410410043 SP_hw1

在debug之前 先利用gcc -g -o 編譯成執行檔

```
bine0619@bine0619-virtual-machine:~/system-programming/ch02$ gcc -g rdtsc.c -o rdtsc
bine0619@bine0619-virtual-machine:~/system-programming/ch02$ ls
clock_gettime.c  clock_gettime_loop.c  makefile  memory_layout.c  pointer1.c  pointer.c  rdtsc  rdtsc.c  table.c
```

使用gdb開起要編譯的檔案
可以從底下的 Reading symbols 確認有讀到該檔案

```
bine0619@bine0619-virtual-machine:~/system-programming/ch02$ gdb ./rdtsc
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 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 ./rdtsc...
```

Q1.
使用 b main 將 breakpoint 設在 main() 的地方

```
(gdb) b main
Breakpoint 1 at 0x1236: file rdtsc.c, line 28.
(gdb) r
Starting program: /home/bine0619/system-programming/ch02/rdtsc

Breakpoint 1, main (argc=0, argv=0x0) at rdtsc.c:28
28      {
```

Q2, Q3.
在程式中的36及38行皆有rdtscp() 的函式
使用n即會直接繼續到下一行
使用s即會跑入該函式

```
(gdb) b 36
Breakpoint 2 at 0x555555555278: file rdtsc.c, line 36.
(gdb) n
29          int tmp=0;
(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


Breakpoint 2, main (argc=1, argv=0x7fffffffdf58) at rdtsc.c:36
36          cycles1 = rdtscp();
(gdb) n
37          tmp++;
(gdb) n
38          cycles2 = rdtscp();
(gdb) s
rdtscp () at rdtsc.c:16
```

Q4.
除錯過程中也可使用 p 印出特定變數的值

```
(gdb) p lo
$1 = 4151474149
```

Q5.
在進入到某函式內，可使用bt查看目前的堆疊
要回到上層的堆疊可以使用up
要到下層可以使用down

```
(gdb) bt
#0  rdtscp () at rdtsc.c:16
#1  0x000055555555528a in main (argc=1, argv=0x7fffffffdf58) at rdtsc.c:38
(gdb) up
#1  0x000055555555528a in main (argc=1, argv=0x7fffffffdf58) at rdtsc.c:38
38          cycles2 = rdtscp();
(gdb) p tmp
$2 = 1
(gdb) down
#0  rdtscp () at rdtsc.c:16
16      {
```

Q6.
使用watch觀測某變數的值
可以看到底下由0改成1

```
(gdb) watch tmp
Hardware watchpoint 2: tmp
(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 watchpoint 2: tmp

Old value = 0
New value = 1
main (argc=1, argv=0x7fffffffdf58) at rdtsc.c:41
41          cycles2 = rdtscp();
```

Q7.

先在程式中加入一個指標宣告和印出指標形成一個錯誤

```c
int main(int argc, char **argv)
{
    int tmp=0;
    uint64_t cycles1, cycles2;
    struct timespec ts1, ts2;
    int *p;

    printf("%d\n", *p);
```

重新編譯後執行除錯就會發生Segmentation fault

透過印出該值發現是沒有指定初始值

```
bine0619@bine0619-virtual-machine:~/system-programming/ch02$ gcc -g rdtsc.c -o rdtsc
bine0619@bine0619-virtual-machine:~/system-programming/ch02$ ls
clock_gettime.c  clock_gettime_loop.c  makefile  memory_layout.c  pointer1.c  pointer.c  rdtsc  rdtsc.c  table.c
bine0619@bine0619-virtual-machine:~/system-programming/ch02$ gdb ./rdtsc
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 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 ./rdtsc...
(gdb) b main
Breakpoint 1 at 0x1230: file rdtsc.c, line 28.
(gdb) r
Starting program: /home/bine0619/system-programming/ch02/rdtsc

Breakpoint 1, main (argc=0, argv=0x0) at rdtsc.c:28
28      {
(gdb) c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555264 in main (argc=1, argv=0x7fffffffdf58) at rdtsc.c:34
34          printf("%d\n", *p);
(gdb) bt
#0  0x0000555555555264 in main (argc=1, argv=0x7fffffffdf58) at rdtsc.c:34
(gdb) print p
$1 = (int *) 0xf0
```