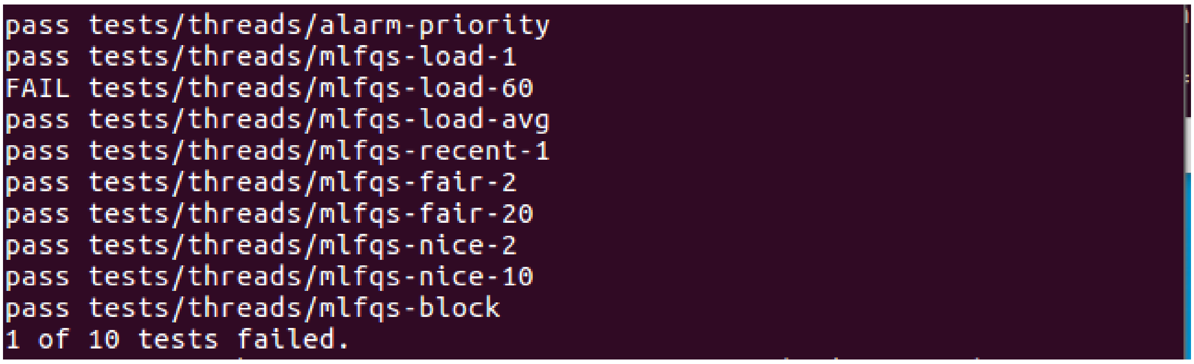
Task4 Report

卢斓 11810935

Start with GDB

To begin with, I use `make check` to run the test files in folder $threads$, the results are as follows:



```
pass tests/threads/alarm-priority
pass tests/threads/mlfqs-load-1
FAIL tests/threads/mlfqs-load-60
pass tests/threads/mlfqs-load-avg
pass tests/threads/mlfqs-recent-1
pass tests/threads/mlfqs-fair-2
pass tests/threads/mlfqs-fair-20
pass tests/threads/mlfqs-nice-2
pass tests/threads/mlfqs-nice-10
pass tests/threads/mlfqs-block
1 of 10 tests failed.
```

It can be seen that test $mlfqs-load-60$ fails.

Configuration

1. Change $GDBMACROS$ value in $pintos-GDB/src/utils/pintos-gdb$ to $../../misc/gdb-macros$.
2. In folder $src/threads/build$, start Pintos by the following command:

   `$ pintos -v --gdb -- -q -mlfqs run mlfqs-load-1`

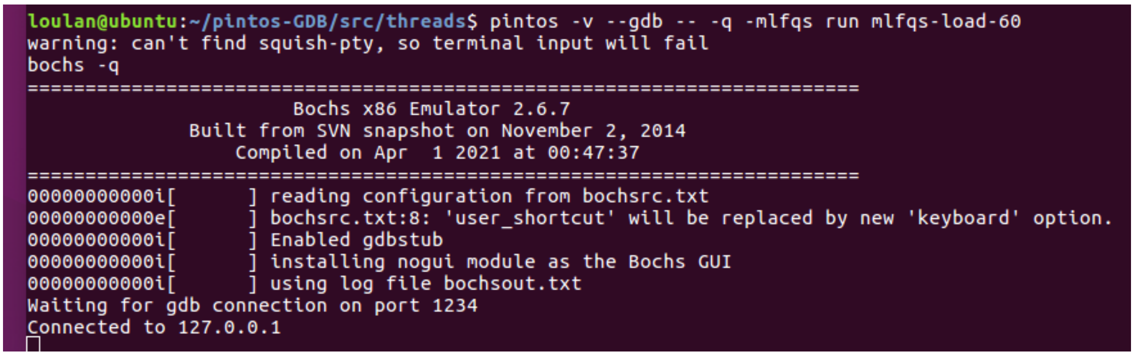
3. Open a second window on the same machine and start gdb:

   `$ pintos-gdb kernel.o`

4. Then tell gdb to attach to the waiting $Pintos$ emulator:

   `(gdb) debugpintos`

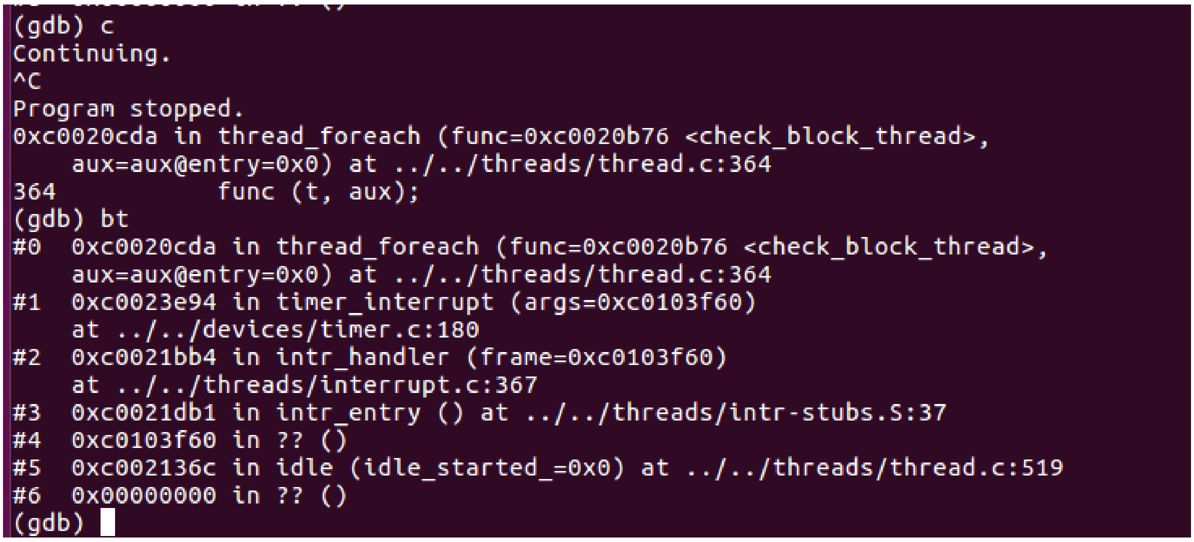
   After this, the result is as follows, indicating the connection is established:

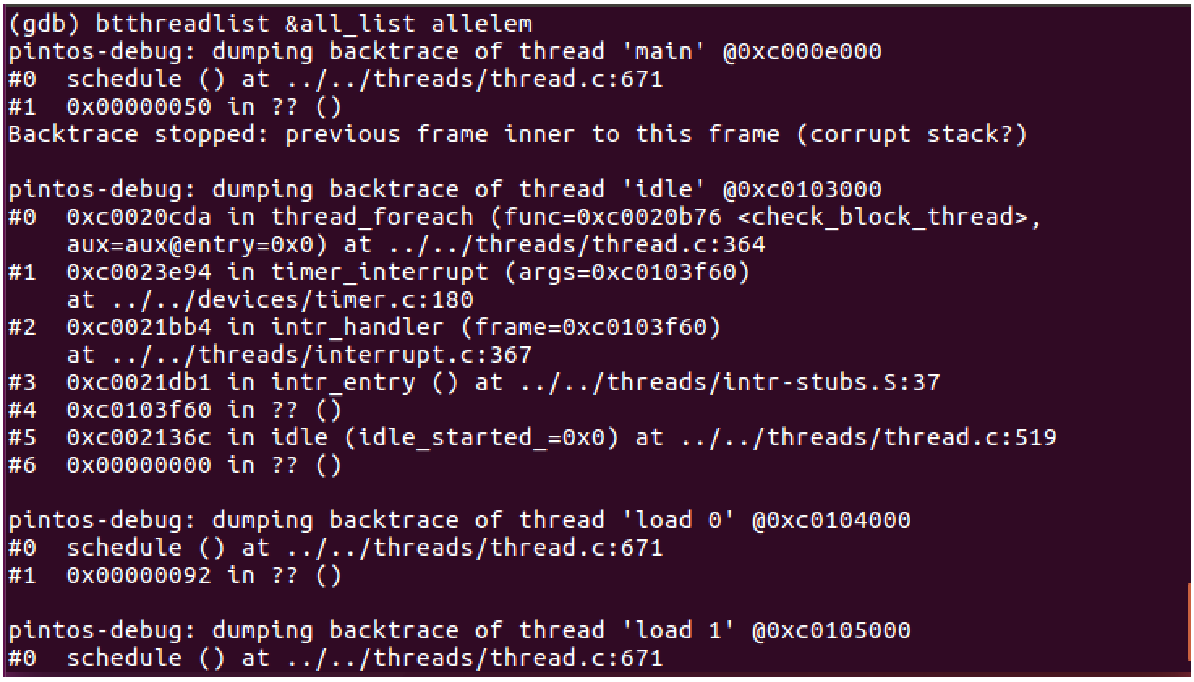
```
loulan@ubuntu:~/pintos-GDB/src/threads$ pintos -v --gdb -- -q -mlfqs run mlfqs-load-60
warning: can't find squish-pty, so terminal input will fail
bochs -q
========================================================================
                       Bochs x86 Emulator 2.6.7
              Built from SVN snapshot on November 2, 2014
                  Compiled on Apr  1 2021 at 00:47:37
========================================================================
00000000000i[      ] reading configuration from bochsrc.txt
00000000000e[      ] bochsrc.txt:8: 'user_shortcut' will be replaced by new 'keyboard' option.
00000000000i[      ] Enabled gdbstub
00000000000i[      ] installing nogui module as the Bochs GUI
00000000000i[      ] using log file bochsout.txt
Waiting for gdb connection on port 1234
Connected to 127.0.0.1
```

Debug with GDB

After that, I tell Pintos to run by executing `c`. Besides, some other useful commands like `ctrl+c`, `bt`, `l *address` and `btthreadlist &all_list allelem` are used for debugging and analyzing.



```
(gdb) c
Continuing.
^C
Program stopped.
0xc0020cda in thread_foreach (func=0xc0020b76 <check_block_thread>,
    aux=aux@entry=0x0) at ../../threads/thread.c:364
364             func (t, aux);
(gdb) bt
#0  0xc0020cda in thread_foreach (func=0xc0020b76 <check_block_thread>,
    aux=aux@entry=0x0) at ../../threads/thread.c:364
#1  0xc0023e94 in timer_interrupt (args=0xc0103f60)
    at ../../devices/timer.c:180
#2  0xc0021bb4 in intr_handler (frame=0xc0103f60)
    at ../../threads/interrupt.c:367
#3  0xc0021db1 in intr_entry () at ../../threads/intr-stubs.S:37
#4  0xc0103f60 in ?? ()
#5  0xc002136c in idle (idle_started_=0x0) at ../../threads/thread.c:519
#6  0x00000000 in ?? ()
(gdb)
```



```
(gdb) btthreadlist &all_list allelem
pintos-debug: dumping backtrace of thread 'main' @0xc000e000
#0  schedule () at ../../threads/thread.c:671
#1  0x00000050 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)

pintos-debug: dumping backtrace of thread 'idle' @0xc0103000
#0  0xc0020cda in thread_foreach (func=0xc0020b76 <check_block_thread>,
    aux=aux@entry=0x0) at ../../threads/thread.c:364
#1  0xc0023e94 in timer_interrupt (args=0xc0103f60)
    at ../../devices/timer.c:180
#2  0xc0021bb4 in intr_handler (frame=0xc0103f60)
    at ../../threads/interrupt.c:367
#3  0xc0021db1 in intr_entry () at ../../threads/intr-stubs.S:37
#4  0xc0103f60 in ?? ()
#5  0xc002136c in idle (idle_started_=0x0) at ../../threads/thread.c:519
#6  0x00000000 in ?? ()

pintos-debug: dumping backtrace of thread 'load 0' @0xc0104000
#0  schedule () at ../../threads/thread.c:671
#1  0x00000092 in ?? ()

pintos-debug: dumping backtrace of thread 'load 1' @0xc0105000
#0  schedule () at ../../threads/thread.c:671
```

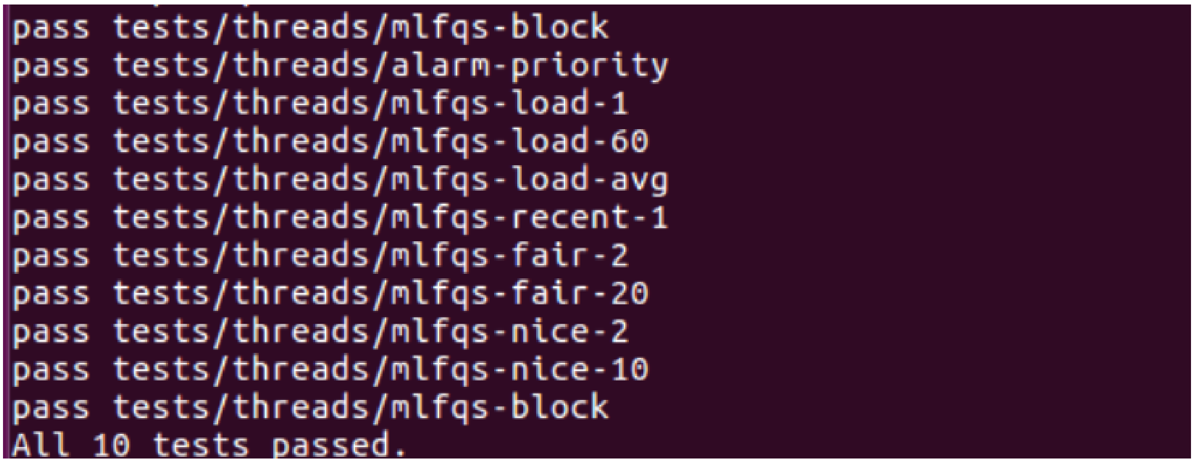
Through this, I find something is wrong with $load\_avg$. To find the exact place where the error occurs, I also examined variables related with $load\_avg$ like $recent\_cpu$. Finally, I find that the mistake is caused by the following false formulation:

```
t->recent_cpu = FP_ADD_MIX(FP_DIV( FP_MULT ( FP_MULT_MIX(load_average, 2), t->recent_cpu) , FP_ADD_MIX ( FP_MULT_MIX(load_average, 2), 1)) , t->nice);
```

And I modified it as:

```
fixed_t tmp = FP_DIV (FP_MULT_MIX (load_average, 2), FP_ADD_MIX (FP_MULT_MIX (load_average, 2), 1));
t->recent_cpu = FP_ADD (FP_MULT (tmp, t->recent_cpu), FP_CONST (t->nice));
```

Finally, all the tests can be passed.



```
pass tests/threads/mlfqs-block
pass tests/threads/alarm-priority
pass tests/threads/mlfqs-load-1
pass tests/threads/mlfqs-load-60
pass tests/threads/mlfqs-load-avg
pass tests/threads/mlfqs-recent-1
pass tests/threads/mlfqs-fair-2
pass tests/threads/mlfqs-fair-20
pass tests/threads/mlfqs-nice-2
pass tests/threads/mlfqs-nice-10
pass tests/threads/mlfqs-block
All 10 tests passed.
```