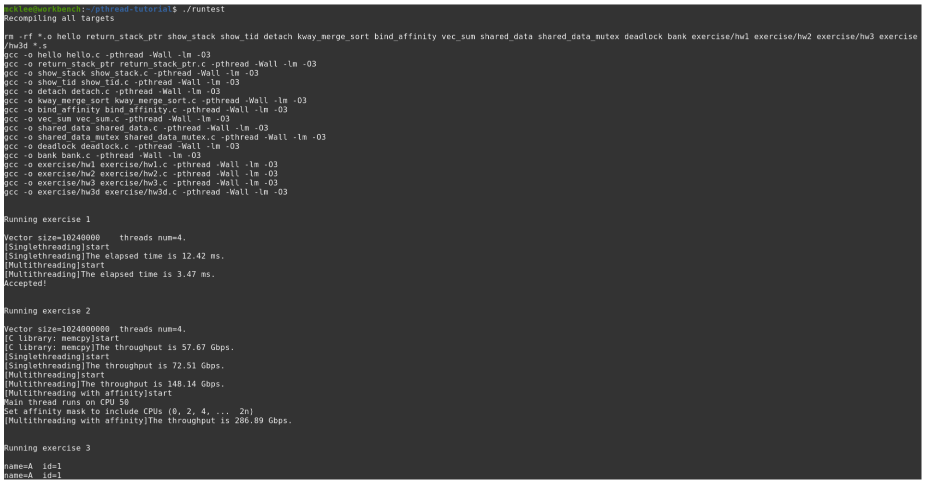
COMP3230 Tutorial 3 Report

Lee Chun Kok Michael
3035569110

Screenshots



```
mcklee@workbench:~/pthread-tutorial$ ./runtest
Recompiling all targets

rm -rf *.o hello return_stack_ptr show_stack show_tid detach kway_merge_sort bind_affinity vec_sum shared_data shared_data_mutex deadlock bank exercise/hw1 exercise/hw2 exercise/hw3 exercise
/hw3d *.s
gcc -o hello hello.c -pthread -Wall -lm -O3
gcc -o return_stack_ptr return_stack_ptr.c -pthread -Wall -lm -O3
gcc -o show_stack show_stack.c -pthread -Wall -lm -O3
gcc -o show_tid show_tid.c -pthread -Wall -lm -O3
gcc -o detach detach.c -pthread -Wall -lm -O3
gcc -o kway_merge_sort kway_merge_sort.c -pthread -Wall -lm -O3
gcc -o bind_affinity bind_affinity.c -pthread -Wall -lm -O3
gcc -o vec_sum vec_sum.c -pthread -Wall -lm -O3
gcc -o shared_data shared_data.c -pthread -Wall -lm -O3
gcc -o shared_data_mutex shared_data_mutex.c -pthread -Wall -lm -O3
gcc -o deadlock deadlock.c -pthread -Wall -lm -O3
gcc -o bank bank.c -pthread -Wall -lm -O3
gcc -o exercise/hw1 exercise/hw1.c -pthread -Wall -lm -O3
gcc -o exercise/hw2 exercise/hw2.c -pthread -Wall -lm -O3
gcc -o exercise/hw3 exercise/hw3.c -pthread -Wall -lm -O3
gcc -o exercise/hw3d exercise/hw3d.c -pthread -Wall -lm -O3


Running exercise 1

Vector size=10240000    threads num=4.
[Singlethreading]start
[Singlethreading]The elapsed time is 12.42 ms.
[Multithreading]start
[Multithreading]The elapsed time is 3.47 ms.
Accepted!


Running exercise 2

Vector size=1024000000  threads num=4.
[C library: memcpy]start
[C library: memcpy]The throughput is 57.67 Gbps.
[Singlethreading]start
[Singlethreading]The throughput is 72.51 Gbps.
[Multithreading]start
[Multithreading]The throughput is 148.14 Gbps.
[Multithreading with affinity]start
Main thread runs on CPU 50
Set affinity mask to include CPUs (0, 2, 4, ...  2n)
[Multithreading with affinity]The throughput is 286.89 Gbps.


Running exercise 3

name=A  id=1
name=A  id=1
```

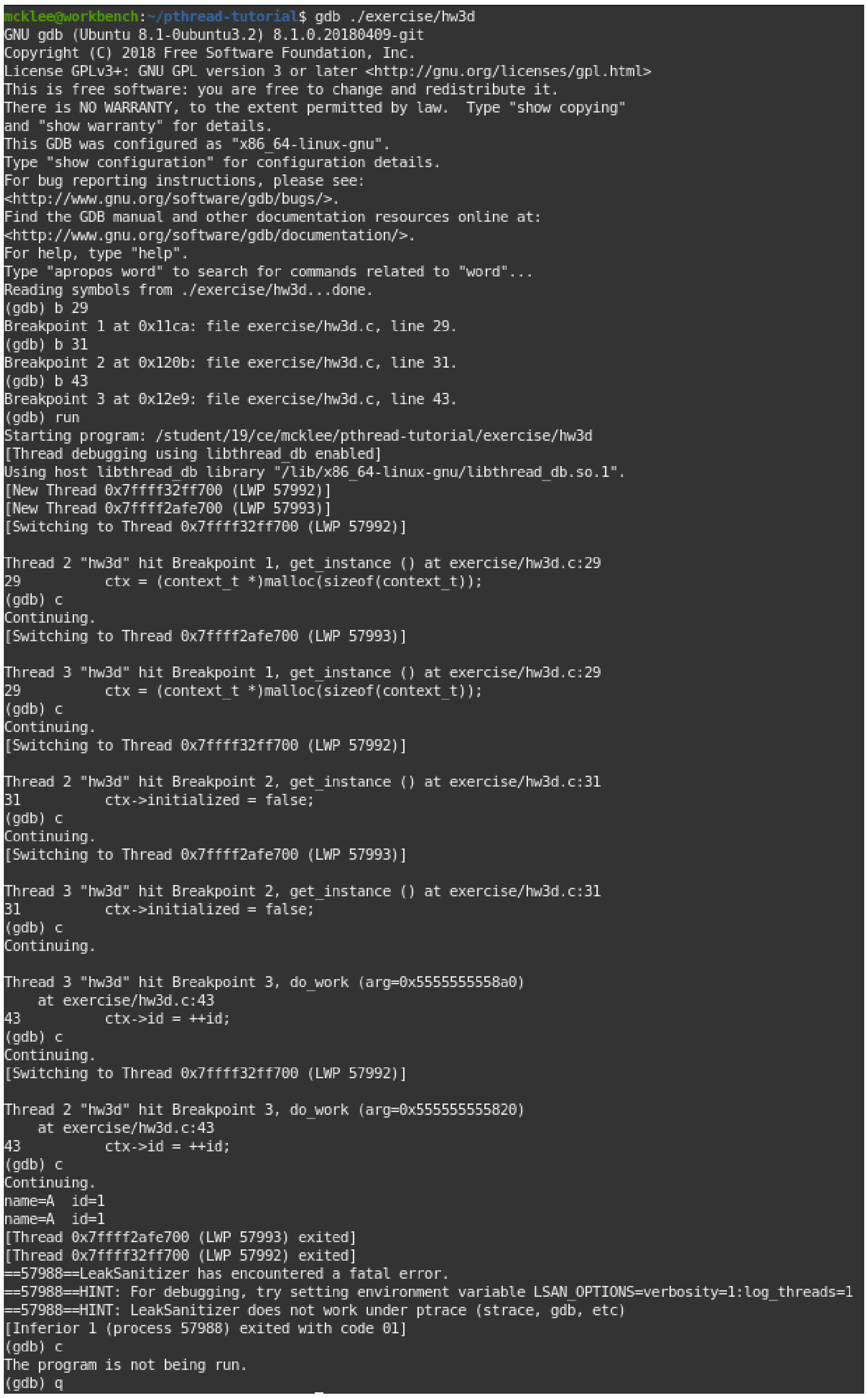
Figure 1: Execution

Homework 3 Explaination

1. The bug is caused by an race condition in the code. The lack of any mutex lock to guard the get__instance function causes the function to be invoked 2 times, which is shown by the fact that `get_instance()` breakpoint was triggered 2 times in the gdb.
2. An easy fix is to guard `get_instance()` with mutex lock at start and unlock at end. However, it would slow down the code since there would be overhead for 2 mutex lock and unlock. Therefore, I adapted the techique of double-checked locking which in most situations will only require locking of one time, hence reducing overhead. I used C11 atomic types to achieve a proper implementation.

Note

A online version of this code can be found at https://github.com/michaellee8/pthread-tutorial



```
mcklee@workbench:~/pthread-tutorial$ gdb ./exercise/hw3d
GNU gdb (Ubuntu 8.1-0ubuntu3.2) 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"...
Reading symbols from ./exercise/hw3d...done.
(gdb) b 29
Breakpoint 1 at 0x11ca: file exercise/hw3d.c, line 29.
(gdb) b 31
Breakpoint 2 at 0x120b: file exercise/hw3d.c, line 31.
(gdb) b 43
Breakpoint 3 at 0x12e9: file exercise/hw3d.c, line 43.
(gdb) run
Starting program: /student/19/ce/mcklee/pthread-tutorial/exercise/hw3d
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7ffff32ff700 (LWP 57992)]
[New Thread 0x7ffff2afe700 (LWP 57993)]
[Switching to Thread 0x7ffff32ff700 (LWP 57992)]

Thread 2 "hw3d" hit Breakpoint 1, get_instance () at exercise/hw3d.c:29
29          ctx = (context_t *)malloc(sizeof(context_t));
(gdb) c
Continuing.
[Switching to Thread 0x7ffff2afe700 (LWP 57993)]

Thread 3 "hw3d" hit Breakpoint 1, get_instance () at exercise/hw3d.c:29
29          ctx = (context_t *)malloc(sizeof(context_t));
(gdb) c
Continuing.
[Switching to Thread 0x7ffff32ff700 (LWP 57992)]

Thread 2 "hw3d" hit Breakpoint 2, get_instance () at exercise/hw3d.c:31
31          ctx->initialized = false;
(gdb) c
Continuing.
[Switching to Thread 0x7ffff2afe700 (LWP 57993)]

Thread 3 "hw3d" hit Breakpoint 2, get_instance () at exercise/hw3d.c:31
31          ctx->initialized = false;
(gdb) c
Continuing.

Thread 3 "hw3d" hit Breakpoint 3, do_work (arg=0x5555555558a0)
    at exercise/hw3d.c:43
43          ctx->id = ++id;
(gdb) c
Continuing.
[Switching to Thread 0x7ffff32ff700 (LWP 57992)]

Thread 2 "hw3d" hit Breakpoint 3, do_work (arg=0x555555555820)
    at exercise/hw3d.c:43
43          ctx->id = ++id;
(gdb) c
Continuing.
name=A  id=1
name=A  id=1
[Thread 0x7ffff2afe700 (LWP 57993) exited]
[Thread 0x7ffff32ff700 (LWP 57992) exited]
==57988==LeakSanitizer has encountered a fatal error.
==57988==HINT: For debugging, try setting environment variable LSAN_OPTIONS=verbosity=1:log_threads=1
==57988==HINT: LeakSanitizer does not work under ptrace (strace, gdb, etc)
[Inferior 1 (process 57988) exited with code 01]
(gdb) c
The program is not being run.
(gdb) q
```

Figure 2: Debug