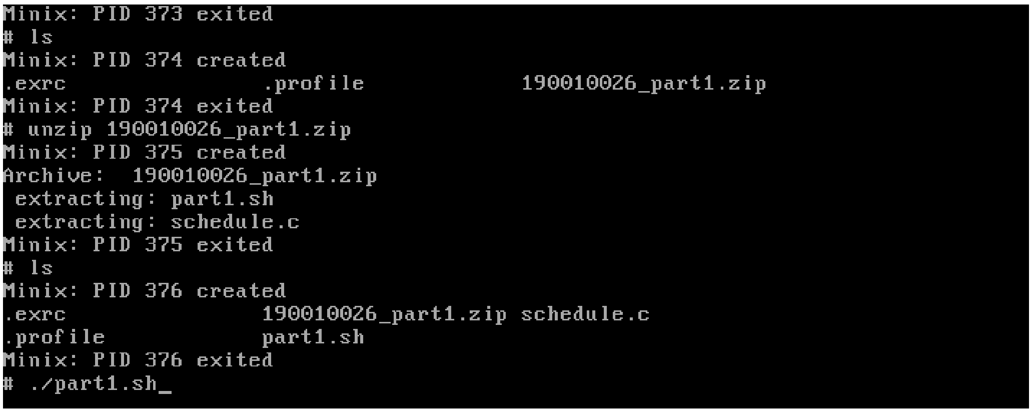
Part 1

NOTE: For all the parts, the work of sending files from Ubuntu to Minix was done using the scp command.

The task here was to modify the Minix3 source code such that the string "PID ¡pid¿ swapped in" was printed, whenever a user-level process is brought in by the scheduler.
From the links provided for reference, I could infer that Minix3 uses multilevel queuing system for scheduling. There were 2 directories given as hints. After going through them, I have modified the code in the file at location minix/servers/sched/schedule.c. Here the follwing lines were added at the schedule_process section to print the required statement(and swap) whenever the priority of the process was greater than the current process.:

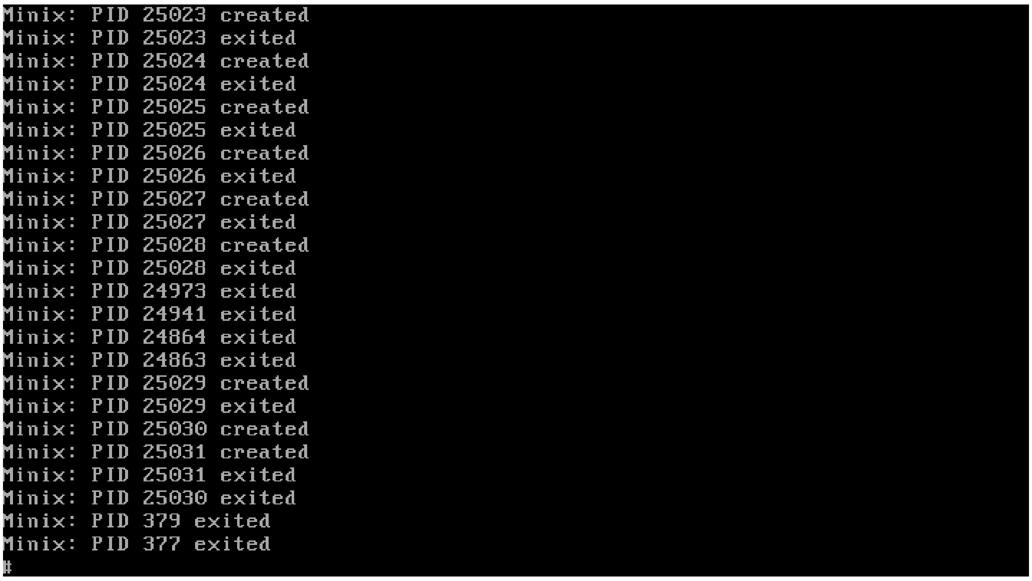
```
if (rmp->priority >= USER_Q){
printf("PID %d swapped in\n", rmp->endpoint);
}
```

For the changes to be reflected in the Minix VM, a script part1.sh was written whose work is to copy schedule.c to correct location and build the system to see changes.
So to make changes the following command should be run:
`./part1.sh`
The Screenshots for reference are given below:



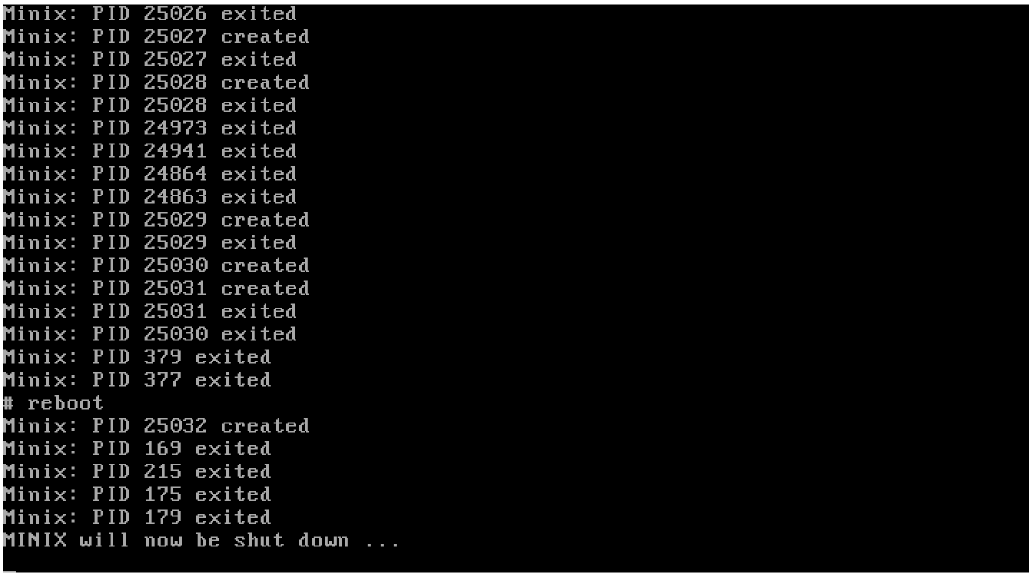
```
Minix: PID 373 exited
# ls
Minix: PID 374 created
.exrc               .profile            190010026_part1.zip
Minix: PID 374 exited
# unzip 190010026_part1.zip
Minix: PID 375 created
Archive:  190010026_part1.zip
 extracting: part1.sh
 extracting: schedule.c
Minix: PID 375 exited
# ls
Minix: PID 376 created
.exrc               190010026_part1.zip schedule.c
.profile            part1.sh
Minix: PID 376 exited
# ./part1.sh_
```

Figure 1: Running ./part1.sh



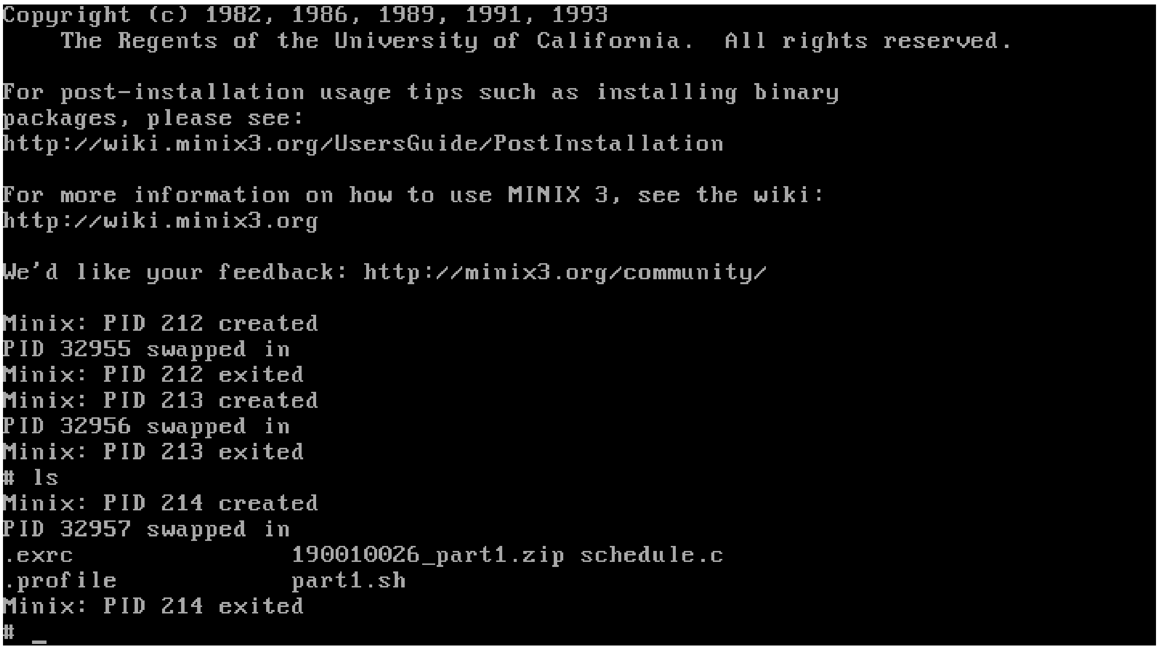
```
Minix: PID 25023 created
Minix: PID 25023 exited
Minix: PID 25024 created
Minix: PID 25024 exited
Minix: PID 25025 created
Minix: PID 25025 exited
Minix: PID 25026 created
Minix: PID 25026 exited
Minix: PID 25027 created
Minix: PID 25027 exited
Minix: PID 25028 created
Minix: PID 25028 exited
Minix: PID 24973 exited
Minix: PID 24941 exited
Minix: PID 24864 exited
Minix: PID 24863 exited
Minix: PID 25029 created
Minix: PID 25029 exited
Minix: PID 25030 created
Minix: PID 25031 created
Minix: PID 25031 exited
Minix: PID 25030 exited
Minix: PID 379 exited
Minix: PID 377 exited
#
```

Figure 2: Running



```
Minix: PID 25026 exited
Minix: PID 25027 created
Minix: PID 25027 exited
Minix: PID 25028 created
Minix: PID 25028 exited
Minix: PID 24973 exited
Minix: PID 24941 exited
Minix: PID 24864 exited
Minix: PID 24863 exited
Minix: PID 25029 created
Minix: PID 25029 exited
Minix: PID 25030 created
Minix: PID 25031 created
Minix: PID 25031 exited
Minix: PID 25030 exited
Minix: PID 379 exited
Minix: PID 377 exited
# reboot
Minix: PID 25032 created
Minix: PID 169 exited
Minix: PID 215 exited
Minix: PID 175 exited
Minix: PID 179 exited
MINIX will now be shut down ...
_
```

Figure 3: Running

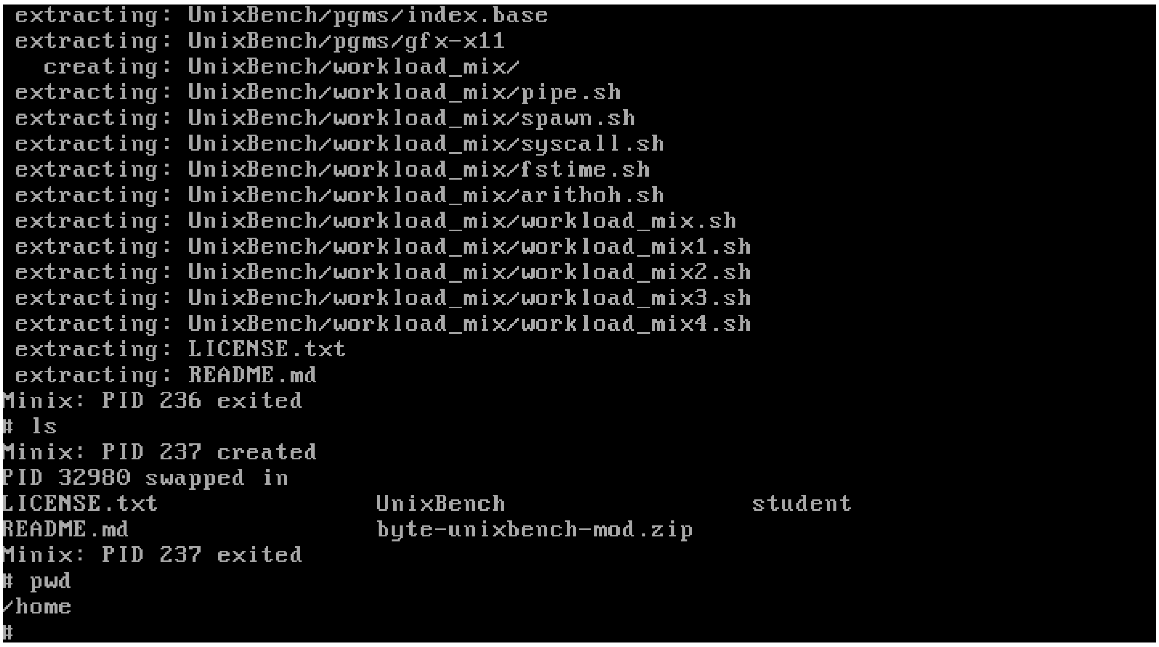


```
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

For post-installation usage tips such as installing binary
packages, please see:
http://wiki.minix3.org/UsersGuide/PostInstallation

For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org

We'd like your feedback: http://minix3.org/community/

Minix: PID 212 created
PID 32955 swapped in
Minix: PID 212 exited
Minix: PID 213 created
PID 32956 swapped in
Minix: PID 213 exited
# ls
Minix: PID 214 created
PID 32957 swapped in
.exrc               190010026_part1.zip schedule.c
.profile            part1.sh
Minix: PID 214 exited
# _
```

Figure 4: Required Output is being printed

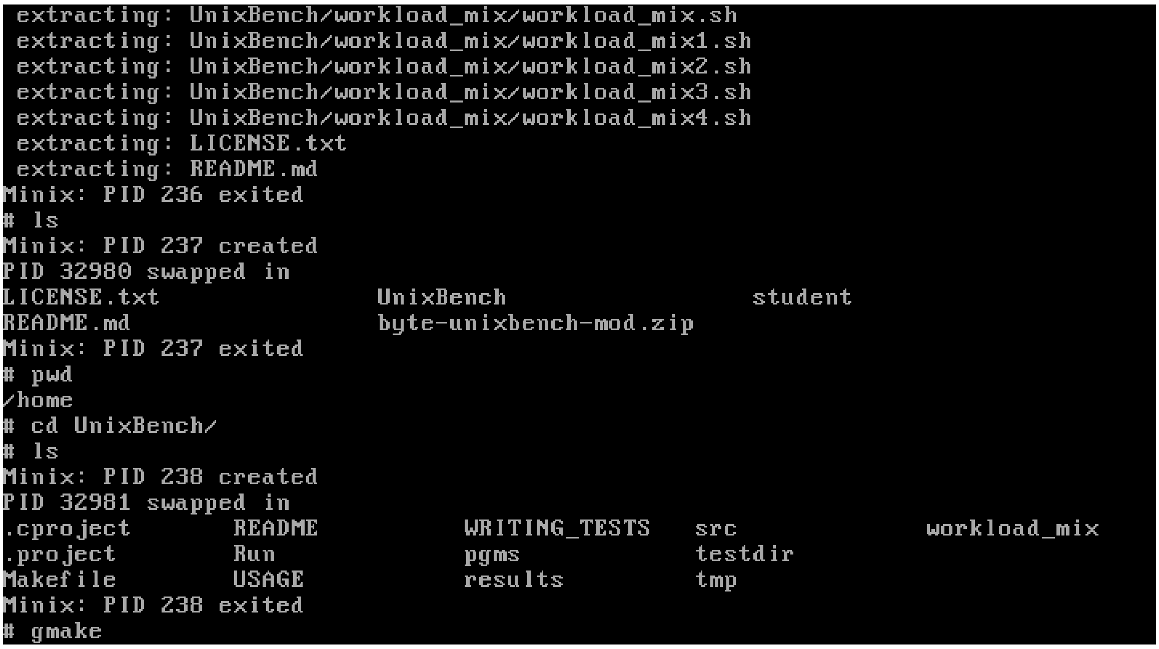
Part 2

For the UnixBench Benchmark Suite, some new new files were added in UnixBench/workload_mix in order to study the behavior of the scheduler by seeing the sequence of PID prints when these workloads are run.
We also see that the "time" command is used to run programs and summarize system resource usage for them.
Initial setup for UnixBench:



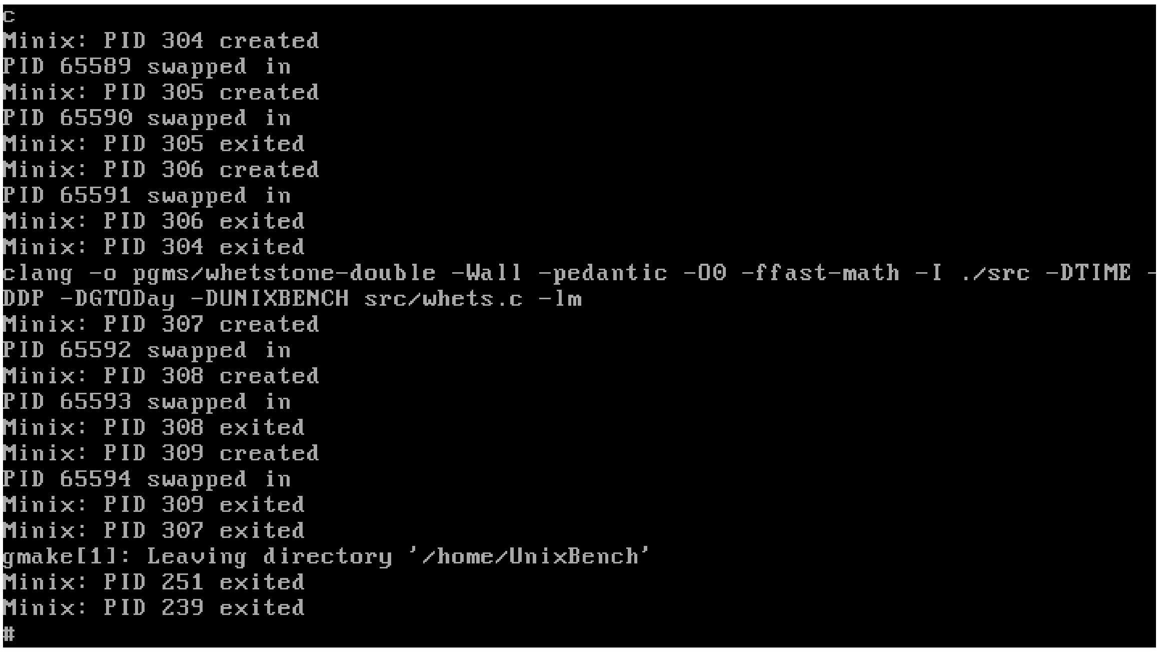
```
 extracting: UnixBench/pgms/index.base
 extracting: UnixBench/pgms/gfx-x11
   creating: UnixBench/workload_mix/
 extracting: UnixBench/workload_mix/pipe.sh
 extracting: UnixBench/workload_mix/spawn.sh
 extracting: UnixBench/workload_mix/syscall.sh
 extracting: UnixBench/workload_mix/fstime.sh
 extracting: UnixBench/workload_mix/arithoh.sh
 extracting: UnixBench/workload_mix/workload_mix.sh
 extracting: UnixBench/workload_mix/workload_mix1.sh
 extracting: UnixBench/workload_mix/workload_mix2.sh
 extracting: UnixBench/workload_mix/workload_mix3.sh
 extracting: UnixBench/workload_mix/workload_mix4.sh
 extracting: LICENSE.txt
 extracting: README.md
Minix: PID 236 exited
# ls
Minix: PID 237 created
PID 32980 swapped in
LICENSE.txt               UnixBench                 student
README.md                 byte-unixbench-mod.zip
Minix: PID 237 exited
# pwd
/home
#
```

Figure 5: Unzipping byte-unixbench-mod.zip in home dir.



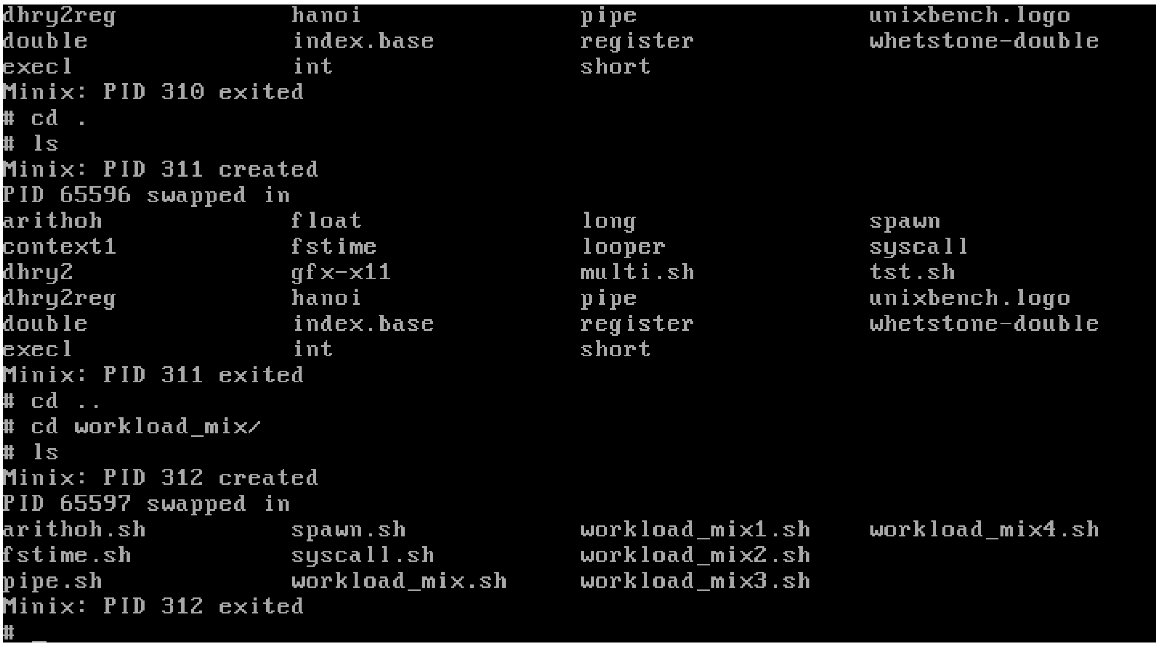
```
 extracting: UnixBench/workload_mix/workload_mix.sh
 extracting: UnixBench/workload_mix/workload_mix1.sh
 extracting: UnixBench/workload_mix/workload_mix2.sh
 extracting: UnixBench/workload_mix/workload_mix3.sh
 extracting: UnixBench/workload_mix/workload_mix4.sh
 extracting: LICENSE.txt
 extracting: README.md
Minix: PID 236 exited
# ls
Minix: PID 237 created
PID 32980 swapped in
LICENSE.txt             UnixBench                 student
README.md               byte-unixbench-mod.zip
Minix: PID 237 exited
# pwd
/home
# cd UnixBench/
# ls
Minix: PID 238 created
PID 32981 swapped in
.cproject       README          WRITING_TESTS   src             workload_mix
.project        Run             pgms            testdir
Makefile        USAGE           results         tmp
Minix: PID 238 exited
# gmake
```

Figure 6: Using gmake



```
c
Minix: PID 304 created
PID 65589 swapped in
Minix: PID 305 created
PID 65590 swapped in
Minix: PID 305 exited
Minix: PID 306 created
PID 65591 swapped in
Minix: PID 306 exited
Minix: PID 304 exited
clang -o pgms/whetstone-double -Wall -pedantic -O0 -ffast-math -I ./src -DTIME -
DDP -DGTODay -DUNIXBENCH src/whets.c -lm
Minix: PID 307 created
PID 65592 swapped in
Minix: PID 308 created
PID 65593 swapped in
Minix: PID 308 exited
Minix: PID 309 created
PID 65594 swapped in
Minix: PID 309 exited
Minix: PID 307 exited
gmake[1]: Leaving directory '/home/UnixBench'
Minix: PID 251 exited
Minix: PID 239 exited
#
```

Figure 7: gmake running



```
dhry2reg            hanoi               pipe                unixbench.logo
double              index.base          register            whetstone-double
execl               int                 short
Minix: PID 310 exited
# cd .
# ls
Minix: PID 311 created
PID 65596 swapped in
arithoh             float               long                spawn
context1            fstime              looper              syscall
dhry2               gfx-x11             multi.sh            tst.sh
dhry2reg            hanoi               pipe                unixbench.logo
double              index.base          register            whetstone-double
execl               int                 short
Minix: PID 311 exited
# cd ..
# cd workload_mix/
# ls
Minix: PID 312 created
PID 65597 swapped in
arithoh.sh          spawn.sh            workload_mix1.sh    workload_mix4.sh
fstime.sh           syscall.sh          workload_mix2.sh
pipe.sh             workload_mix.sh     workload_mix3.sh
Minix: PID 312 exited
# _
```

Figure 8: Completed and Ready for running the workload_mix*.sh files

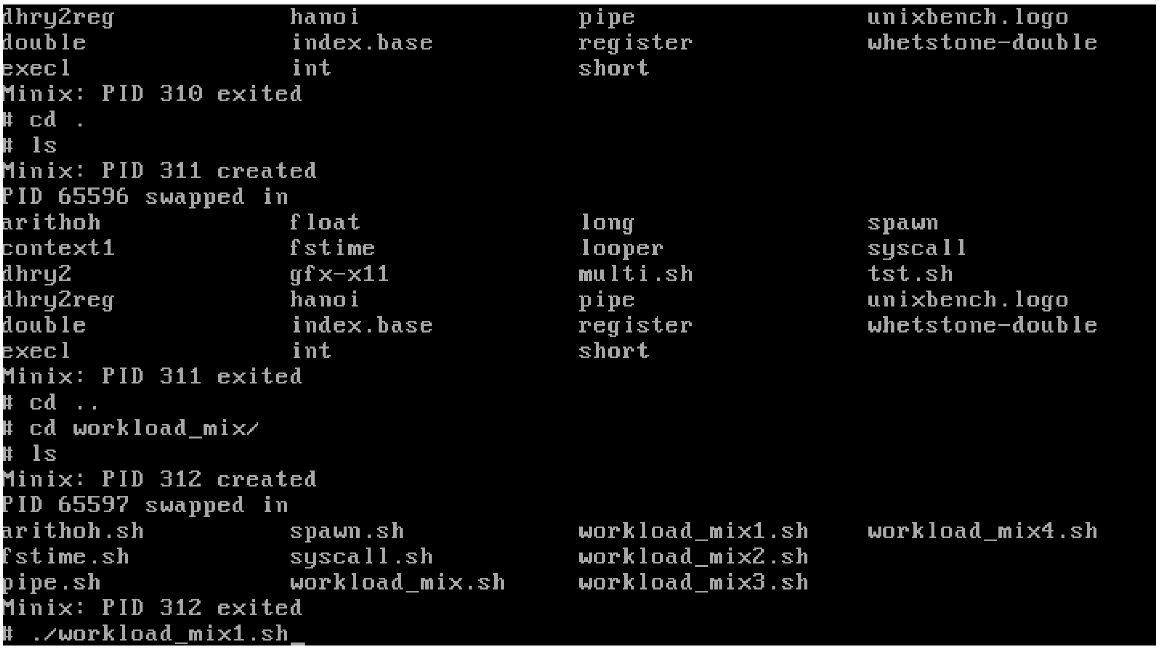
After an examination of the files, I could infer that, arithoh.sh is a CPU intensive task, fstime.sh is a I/O task, syscall.sh is a also a CPU intensive but uses system calls. I have used these 3 to create the workoads.
The following subsections show the 3 different combinations of workload_mix*.sh files being implemented and an explanation for the output received is given.

workload_mix1.sh

The code is:

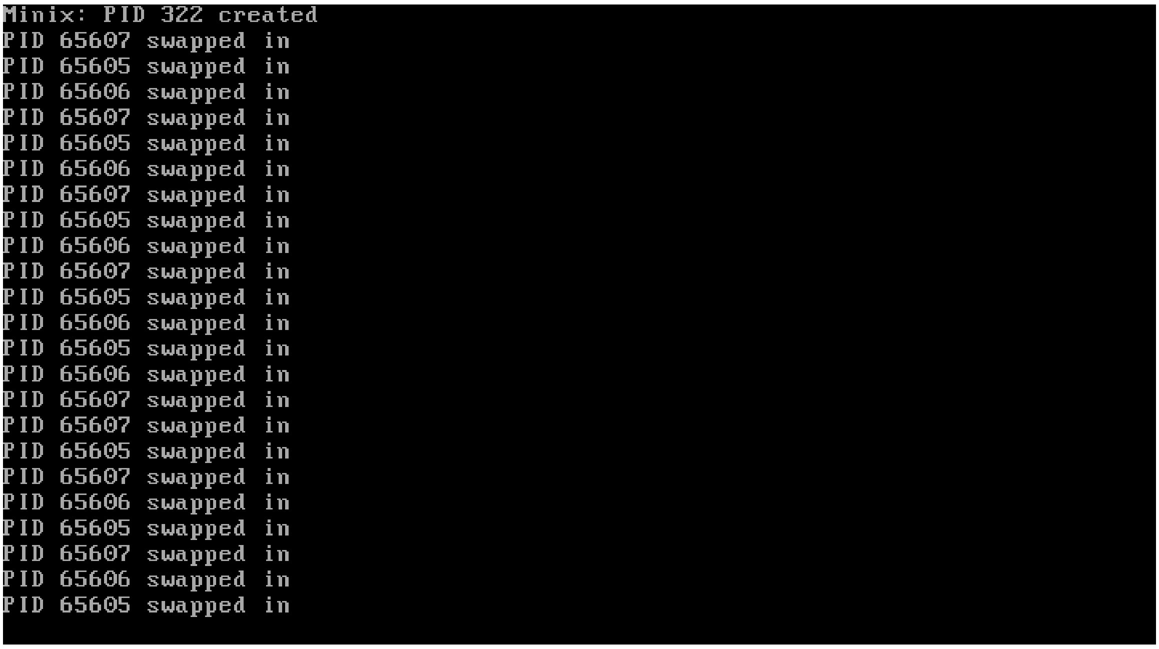
```
#!/bin/sh
./arithoh.sh &
./arithoh.sh &
./arithoh.sh &
wait
```

1st process has PID: 65605, 2nd one has PID: 65606, last process has PID: 65607. We see that the processes run alternatively one after the other. That is, 1st proc then 2nd proc then 3rd proc then this cycle repeats. At the end the arithoh completed is printed. So, we can see that if CPU intense processes start simultaneously then they get scheduled alternatively.



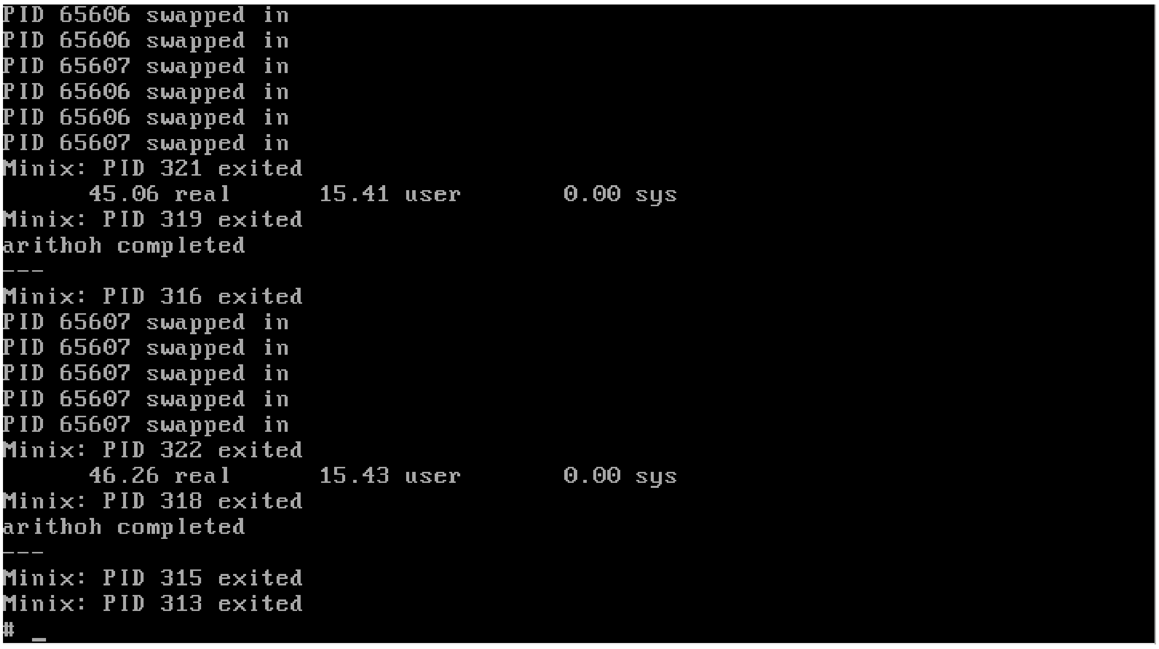
```
dhry2reg            hanoi               pipe                unixbench.logo
double              index.base          register            whetstone-double
execl               int                 short
Minix: PID 310 exited
# cd .
# ls
Minix: PID 311 created
PID 65596 swapped in
arithoh             float               long                spawn
context1            fstime              looper              syscall
dhry2               gfx-x11             multi.sh            tst.sh
dhry2reg            hanoi               pipe                unixbench.logo
double              index.base          register            whetstone-double
execl               int                 short
Minix: PID 311 exited
# cd ..
# cd workload_mix/
# ls
Minix: PID 312 created
PID 65597 swapped in
arithoh.sh          spawn.sh            workload_mix1.sh    workload_mix4.sh
fstime.sh           syscall.sh          workload_mix2.sh
pipe.sh             workload_mix.sh     workload_mix3.sh
Minix: PID 312 exited
# ./workload_mix1.sh_
```

Figure 9: Executing workload_mix1.sh



```
Minix: PID 322 created
PID 65607 swapped in
PID 65605 swapped in
PID 65606 swapped in
PID 65607 swapped in
PID 65605 swapped in
PID 65606 swapped in
PID 65607 swapped in
PID 65605 swapped in
PID 65606 swapped in
PID 65607 swapped in
PID 65605 swapped in
PID 65606 swapped in
PID 65605 swapped in
PID 65606 swapped in
PID 65607 swapped in
PID 65607 swapped in
PID 65605 swapped in
PID 65607 swapped in
PID 65606 swapped in
PID 65605 swapped in
PID 65607 swapped in
PID 65606 swapped in
PID 65605 swapped in
```

Figure 10: Running



```
PID 65606 swapped in
PID 65606 swapped in
PID 65607 swapped in
PID 65606 swapped in
PID 65606 swapped in
PID 65607 swapped in
Minix: PID 321 exited
      45.06 real       15.41 user        0.00 sys
Minix: PID 319 exited
arithoh completed
---
Minix: PID 316 exited
PID 65607 swapped in
PID 65607 swapped in
PID 65607 swapped in
PID 65607 swapped in
PID 65607 swapped in
Minix: PID 322 exited
      46.26 real       15.43 user        0.00 sys
Minix: PID 318 exited
arithoh completed
---
Minix: PID 315 exited
Minix: PID 313 exited
# _
```

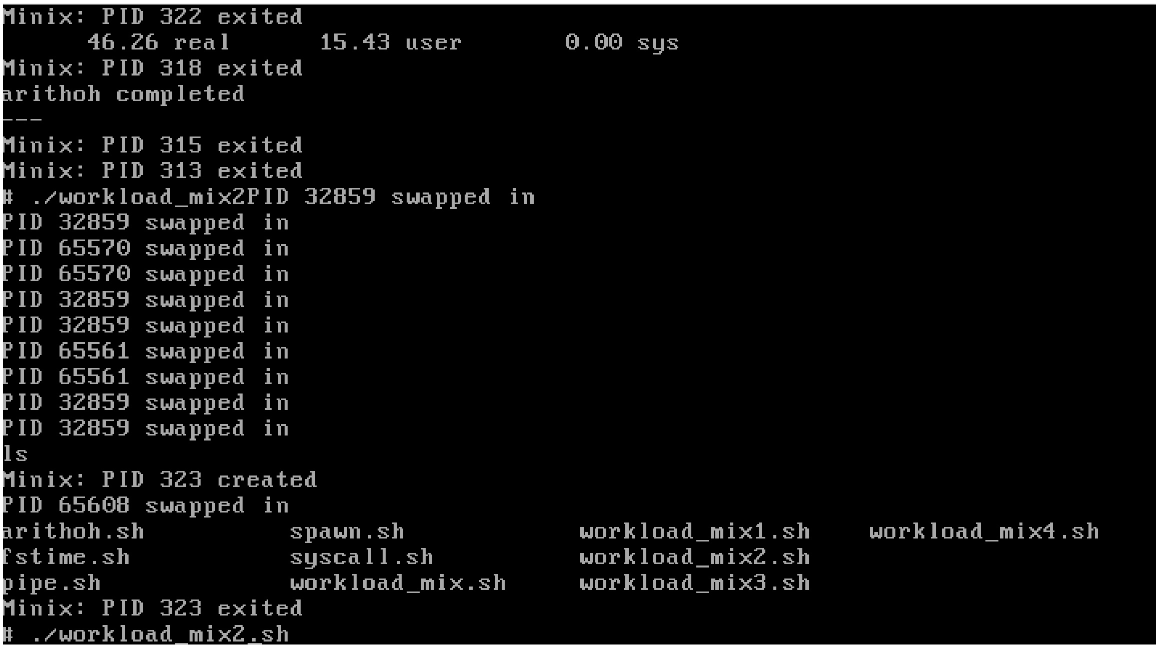
Figure 11: Completed

workload_mix2.sh

The code is:

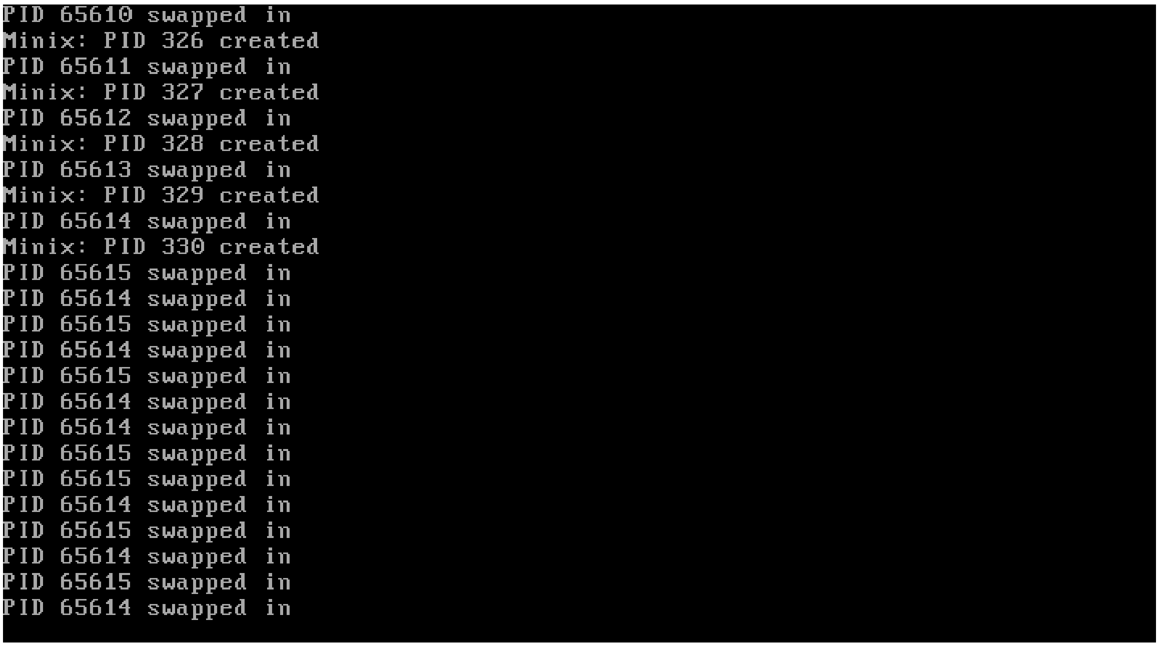
```
#!/bin/sh
./arithoh.sh &
./syscall.sh &
wait
```

We see that the process for arithoh.sh has PID: 65614 and for syscall.sh the process PID is: 65615. Here too we see that both are CPU intensive but we see them alternating their positions till the syscall.sh completes (syscall completes 1st maybe because it only makes system calls thus is a little less CPU intensive than arithoh.sh). Then after this only PID: 65614 runs continuously till the process for arithoh.sh completes.



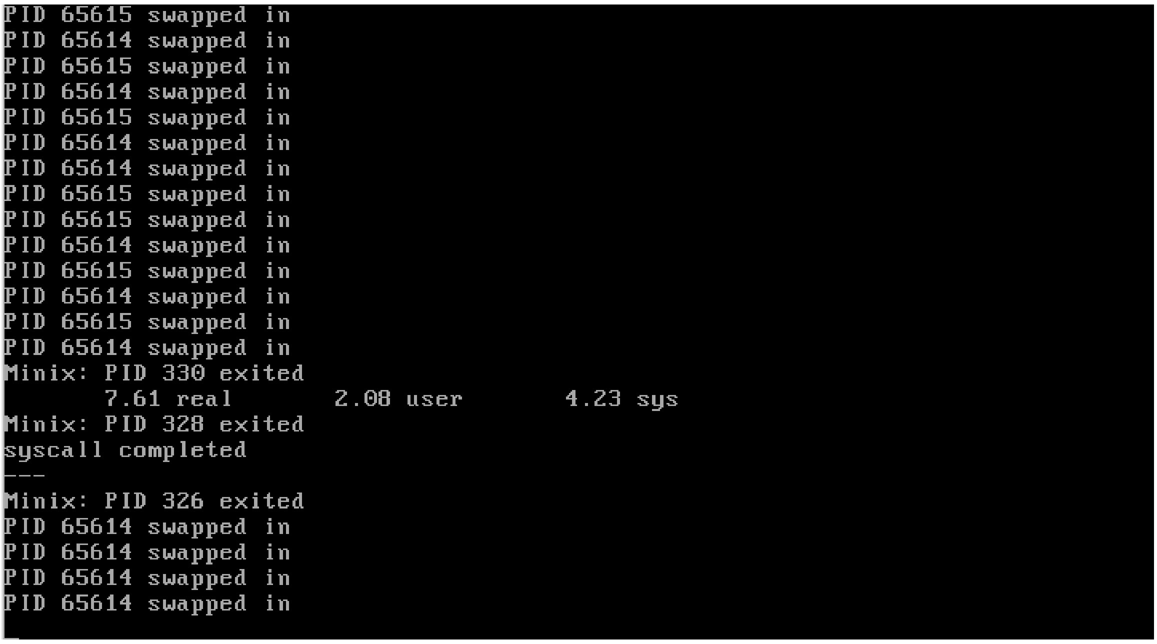
```
Minix: PID 322 exited
      46.26 real      15.43 user       0.00 sys
Minix: PID 318 exited
arithoh completed
---
Minix: PID 315 exited
Minix: PID 313 exited
# ./workload_mix2PID 32859 swapped in
PID 32859 swapped in
PID 65570 swapped in
PID 65570 swapped in
PID 32859 swapped in
PID 32859 swapped in
PID 65561 swapped in
PID 65561 swapped in
PID 32859 swapped in
PID 32859 swapped in
ls
Minix: PID 323 created
PID 65608 swapped in
arithoh.sh          spawn.sh            workload_mix1.sh    workload_mix4.sh
fstime.sh           syscall.sh          workload_mix2.sh
pipe.sh             workload_mix.sh     workload_mix3.sh
Minix: PID 323 exited
# ./workload_mix2.sh
```

Figure 12: Executing workload_mix2.sh



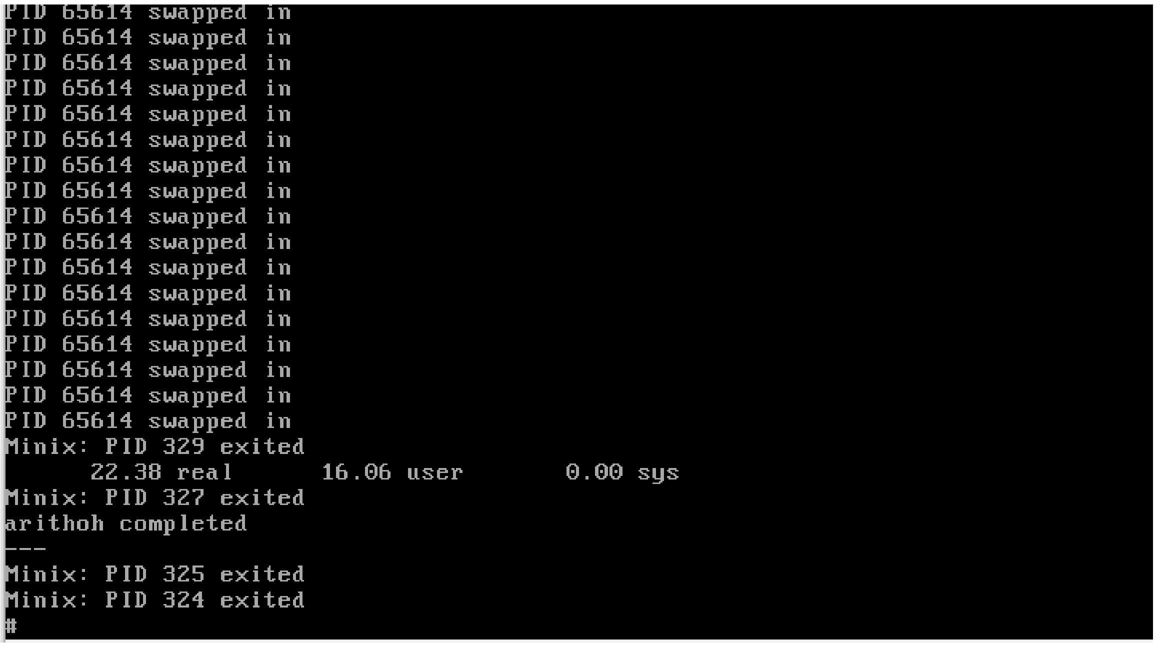
```
PID 65610 swapped in
Minix: PID 326 created
PID 65611 swapped in
Minix: PID 327 created
PID 65612 swapped in
Minix: PID 328 created
PID 65613 swapped in
Minix: PID 329 created
PID 65614 swapped in
Minix: PID 330 created
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
```

Figure 13: Running



```
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
PID 65615 swapped in
PID 65614 swapped in
Minix: PID 330 exited
       7.61 real       2.08 user       4.23 sys
Minix: PID 328 exited
syscall completed
---
Minix: PID 326 exited
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
_
```

Figure 14: Running



```
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
Minix: PID 329 exited
      22.38 real      16.06 user       0.00 sys
Minix: PID 327 exited
arithoh completed
---
Minix: PID 325 exited
Minix: PID 324 exited
#
```

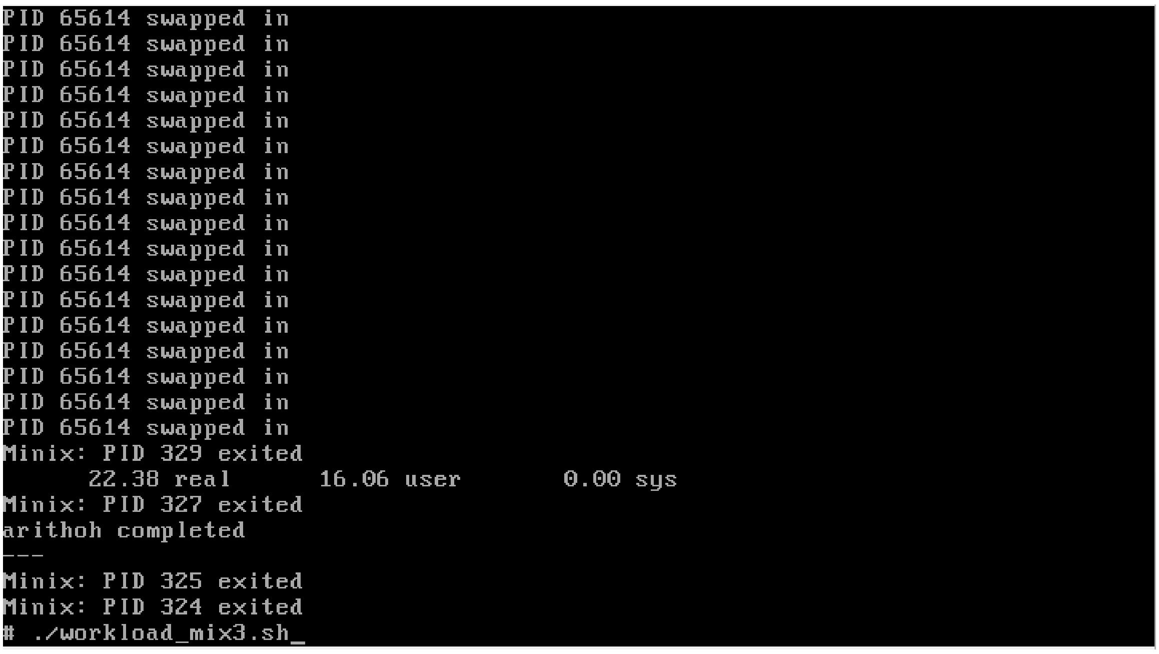
Figure 15: Completed

workload_mix3.sh

The code is:

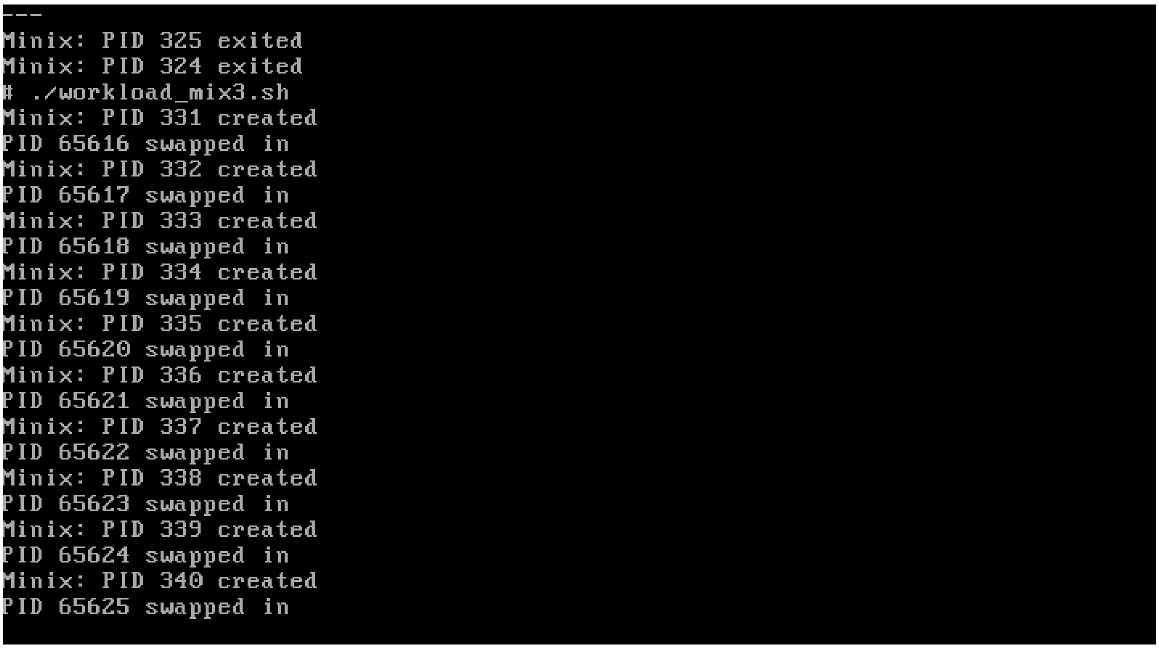
```
#!/bin/sh
./fstime.sh &
./fstime.sh &
./fstime.sh &
wait
```

The similar processes the PIDs in the following order, 1st proc: 65623, 2nd: 65624, 3rd: 65625. We see that all are I/O processes, so they all wait for input. The one which receives the input first completes terminates. Here the 2nd terminates first then 1st then the 3rd one as they receive input in that order.



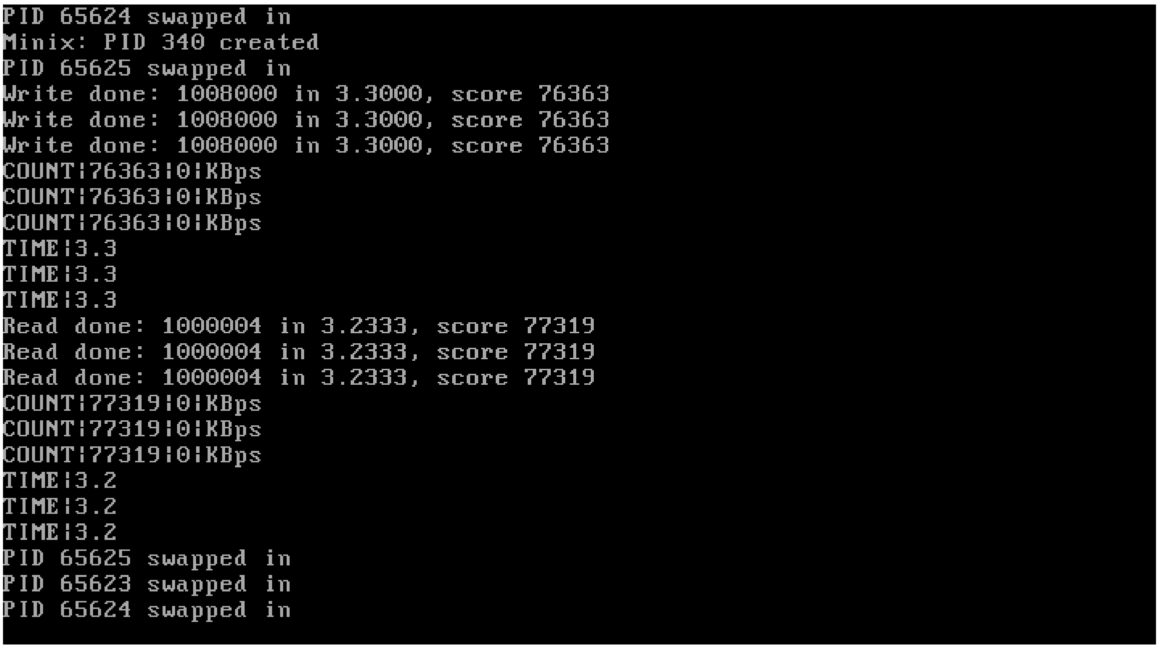
```
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
PID 65614 swapped in
Minix: PID 329 exited
      22.38 real       16.06 user        0.00 sys
Minix: PID 327 exited
arithoh completed
---
Minix: PID 325 exited
Minix: PID 324 exited
# ./workload_mix3.sh_
```

Figure 16: Executing workload_mix3.sh



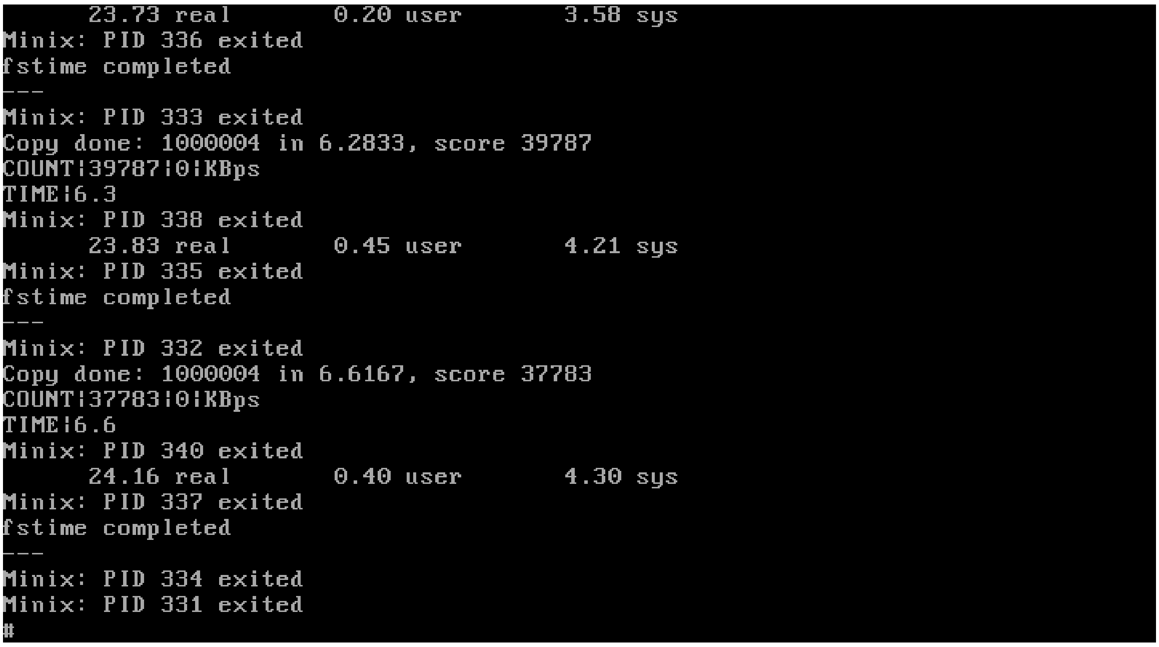
```
---
Minix: PID 325 exited
Minix: PID 324 exited
# ./workload_mix3.sh
Minix: PID 331 created
PID 65616 swapped in
Minix: PID 332 created
PID 65617 swapped in
Minix: PID 333 created
PID 65618 swapped in
Minix: PID 334 created
PID 65619 swapped in
Minix: PID 335 created
PID 65620 swapped in
Minix: PID 336 created
PID 65621 swapped in
Minix: PID 337 created
PID 65622 swapped in
Minix: PID 338 created
PID 65623 swapped in
Minix: PID 339 created
PID 65624 swapped in
Minix: PID 340 created
PID 65625 swapped in
```

Figure 17: Running



```
PID 65624 swapped in
Minix: PID 340 created
PID 65625 swapped in
Write done: 1008000 in 3.3000, score 76363
Write done: 1008000 in 3.3000, score 76363
Write done: 1008000 in 3.3000, score 76363
COUNT|76363|0|KBps
COUNT|76363|0|KBps
COUNT|76363|0|KBps
TIME|3.3
TIME|3.3
TIME|3.3
Read done: 1000004 in 3.2333, score 77319
Read done: 1000004 in 3.2333, score 77319
Read done: 1000004 in 3.2333, score 77319
COUNT|77319|0|KBps
COUNT|77319|0|KBps
COUNT|77319|0|KBps
TIME|3.2
TIME|3.2
TIME|3.2
PID 65625 swapped in
PID 65623 swapped in
PID 65624 swapped in
```

Figure 18: Running



```
      23.73 real       0.20 user       3.58 sys
Minix: PID 336 exited
fstime completed
---
Minix: PID 333 exited
Copy done: 1000004 in 6.2833, score 39787
COUNT|39787|0|KBps
TIME|6.3
Minix: PID 338 exited
      23.83 real       0.45 user       4.21 sys
Minix: PID 335 exited
fstime completed
---
Minix: PID 332 exited
Copy done: 1000004 in 6.6167, score 37783
COUNT|37783|0|KBps
TIME|6.6
Minix: PID 340 exited
      24.16 real       0.40 user       4.30 sys
Minix: PID 337 exited
fstime completed
---
Minix: PID 334 exited
Minix: PID 331 exited
#
```

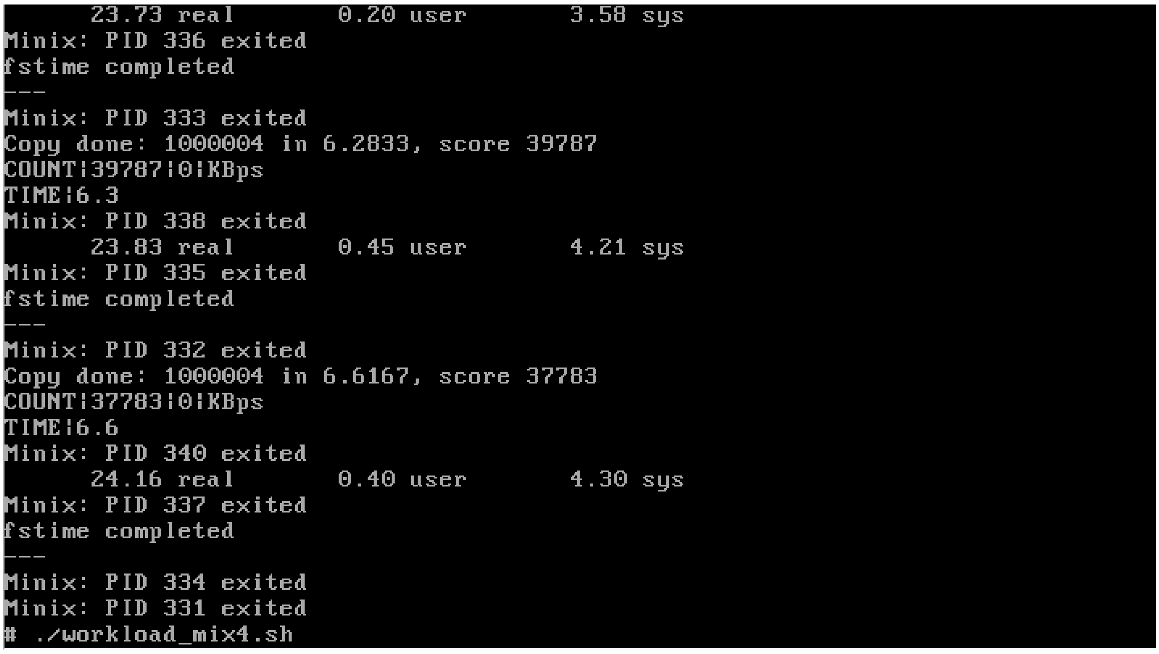
Figure 19: Completed

workload_mix4.sh

The code is:

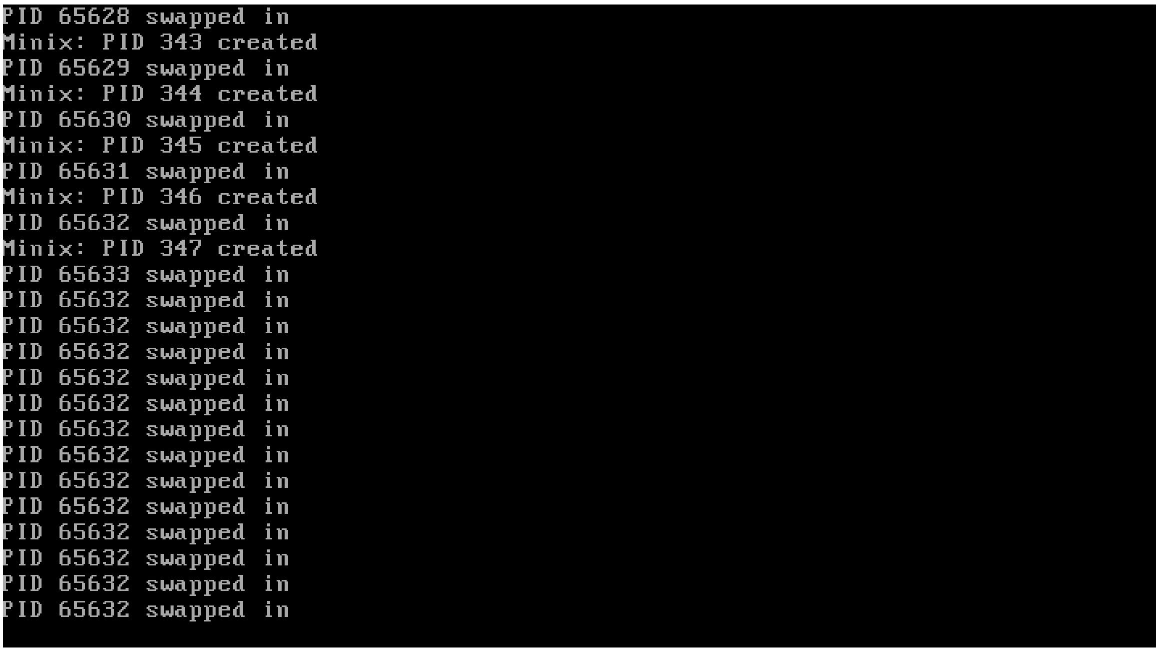
```
#!/bin/sh
./arithoh.sh &
./fstime.sh &
wait
```

Here arithoh,sh process PID id: 65632, and PID for the fstime.sh process is: 65633. We know that arithoh.sh is CPU intensive whereas fstime.sh is I/O task. So, initially we see that PID for arithoh.sh is being swapped whereas the fstime.sh process waits for the input. Then we see that fstime.sh gets its input it is scheduled and is completed before arithoh.sh (arithoh.sh gets into waiting state) then arithoh.sh is scheduled till its completion.



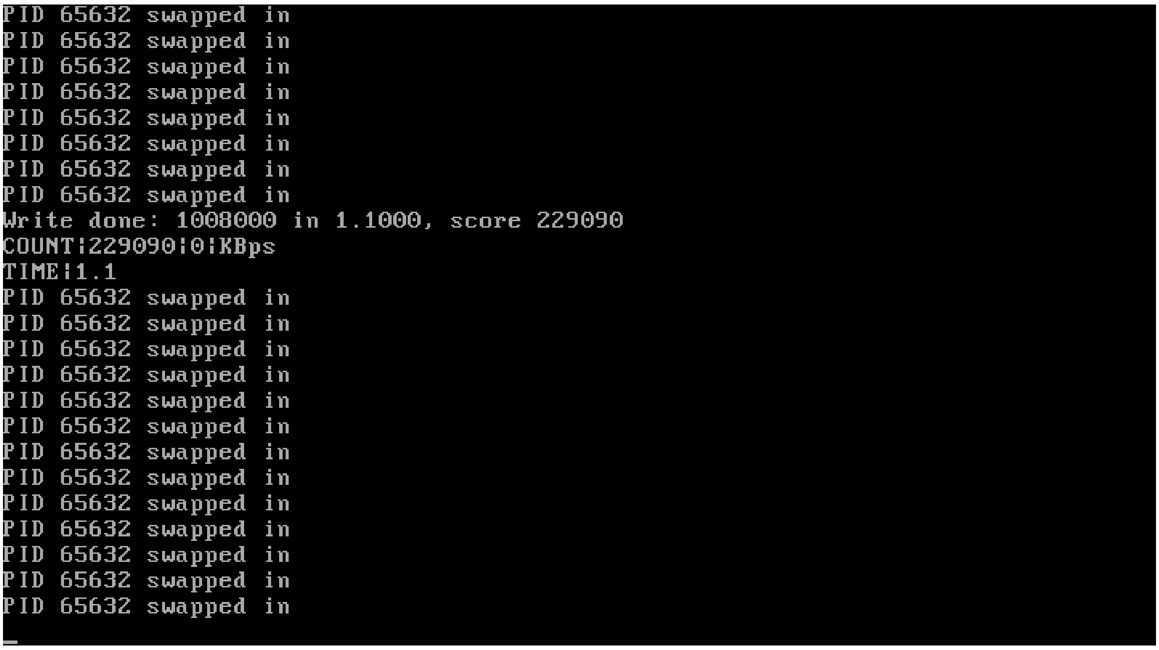
```
      23.73 real       0.20 user       3.58 sys
Minix: PID 336 exited
fstime completed
---
Minix: PID 333 exited
Copy done: 1000004 in 6.2833, score 39787
COUNT|39787|0|KBps
TIME|6.3
Minix: PID 338 exited
      23.83 real       0.45 user       4.21 sys
Minix: PID 335 exited
fstime completed
---
Minix: PID 332 exited
Copy done: 1000004 in 6.6167, score 37783
COUNT|37783|0|KBps
TIME|6.6
Minix: PID 340 exited
      24.16 real       0.40 user       4.30 sys
Minix: PID 337 exited
fstime completed
---
Minix: PID 334 exited
Minix: PID 331 exited
# ./workload_mix4.sh
```

Figure 20: Executing workload_mix4.sh



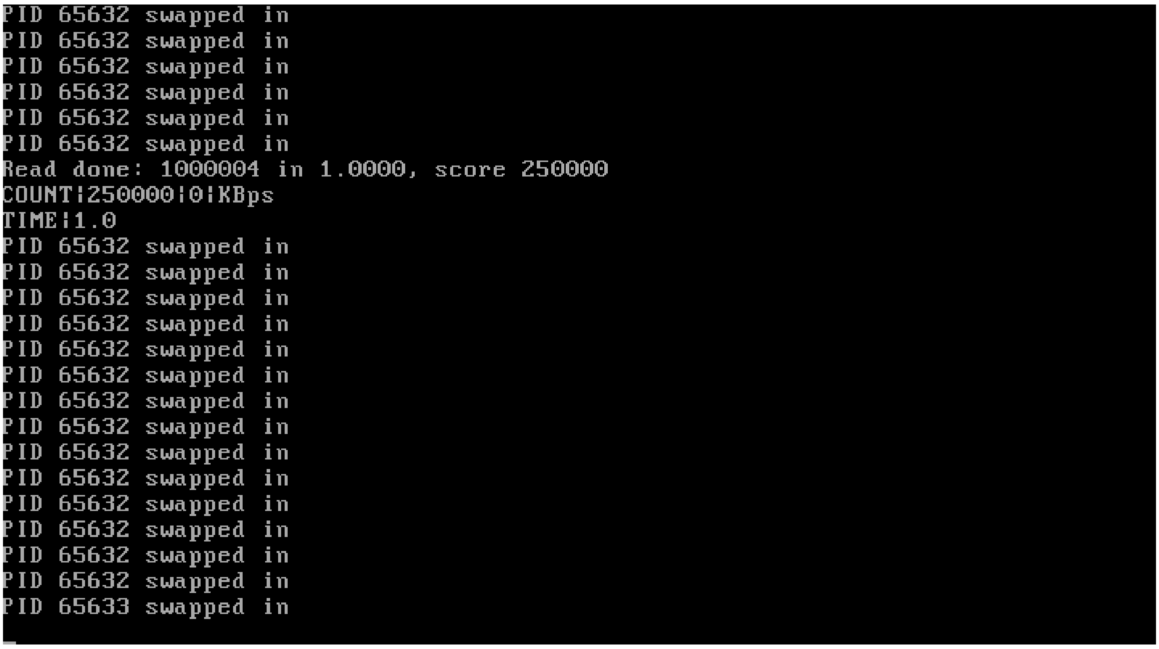
```
PID 65628 swapped in
Minix: PID 343 created
PID 65629 swapped in
Minix: PID 344 created
PID 65630 swapped in
Minix: PID 345 created
PID 65631 swapped in
Minix: PID 346 created
PID 65632 swapped in
Minix: PID 347 created
PID 65633 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
```

Figure 21: Running



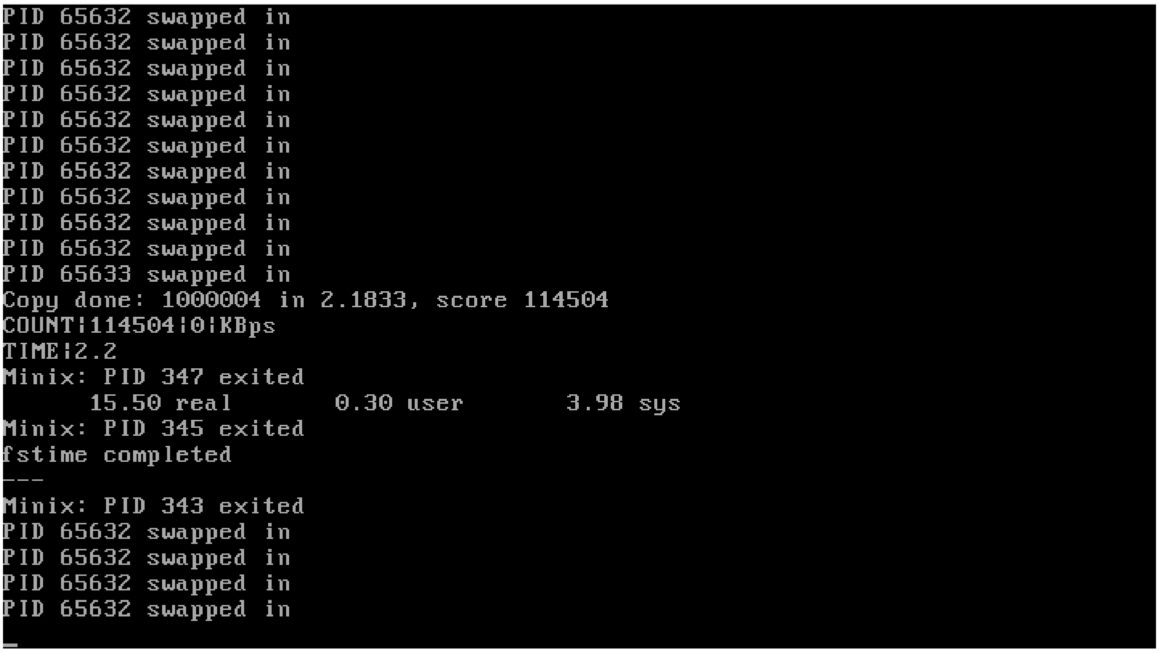
```
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
Write done: 1008000 in 1.1000, score 229090
COUNT¦229090¦0¦KBps
TIME¦1.1
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
```

Figure 22: Running



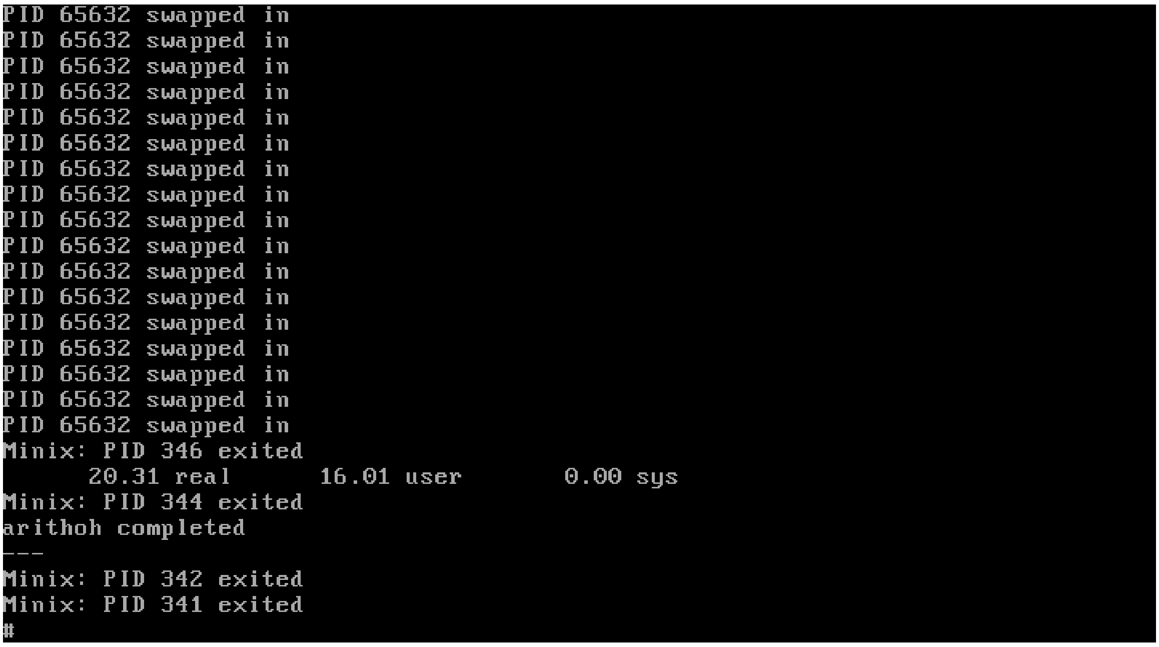
```
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
Read done: 1000004 in 1.0000, score 250000
COUNT¦250000¦0¦KBps
TIME¦1.0
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65633 swapped in
```

Figure 23: Running



```
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65633 swapped in
Copy done: 1000004 in 2.1833, score 114504
COUNT|114504|0|KBps
TIME|2.2
Minix: PID 347 exited
      15.50 real        0.30 user        3.98 sys
Minix: PID 345 exited
fstime completed
---
Minix: PID 343 exited
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
_
```

Figure 24: Running



```
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
PID 65632 swapped in
Minix: PID 346 exited
      20.31 real       16.01 user        0.00 sys
Minix: PID 344 exited
arithoh completed
---
Minix: PID 342 exited
Minix: PID 341 exited
#
```

Figure 25: Completed