1. 設定中斷點,將中斷點設定在main函數

```
ulohg@ulohg-VirtualBox: ~/system-programming/ch02
                                                            Q
                                                                            ♂
GNU gdb (Ubuntu 9.2-Oubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <a href="http://gnu.org/licenses/gpl.ht">http://gnu.org/licenses/gpl.ht</a>
ml>
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:
<a href="http://www.gnu.org/software/gdb/bugs/">http://www.gnu.org/software/gdb/bugs/>.</a>
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 ./rdtsc...
(gdb) b main
Breakpoint 1 at 0x1236: file rdtsc.c, line 28.
(ddb)
```

- 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

```
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./rdtsc...
(gdb) b main
Breakpoint 1 at 0x1236: file rdtsc.c, line 28.
(gdb) r
Starting program: /home/ulohg/system-programming/ch02/rdtsc
Breakpoint 1, main (argc=0, argv=0x0) at rdtsc.c:28
(gdb) b 36
Breakpoint 2 at 0x5555555555278: file rdtsc.c, line 36.
(gdb) n
(gdb) c
vs-v-v
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
Breakpoint 2, main (argc=1, argv=0x7ffffffffdfa8) at rdtsc.c:36
            cycles1 = rdtscp();
(gdb) n
(gdb) b 38
Breakpoint 3 at 0x5555555555285: file rdtsc.c, line 38.
(gdb) c
Continuing.
Breakpoint 3, main (argc=1, argv=0x7fffffffdfa8) at rdtsc.c:38
38 cycles2 = rdtscp();
(gdb) s
rdtscp () at rdtsc.c:16
(gdb)
```

4. 列印變數的值

5. 使用bt、up、down, 印出caller和callee各自的變數

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
<mark>main (argc=1, argv=</mark>0x7ffffffffffa8) 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
 nain (argc=1, argv=0x7ffffffffdfa8) 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

printf("%d/n",*ptr);
(gdb) ■
```