

1 Latency test

1.1 Supported scheduling policies

1.2 Cyclicttest

Cyclicttest is a programme included in the `rt_test` suite to measure the latency of an particular environnement. It creates a defined number of threads that it wakes up at set intervals and then measures the latency between the moment the thread was expected to wake-up and the actual time the action took place.

Reading the `-h` guide we can learn that `-t` allows us to set the number of threads we wish to create and `-p` sets the priority.

threads -t More precisely if no parameter is give the number of threads will be equal to the number of cpus and the execution of thows threads will be ballanced equalay¹ between they s different cpu's. As we are currently on a 4 cpu system if we wish to create 4 threads leaving it to the default option would be correct. But if we didn't specify the `-t` option then only 1 thread would have been created.

Listing 1: cyclicttest -h

```
-t          --threads          one thread per available processor
-t [NUM]   --threads=NUM      number of threads:
without NUM, threads = max_cpus
without -t default = 1
```

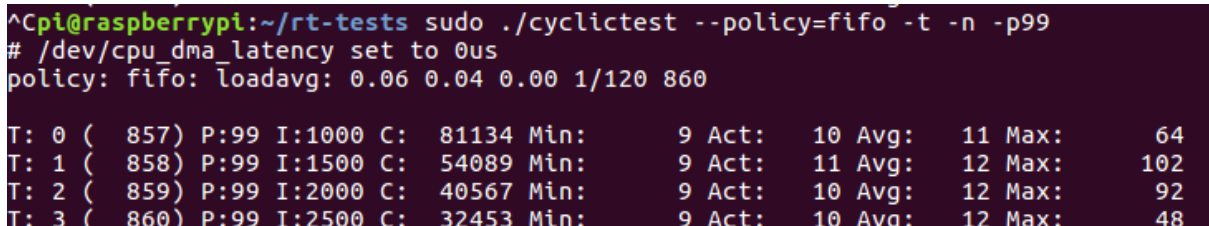
priorities -p In linux priority ranges are fixed by the scheduling algorithme used. by default cyclicttest runs `SCHED_FIFO`² and it's priorities will range from 0 to 99. Here priorities are reversed with 99 beeing the highest priority and 0 the lowest.

Listing 2: cyclicttest -h

```
-p PRIO    --prio=PRIO        priority of highest prio thread
```

1.3 Test

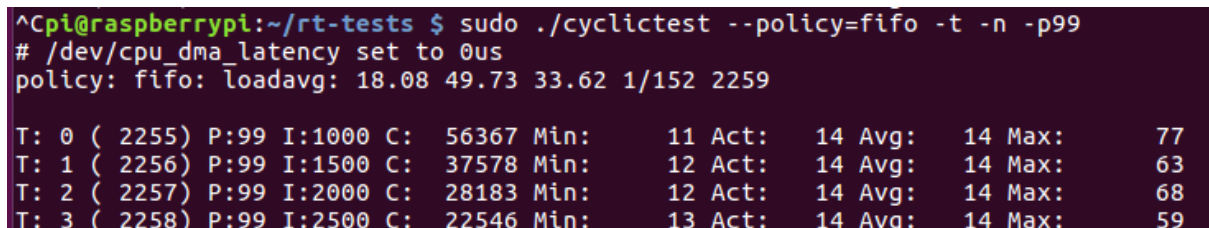
SHED_FIFO *Missing shed_other !!*



```
^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=fifo -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.06 0.04 0.00 1/120 860

T: 0 ( 857) P:99 I:1000 C: 81134 Min:      9 Act:   10 Avg:   11 Max:      64
T: 1 ( 858) P:99 I:1500 C: 54089 Min:      9 Act:   11 Avg:   12 Max:     102
T: 2 ( 859) P:99 I:2000 C: 40567 Min:      9 Act:   10 Avg:   12 Max:      92
T: 3 ( 860) P:99 I:2500 C: 32453 Min:      9 Act:   10 Avg:   12 Max:      48
```

Figure 1: SHED_FIFO voluntary



```
^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=fifo -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 18.08 49.73 33.62 1/152 2259

T: 0 ( 2255) P:99 I:1000 C: 56367 Min:     11 Act:   14 Avg:   14 Max:      77
T: 1 ( 2256) P:99 I:1500 C: 37578 Min:     12 Act:   14 Avg:   14 Max:      63
T: 2 ( 2257) P:99 I:2000 C: 28183 Min:     12 Act:   14 Avg:   14 Max:      68
T: 3 ( 2258) P:99 I:2500 C: 22546 Min:     13 Act:   14 Avg:   14 Max:      59
```

Figure 2: SHED_FIFO preempt-rl

¹under the best possible conditions

²See patch for fix <https://www.spinics.net/lists/linux-rt-users/msg05449.html>

```

^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=rr -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: rr: loadavg: 0.06 0.03 0.00 1/120 889

T: 0 ( 870) P:99 I:1000 C: 105636 Min:      9 Act:   10 Avg:   12 Max:   103
T: 1 ( 871) P:99 I:1500 C:  70424 Min:      9 Act:   14 Avg:   13 Max:    84
T: 2 ( 872) P:99 I:2000 C:  52818 Min:      9 Act:   10 Avg:   12 Max:    80
T: 3 ( 873) P:99 I:2500 C:  42254 Min:     10 Act:   13 Avg:   12 Max:    70

```

Figure 3: SHED_RR voluntary

```

^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=rr -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: rr: loadavg: 8.01 42.11 31.87 1/153 2272

T: 0 ( 2268) P:99 I:1000 C:  43078 Min:     11 Act:   34 Avg:   14 Max:    65
T: 1 ( 2269) P:99 I:1500 C:  28719 Min:     12 Act:   14 Avg:   14 Max:    56
T: 2 ( 2270) P:99 I:2000 C:  21539 Min:     12 Act:   15 Avg:   14 Max:    66
T: 3 ( 2271) P:99 I:2500 C:  17231 Min:     12 Act:   16 Avg:   14 Max:    61

```

Figure 4: SHED_RR preempt-rl

SHED_RR

1.4 Cyclicttest + Hackbench

1.4.1 Hackbench under SHED_OTHER and priority 20

```

pi@raspberrypi:~/rt-tests $ ./hackbench -l 1000000
Running in process mode with 10 groups using 40 file descriptors each (== 400 tasks)
Each sender will pass 1000000 messages of 100 bytes

```

Figure 5: Hackbench under SHED_OTHER and priority 20

```

^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=fifo -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 72.10 34.41 13.61 50/521 2118

T: 0 ( 2114) P:99 I:1000 C:  66022 Min:      7 Act:   11 Avg:   13 Max:   192
T: 1 ( 2115) P:99 I:1500 C:  44015 Min:      9 Act:   12 Avg:   16 Max:   173
T: 2 ( 2116) P:99 I:2000 C:  33011 Min:      9 Act:   20 Avg:   18 Max:   182
T: 3 ( 2117) P:99 I:2500 C:  26409 Min:      8 Act:   17 Avg:   17 Max:   119

```

Figure 6: Voluntary fifo with hackbench

```

^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=rr -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: rr: loadavg: 82.19 43.52 17.92 108/521 2329

T: 0 ( 2127) P:99 I:1000 C:  43577 Min:      7 Act:   11 Avg:   14 Max:   155
T: 1 ( 2128) P:99 I:1500 C:  29051 Min:      8 Act:   14 Avg:   16 Max:   175
T: 2 ( 2129) P:99 I:2000 C:  21788 Min:      9 Act:   23 Avg:   18 Max:   124
T: 3 ( 2130) P:99 I:2500 C:  17430 Min:     10 Act:   68 Avg:   28 Max:   223

```

Figure 7: Voluntary rr with hackbench

Voluntary

```

^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=fifo -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 72.10 34.41 13.61 50/521 2118

T: 0 ( 2114) P:99 I:1000 C: 66022 Min:      7 Act:   11 Avg:   13 Max:   192
T: 1 ( 2115) P:99 I:1500 C: 44015 Min:      9 Act:   12 Avg:   16 Max:   173
T: 2 ( 2116) P:99 I:2000 C: 33011 Min:      9 Act:   20 Avg:   18 Max:   182
T: 3 ( 2117) P:99 I:2500 C: 26409 Min:      8 Act:   17 Avg:   17 Max:   119

```

Figure 8: Voluntary fifo with hackbench

```

^Cpi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=rr -t -n -p99
# /dev/cpu_dma_latency set to 0us
policy: rr: loadavg: 82.19 43.52 17.92 108/521 2329

T: 0 ( 2127) P:99 I:1000 C: 43577 Min:      7 Act:   11 Avg:   14 Max:   155
T: 1 ( 2128) P:99 I:1500 C: 29051 Min:      8 Act:   14 Avg:   16 Max:   175
T: 2 ( 2129) P:99 I:2000 C: 21788 Min:      9 Act:   23 Avg:   18 Max:   124
T: 3 ( 2130) P:99 I:2500 C: 17430 Min:     10 Act:   68 Avg:   28 Max:   223

```

Figure 9: Voluntary rr with hackbench

Preempt-rt *Missing tests on preempt-rt kernel*

1.4.2 Hackbench under RT scheduling policies and priority 49

```

pi@raspberrypi:~/rt-tests $ sudo chrt --fifo 49 ./hackbench -l 1000000 -g 1q
Running in process mode with 1 groups using 40 file descriptors each (== 40 task
s)
Each sender will pass 1000000 messages of 100 bytes

```

Figure 10: Hackbench with shed.fifo and priority 49

```

pi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=fifo -t -n p99
defaulting realtime priority to 5
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 32.51 22.96 10.55 12/192 900

T: 0 ( 896) P: 5 I:1000 C: 53641 Min:      6 Act:  102 Avg:  638 Max:  696633
^CT: 1 ( 897) P: 5 I:1500 C: 39562 Min:      7 Act:  552 Avg:  684 Max:  6957
0: 2 ( 898) P: 5 I:2000 C: 30806 Min:      8 Act:   33 Avg:  730 Max:  61437
T: 2 ( 898) P: 5 I:2000 C: 30896 Min:      8 Act:   59 Avg:  730 Max:  61437
T: 3 ( 899) P: 5 I:2500 C: 25055 Min:      6 Act: 1030 Avg:  819 Max:  69314

```

Figure 11: Preempt-rt with fifo

```

pi@raspberrypi:~/rt-tests $ sudo ./cyclicttest --policy=fifo -t -n p99
defaulting realtime priority to 5
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 32.51 22.96 10.55 12/192 900

T: 0 ( 896) P: 5 I:1000 C: 53641 Min:      6 Act:  102 Avg:  638 Max:  696633
^CT: 1 ( 897) P: 5 I:1500 C: 39562 Min:      7 Act:  552 Avg:  684 Max:  6957
0: 2 ( 898) P: 5 I:2000 C: 30806 Min:      8 Act:   33 Avg:  730 Max:  61437
T: 2 ( 898) P: 5 I:2000 C: 30896 Min:      8 Act:   59 Avg:  730 Max:  61437
T: 3 ( 899) P: 5 I:2500 C: 25055 Min:      6 Act: 1030 Avg:  819 Max:  69314

```

Figure 12: Preempt-rt with fifo

SHED_FIFO

2 ADA concurrent programs

2.1 Program 1

```
with Ada.Text_IO;
use Ada.Text_IO;
procedure Main is
--declearation of tasks
    task A;
    task B;
    task body A is
begin
loop
Put_Line("A");
Put_Line("B");
delay 1.0;
end loop;
end A;
    task body B is
begin
loop
Put_Line("C");
Put_Line("D");
delay 1.0;
end loop;
end B;
--creation of 2 instances of task
--main task
begin
null;
end Main;
```

Figure 13: Program 1

code

Observations The A,B,C,D are printed in order throwout the execution with there not beeing any apparent modification of that ordered sequence.

2.2 Program 2

code

```
with Ada.Text_IO;
use Ada.Text_IO;
with Ada.Real_Time;

with Ada.Calendar;

procedure Main2 is
--declearation of tasks
    task type A;
    task type B;
    task body A is
use Ada.Calendar;
begin
```

```

loop
delay 0.1 ;
Put_Line("A");
Put_Line("B");

end loop;
end A;
    task body B is
--declare usage
use Ada.Real_Time;
--variables of task
Wait_Time: Ada.Real_Time.Time;
Interval :constant Ada.Real_Time.Time_Span := Ada.Real_Time.Milliseconds (100);
begin
loop
Wait_Time := Clock+Interval;
delay until Wait_Time ;
Put_Line("C");
Put_Line("D");

end loop;
end B;
--creation of 2 instances of tasknatm

type A_ptr is access A;
type B_ptr is access B;

AA : A_ptr;
BB : B_ptr;
--main task
begin
AA := new A;
BB := new B;
end Main2;

```

Observations In this test we can observe that even though both threads are configured to sleep for the same 100ms time interval they eventually end up off sync. We can attribute this to the difference between the way time is treated in the *delay* and *delay until*.