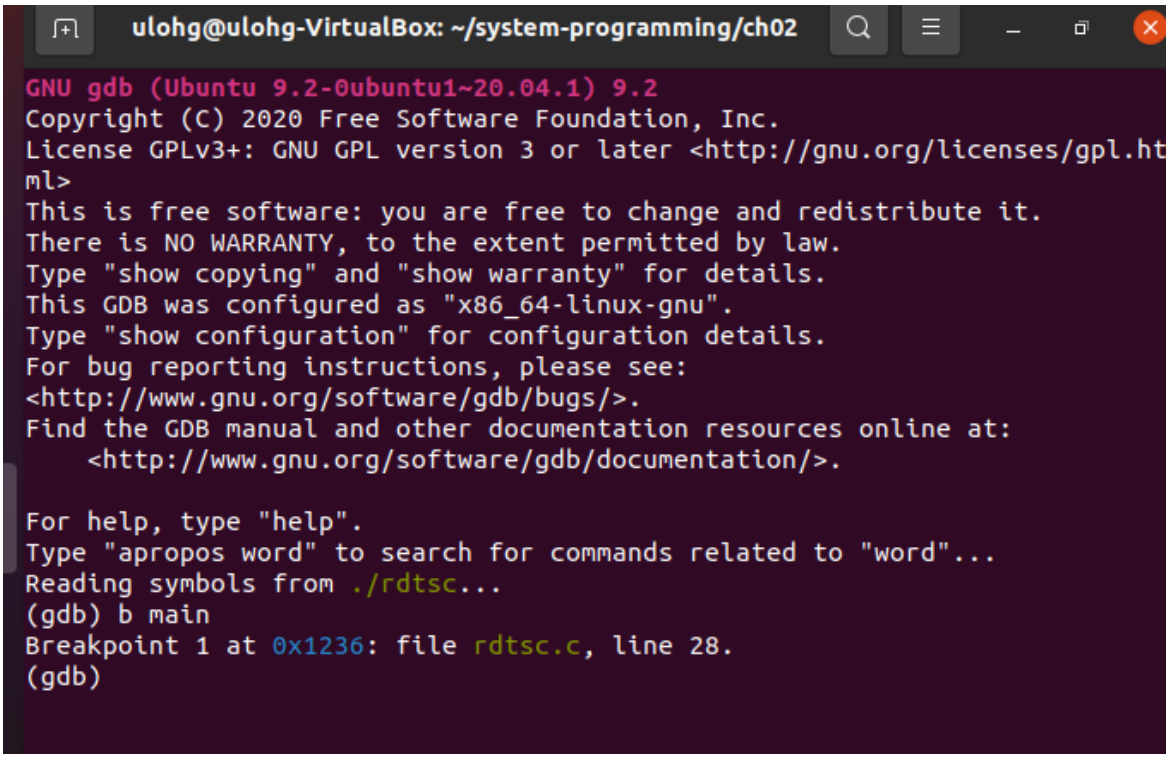
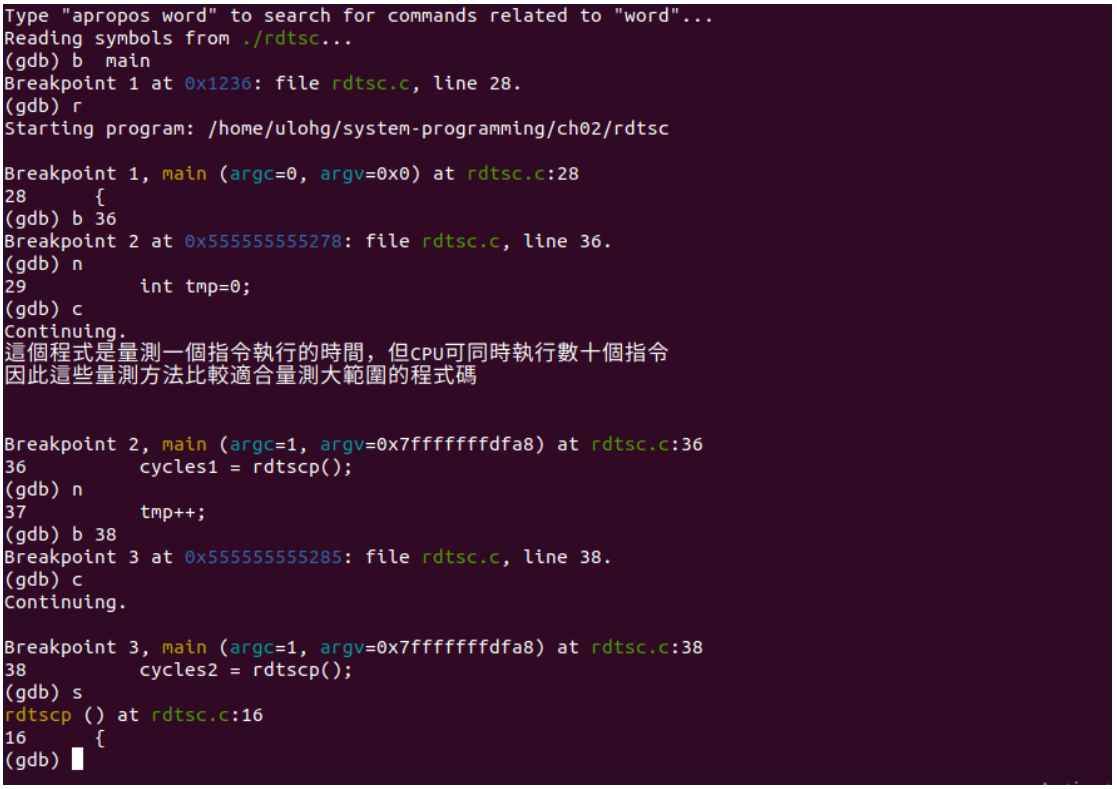
許家文 408530049

1. 設定中斷點，將中斷點設定在main函數



2. 單步執行，遇到函數不會進入
3. 單步執行，遇到函數會進入



4. 列印變數的值

```
Breakpoint 3, rdtscp () at rdtsc.c:38
38          cycles2 = rdtscp();
(gdb) s
main (argc=<optimized out>, argv=<optimized out>) at rdtsc.c:20
20          return ((uint64_t) lo) | (((uint64_t) hi) << 32);
(gdb) p tmp
$1 = 1
(gdb)
```

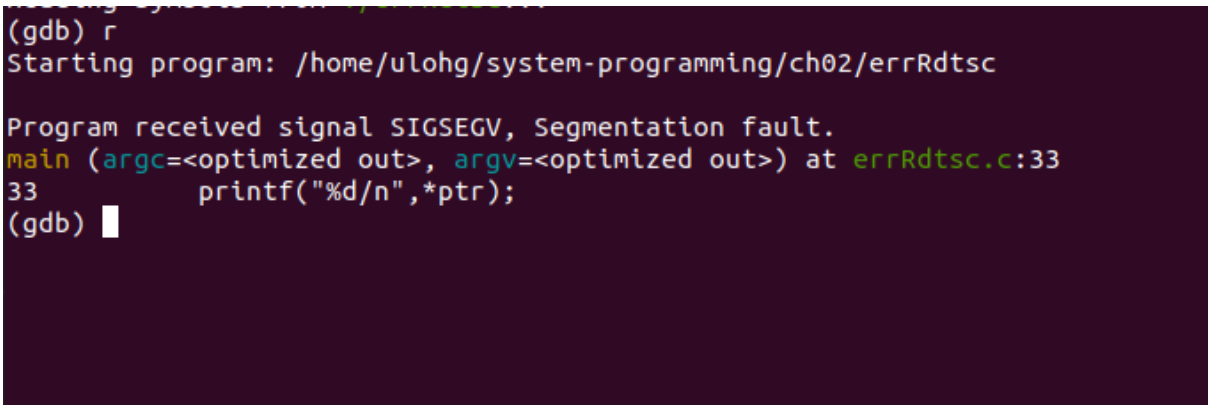
5. 使用bt、up、down，印出caller和callee各自的變數

```
Breakpoint 1, main (argc=1, argv=0x7fffffffdfa8) at rdtsc.c:36
36          cycles1 = rdtscp();
(gdb) s
rdtscp () at rdtsc.c:16
16      {
(gdb) n
19          __asm__ __volatile__("rdtscp":"=a"(lo), "=d"(hi));
(gdb) n
20          return ((uint64_t) lo) | (((uint64_t) hi) << 32);
(gdb) p lo
$1 = 1618793448
(gdb) bt
#0  rdtscp () at rdtsc.c:20
#1  0x000055555555527d in main (argc=1, argv=0x7fffffffdfa8) at rdtsc.c:36
(gdb) down
Bottom (innermost) frame selected; you cannot go down.
(gdb) up
#1  0x000055555555527d in main (argc=1, argv=0x7fffffffdfa8) at rdtsc.c:36
36          cycles1 = rdtscp();
(gdb) p lo
No symbol "lo" in current context.
(gdb) p tmp
$2 = 0
(gdb)
```

6. 使用watch查看變數被修改的情況

```
(gdb) b main
Breakpoint 1 at 0x1236: file rdtsc.c, line 28.
(gdb) r
Starting program: /home/ulohg/system-programming/ch02/a.out

Breakpoint 1, main (argc=0, argv=0x0) at rdtsc.c:28
28      {
(gdb) awatch tmp
Hardware access (read/write) watchpoint 2: tmp
(gdb) c
Continuing.

Hardware access (read/write) watchpoint 2: tmp

Value = 0
main (argc=1, argv=0x7fffffffdfa8) at rdtsc.c:33
33          printf("這個程式是量測一個指令執行的時間，但CPU可同時執行數十個指令\n");
(gdb) c
Continuing.
這個程式是量測一個指令執行的時間，但CPU可同時執行數十個指令
因此這些量測方法比較適合量測大範圍的程式碼


Hardware access (read/write) watchpoint 2: tmp

Old value = 0
New value = 1
main (argc=1, argv=0x7fffffffdfa8) at rdtsc.c:38
38          cycles2 = rdtscp();
(gdb)
```

7. 修改程式碼，故意存取錯誤的記憶體，看看會發生什麼事


(gdb) r
Starting program: /home/ulohg/system-programming/ch02/errRdtsc
Program received signal SIGSEGV, Segmentation fault.
main (argc=<optimized out>, argv=<optimized out>) at errRdtsc.c:33
33 printf("%d/n",*ptr);
(gdb)