

Operating Systems Lab

Assignment 3

Duddu Hriday

Part 1:

Code Modified:

```
if(rmp->priority >= USER_Q)
{
    printf("MINIX 210010016: PID %d swapped in\n",_ENDPOINT_P(rmp->endpoint));
}
```

Run.sh

```
$ run.sh
$ run.sh
1 cp schedule.c /usr/src/minix/servers/sched/schedule.c
2 cd /usr/src
3 make build MKUPDATE=YES
```

Part 2:

arithoh.sh

```
# ./arithoh.sh
Minix 210010016: PID 554 created
MINIX 210010016: PID 88 swapped in
Minix 210010016: PID 555 created
MINIX 210010016: PID 89 swapped in
Minix 210010016: PID 556 created
MINIX 210010016: PID 90 swapped in
MINIX 210010016: PID 90 swapped in
MINIX 210010016: PID 90 swapped in
```

```
MINIX 210010016: PID 90 swapped in
MINIX 210010016: PID 90 swapped in
MINIX 210010016: PID 90 swapped in
MINIX 210010016: PID 90 swapped in
MINIX 210010016: PID 90 swapped in
Minix 210010016: PID 556 exited
    9.83 real    9.83 user    0.00 sys
Minix 210010016: PID 555 exited
arithoh completed
---
Minix 210010016: PID 554 exited
```

The source code in order to understand the working of arithoh.sh is available in UnixBench/src/arith.c

- It measures the performance of arithmetic operations on a computer system
- It performs a series of arithmetic operations in a loop to measure the performance of the system.

pipe.sh

```
# ./pipe.sh
Minix 210010016: PID 560 created
MINIX 210010016: PID 95 swapped in
Minix 210010016: PID 561 created
MINIX 210010016: PID 96 swapped in
Minix 210010016: PID 562 created
MINIX 210010016: PID 97 swapped in
MINIX 210010016: PID 97 swapped in
MINIX 210010016: PID 97 swapped in
MINIX 210010016: PID 97 swapped in
Minix 210010016: PID 562 exited
      8.50 real      0.96 user      7.53 sys
Minix 210010016: PID 561 exited
pipe completed
---
Minix 210010016: PID 560 exited
```

The source code in order to understand the working of arithoh.sh is available in UnixBench/src/pipe.c

- The purpose of this code is to measure the throughput of a single process pipe without involving context switching.

syscall.sh

```
# ./syscall.sh
Minix 210010016: PID 566 created
MINIX 210010016: PID 101 swapped in
Minix 210010016: PID 567 created
MINIX 210010016: PID 102 swapped in
Minix 210010016: PID 568 created
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
MINIX 210010016: PID 103 swapped in
Minix 210010016: PID 568 exited
      6.25 real      2.18 user      4.06 sys
Minix 210010016: PID 567 exited
syscall completed
---
Minix 210010016: PID 566 exited
```

The source code in order to understand the working of arithoh.sh is available in
UnixBench/src/syscall.c

- The aim of this code is to measure the performance of system calls by repeatedly executing them in a loop.

fstime.sh

```
# ./fstime.sh
Minix 210010016: PID 571 created
MINIX 210010016: PID 106 swapped in
Minix 210010016: PID 572 created
MINIX 210010016: PID 107 swapped in
Minix 210010016: PID 573 created
MINIX 210010016: PID 108 swapped in
Write done: 1008000 in 1.1000, score 229090
COUNT:229090:0:KBps
TIME:1.1
Read done: 1000004 in 0.9833, score 254238
COUNT:254238:0:KBps
TIME:1.0
MINIX 210010016: PID 108 swapped in
Copy done: 1000004 in 2.2333, score 111940
COUNT:111940:0:KBps
TIME:2.2
Minix 210010016: PID 573 exited
      15.31 real      0.40 user      3.91 sys
Minix 210010016: PID 572 exited
fstime completed
---
Minix 210010016: PID 571 exited
```

The source code in order to understand the working of arithoh.sh is available in
UnixBench/src/fstime.c

- The aim of this code is to measure the file system performance by running tests related to reading writing and copying data

spawn.sh

```
MINIX 210010016: PID 10576 created
MINIX 210010016: PID 253 swapped in
Minix 210010016: PID 10576 exited
Minix 210010016: PID 10577 created
MINIX 210010016: PID 254 swapped in
Minix 210010016: PID 10577 exited
Minix 210010016: PID 10578 created
MINIX 210010016: PID 255 swapped in
Minix 210010016: PID 10578 exited
Minix 210010016: PID 10579 created
MINIX 210010016: PID 7 swapped in
Minix 210010016: PID 10579 exited
Minix 210010016: PID 10580 created
MINIX 210010016: PID 11 swapped in
Minix 210010016: PID 10580 exited
Minix 210010016: PID 10581 created
MINIX 210010016: PID 15 swapped in
Minix 210010016: PID 10581 exited
Minix 210010016: PID 580 exited
      17.81 real      0.00 user      16.86 sys
Minix 210010016: PID 579 exited
spawn completed
---
Minix 210010016: PID 578 exited
```

The source code in order to understand the working of arithoh.sh is available in UnixBench/src/spawn.c

- This code measures the performance of creating and immediately termination child process in a loop

workload_mix1.sh

```
$ workload_mix1.sh X
$ workload_mix1.sh
1  #!/bin/sh
2  ./arithoh.sh &
3  ./fstime.sh &
4  wait
5

Minix 210010016:PID 225 swapped in
Minix 210010016:PID 225 swapped in
Read done: 1000004 in 0.9333, score 267858
COUNT:267858:0:KBps
TIME:0.9
Minix 210010016:PID 225 swapped in
    10.55 real      8.60 user      0.06 sys
arithoh completed
---
Minix 210010016:PID 226 swapped in
Minix 210010016:PID 226 swapped in
Copy done: 1000004 in 2.2167, score 112782
COUNT:112782:0:KBps
TIME:2.2
    15.38 real      0.36 user      3.73 sys
fstime completed
---
```

- By the analysis from code and the order in which PIDs are printed, we can say that arithoh is CPU Bound Process and fstime is I/O Bound Process.
- The arithoh contains arithmetic operations so it is CPU Bound Process. The fstime contains file operations so it is I/O Bound Process.

workload_mix2.sh

```
$ workload_mix2.sh
1  #!/bin/sh
2  ./fstime.sh &
3  ./syscall.sh &
4  wait
```

```
Minix 210010016:PID 238 swapped in
Minix 210010016:PID 238 swapped in
Write done: 1008000 in 0.8833, score 285283
COUNT:285283:0:KBps
TIME:0.9
Minix 210010016:PID 238 swapped in
5.95 real 1.46 user 3.58 sys
syscall completed
---
Read done: 1000004 in 0.8167, score 306123
COUNT:306123:0:KBps
TIME:0.8
Minix 210010016:PID 237 swapped in
Copy done: 1000004 in 1.9000, score 131579
COUNT:131579:0:KBps
TIME:1.9
14.68 real 0.25 user 3.38 sys
fstime completed
---
```

- We already know that fstime is a I/O Bound process.
- Now, on analysis of code and the order in which PIDs are printed, we can say syscall is a mix of CPU Bound Process and IO Bound Process.
- The mix test, getpid test and exec test make syscall more CPU Bound and close test make it an I/O Bound Process.

workload_mix3.sh

```
$ workload_mix3.sh
1  #!/bin/sh
2  ./arithoh.sh &
3  ./spawn.sh &
4  wait
```

```
Minix 210010016:PID 66 swapped in
Minix 210010016:PID 67 swapped in
Minix 210010016:PID 68 swapped in
Minix 210010016:PID 69 swapped in
7.40 real 0.00 user 6.98 sys
spawn completed
---
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
```

```
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
Minix 210010016:PID 51 swapped in
15.26 real 8.28 user 0.00 sys
arithoh completed
---
```

- We already know that arithoh is a CPU Bound Process.
- Now by the analysis of code and the order in which PIDs are printed , we can say that spawn.sh is a CPU Bound Process.
- The spawn.c deals with Process creation and termination in a loop making this a CPU Bound Process.

workload_mix4.sh

```
$ workload_mix4.sh
1  #!/bin/sh
2  ./arithoh.sh &
3  ./pipe.sh &
4  wait
```

```
Minix 210010016:PID 76 swapped in
Minix 210010016:PID 76 swapped in
Minix 210010016:PID 77 swapped in
7.96 real      0.65 user      6.88 sys
pipe completed
---
Minix 210010016:PID 76 swapped in
Minix 210010016:PID 76 swapped in
```

```
Minix 210010016:PID 76 swapped in
Minix 210010016:PID 76 swapped in
Minix 210010016:PID 76 swapped in
16.10 real    8.56 user      0.00 sys
arithoh completed
---
```

- We already know that arithoh is CPU Bound Process
- On analysis of code and the order in which PIDs are printed , we can say that pipe is I/O Bound Process.
- The pipe involves Inter-process communications through pipes which makes it I/O Bound process.

workload_mix5.sh

```
$ workload_mix5.sh
1  #!/bin/sh
2  ./fstime.sh &
3  ./pipe.sh &
4  wait
```

```
Minix 210010016:PID 85 swapped in
Minix 210010016:PID 9 swapped in
Write done: 1008000 in 0.9167, score 274909
COUNT:274909:0:KBps
TIME:0.9
Minix 210010016:PID 85 swapped in
Minix 210010016:PID 85 swapped in
8.53 real      0.70 user      6.91 sys
pipe completed
---
Read done: 1000004 in 0.8667, score 288462
COUNT:288462:0:KBps
TIME:0.9
Minix 210010016:PID 84 swapped in
Copy done: 1000004 in 1.9000, score 131579
COUNT:131579:0:KBps
TIME:1.9
14.75 real      0.51 user      3.18 sys
fstime completed
---
```

Process	Type
fstime	I/O Bound
arithoh	CPU Bound
syscall	CPU Bound
spawn	CPU Bound
pipe	I/O Bound