CS130 Project-0 Report

I. Installation and Testing

0. Local Environment

Ubuntu 18.04 LTS
gcc (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0
g++ (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0

1. Create workdir for this project

```
$ cd ~/桌面/
$ mkdir pintos && cd pintos
```

2. Download pintos source && Bochs emulator

- Clone pintos

  ```
  $ git clone git://pintos-os.org/pintos-anon pintos
  ```

- Download Bochs-2.2.6 into workdir
- After doing the following instructions, the directory is like this:

```
.
├── pintos
│ └── src
│ └── ... # other files and directories are omitted
├── bochs-2.2.6.tar.gz
```

3. Install pintos

- Run the installing script to install bochs

```
$ cd ./pintos/src/misc/
$ sudo env bochs-2.2.6-build.sh \
        SRCDIR=/home/wuty/桌面/pintos/ \
        PINTOSDIR=/home/wuty/桌面/pintos/pintos \
        DSTDIR=/usr/local/
```

Then, Bochs are installed.

- Fix pintos-gdb path && Make utils

```
$ cd ../utils/
$ vim ./pintos-gdb \
# Change LINE 4 to "GDBMACROS=/home/wuty/桌面/pintos/pintos/src/misc/gdb-
macros"
$ make
```

- Copy utils to `/usr/bin`

```
$ cd ˜/pintos/src/utils
$ sudo cp backtrace /usr/bin/
$ sudo cp pintos /usr/bin/
$ sudo cp pintos-gdb /usr/bin/
$ sudo cp pintos-mkdisk /usr/bin/
$ sudo cp Pintos.pm /usr/bin/
$ sudo cp squish-pty /usr/bin/
```

- Give permission to following utils

```
$ cd /usr/bin/
$ sudo chmod a+rx backtrace
$ sudo chmod a+rx pintos*
$ sudo chmod a+rx gdb-macros
$ sudo chmod a+rx Pintos.pm
$ sudo chmod a+rx /usr/bin/squish-pty
```

- Build Pintos

```
$ cd ~/桌面/pintos/pintos/src/threads/
$ make
```

The target is in directory `build`

- Test pintos-gdb & pintos, Run testcases

1. Test pintos-gdb & backtrace

```
$ pintos-gdb
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb)
```

As shown, it runs perfectly. No warnings like `gdb-macros not defined`

```
$ backtrace
backtrace: at least one argument required (use --help for help)
```

As shown, backtrace also works perfectly.

2. Run testcases

```
$ pintos -- run alarm-multiple
Prototype mismatch: sub main::SIGVTALRM () vs none at /usr/bin/pintos line
935.
Constant subroutine SIGVTALRM redefined at /usr/bin/pintos line 927.
squish-pty bochs -q
00000000000i[APIC?] local apic in  initializing
========================================================================
                       Bochs x86 Emulator 2.2.6
              Build from CVS snapshot on January 29, 2006
========================================================================
00000000000i[     ] reading configuration from bochsrc.txt
00000000000e[     ] user_shortcut: old-style syntax detected
00000000000i[     ] installing x module as the Bochs GUI
00000000000i[     ] using log file bochsout.txt
PiLo hda1
Loading...........
Kernel command line: run alarm-multiple
Pintos booting with 4,096 kB RAM...
383 pages available in kernel pool.
383 pages available in user pool.
Calibrating timer...  163,400 loops/s.
Boot complete.
Executing 'alarm-multiple':
(alarm-multiple) begin
(alarm-multiple) Creating 5 threads to sleep 7 times each.
(alarm-multiple) Thread 0 sleeps 10 ticks each time,
(alarm-multiple) thread 1 sleeps 20 ticks each time, and so on.
(alarm-multiple) If successful, product of iteration count and
```

```
(alarm-multiple) sleep duration will appear in nondescending order.
(alarm-multiple) thread 0: duration=10, iteration=1, product=10
(alarm-multiple) thread 0: duration=10, iteration=2, product=20
(alarm-multiple) thread 1: duration=20, iteration=1, product=20
(alarm-multiple) thread 2: duration=30, iteration=1, product=30
...
# It's too long, more information were ommited.
```

As shown, testcases run perfectly. No warnings like `missing squish-pty`

3. Testing pintos threads

```
$ make check
# Here is the result
pass tests/threads/alarm-single
pass tests/threads/alarm-multiple
pass tests/threads/alarm-simultaneous
FAIL tests/threads/alarm-priority
pass tests/threads/alarm-zero
pass tests/threads/alarm-negative
FAIL tests/threads/priority-change
FAIL tests/threads/priority-donate-one
FAIL tests/threads/priority-donate-multiple
FAIL tests/threads/priority-donate-multiple2
FAIL tests/threads/priority-donate-nest
FAIL tests/threads/priority-donate-sema
FAIL tests/threads/priority-donate-lower
FAIL tests/threads/priority-fifo
FAIL tests/threads/priority-preempt
FAIL tests/threads/priority-sema
FAIL tests/threads/priority-condvar
FAIL tests/threads/priority-donate-chain
FAIL tests/threads/mlfqs-load-1
FAIL tests/threads/mlfqs-load-60
FAIL tests/threads/mlfqs-load-avg
FAIL tests/threads/mlfqs-recent-1
pass tests/threads/mlfqs-fair-2
pass tests/threads/mlfqs-fair-20
FAIL tests/threads/mlfqs-nice-2
FAIL tests/threads/mlfqs-nice-10
FAIL tests/threads/mlfqs-block
20 of 27 tests failed.
../../tests/Make.tests:26: recipe for target 'check' failed
make[1]: *** [check] Error 1
make[1]: 离开目录“/home/wuty/桌面/pintos/pintos/src/threads/build”
../Makefile.kernel:10: recipe for target 'check' failed
make: *** [check] Error 2
```

7 testcases passed, it's perfect.

II.Problems

Because I've already installed pintos in this summer vacation, and finished some parts of projects. So, I didn't meet many problems. The only one problem was, when testing `mlfqs-load-1`, it took extremely long time (About one or two minutes). The infomation is shown.

```
FAIL tests/threads/mlfqs-load-1
Kernel panic in run: PANIC at ../../tests/threads/tests.c:93 in fail():
test failed
```

III.Result

I've installed pintos on local environmemt, and make tests.
It can run testcases perfectly, and its behaviours are as expected.