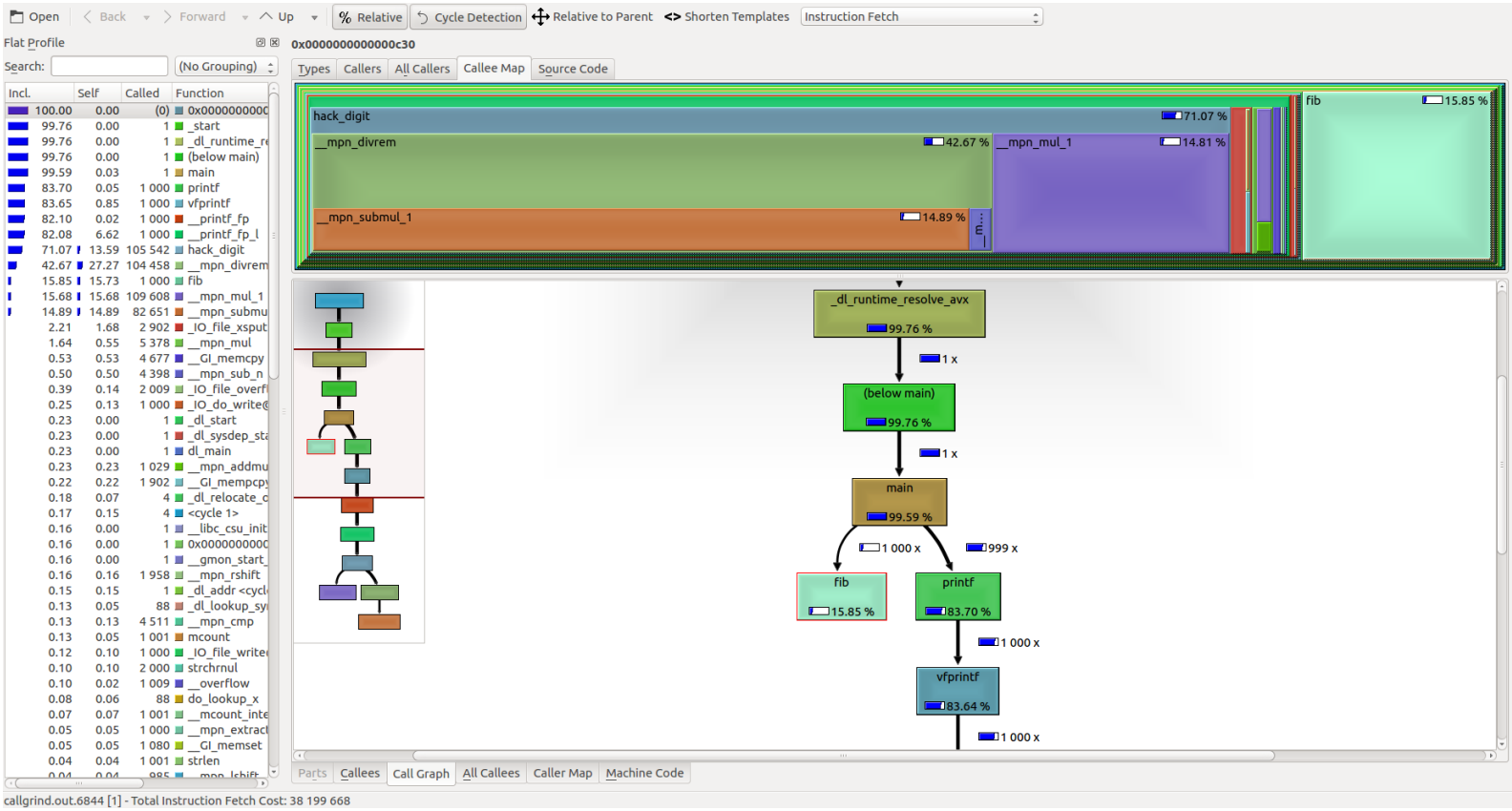
405410016　楊淨

費波納契程式

```
#include <stdio.h>

long double  fib(int n){
   long  double a=0,b=1,c=0;
    int i=0;
    if(n==0){

        return 0;
    }
    if(n==1){
        return 1;
    }
    for(i=2;i<=n;i++){
        c=a+b;
        a=b;
        b=c;

    }
    return c;
}

int main(){
    int i=0;
    for(i=0;i<1000;i++){
        printf("%.0LF\n",fib(i));
    }
     return 0;
}
```

由於 UBUNTU16.10 版本 gcc 5.4 的 GPROF 有 BUG 無法使用

所以改用 VALGRIND 與 PERF 來觀察



上圖可知 大部份的時間都花在 PRINTF 內

所以效能瓶頸就是卡在呼叫 PRINTF

我再使用 perf 來觀察 確認實際運算的情況 為了能夠重複觀察 我將程式改寫一下 增加了 WHILE（1）

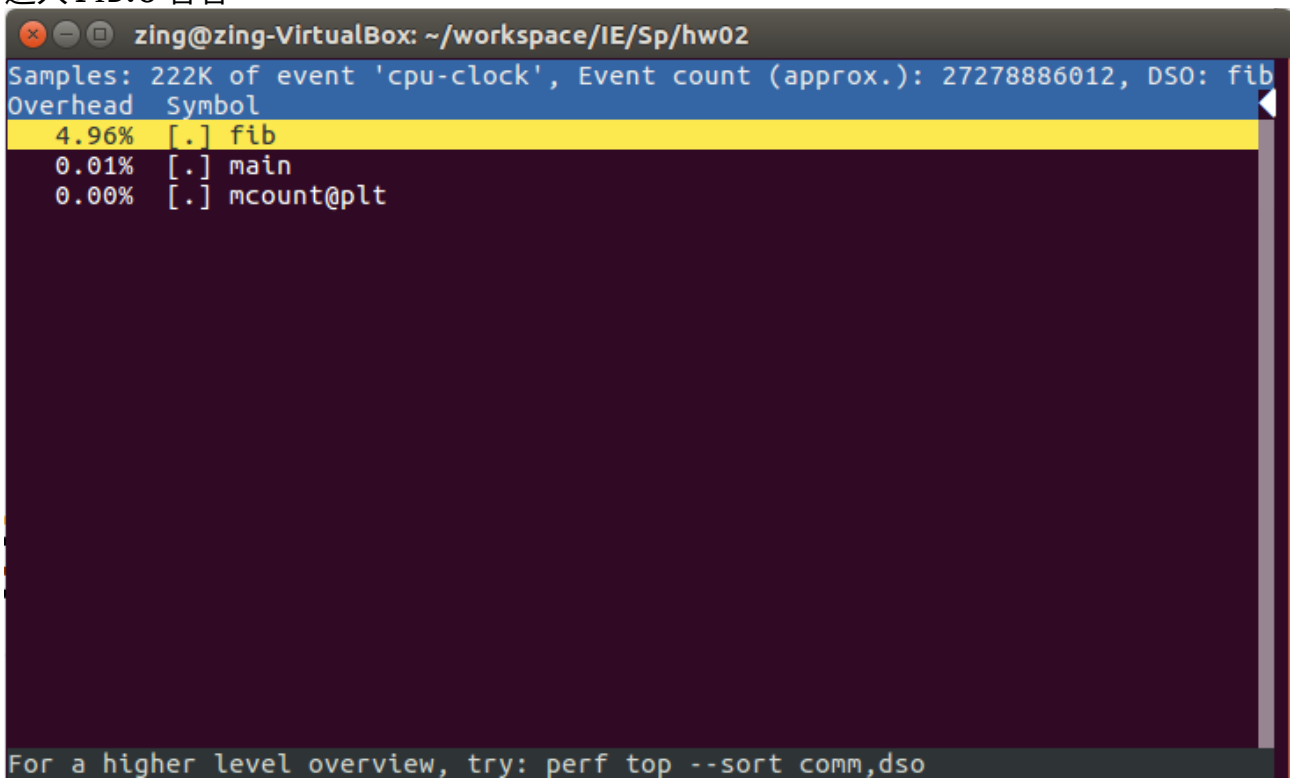
```
int main(){
    int i=0;
    while(1){
        for(i=0;i<1000;i++){
            printf("%.0LF\n",fib(i));
        }
    }
    return 0;
}
```

執行之後 使用 sudo perf top

```
zing@zing-VirtualBox: ~/workspace/IE/Sp/hw02
Samples: 433K of event 'cpu-clock', Event count (approx.): 27521519439
Overhead  Shared Object                     Symbol
   8.58%  libc-2.23.so                      [.] __mpn_divrem
   7.24%  [kernel]                          [k] exit_to_usermode_loop
   5.87%  fib.o                             [.] fib
   4.32%  [kernel]                          [k] queue_work_on
   4.07%  [kernel]                          [k] __lock_text_start
   3.48%  [kernel]                          [k] flush_work
   2.43%  perf-2218.map                     [.] 0x00007f9fc82ad7a9
   2.27%  [kernel]                          [k] finish_task_switch
   2.22%  [kernel]                          [k] copy_user_generic_string
   1.90%  libc-2.23.so                      [.] hack_digit
   1.71%  libc-2.23.so                      [.] __mpn_submul_1
   1.60%  [kernel]                          [k] entry_SYSCALL_64_fastpath
   1.54%  libc-2.23.so                      [.] __mpn_mul_1
   1.44%  [kernel]                          [k] process_one_work
   1.29%  libc-2.23.so                      [.] __memcpy_avx_unaligned
   0.78%  libc-2.23.so                      [.] __GI___printf_fp_l
   0.59%  libX11.so.6.3.0                   [.] _XSetImage
   0.56%  libglib-2.0.so.0.4800.2           [.] g_string_insert_unichar
   0.55%  libapt-pkg.so.5.0.0               [.] pkgDepCache::CheckDep
   0.49%  [kernel]                          [k] __fget
   0.48%  libX11.so.6.3.0                   [.] 0x0000000000026e82
Press '?' for help on key bindings
```

```
zing@zing-VirtualBox: ~/workspace/IE/Sp/hw02
1553852978984813852074059673161830198176669276323696942223312887344790313025911244481713351501165260376354956082694955466752075227689492739914362378502737924912029514120709064020367285713957343354300306514445626952141648594867355469117315946694391397976383143256399883518028228488915172593309678729331816202756170609917233767383041217209584943765174751692493742246809297354644133936122592884043564804209053322106010122500401299800747614364334989665120583256087528374787311722525072597915062970414810138794483422403916662181867293140582419694864798711643183754775211214959295924958512245763262657411831382377520563872504663920218177857494211690555081491320644972358513598073513105577053548828042146963479069260878015844314783682810396308144128318669606481492949312717001486374273888985049535851244885862522670304196110970935647651452221908555016878341984313879718508049193888818213862228023042748071927766676271706488228500683539503046290692395499525156182544686093811393870475056364766670718681881143200247265739330783575294850447439463894977368760276971994887368971544589193111464706081989858487431089080861293418972593880408494622606310212688856848793683428786095010233049561487336356031128070798360474379486142936480758100878895444419297490134344644676976773365861835995496238237654860218523840565112636675706632389482952471214058880350696948006265724862464134990611541871171163500192086919678794777985179285811488615593874065936631843948893692129084118332279746493314735525710942130168769507279343739149986947566515245323672678410018143826576760050133154293350400
```

進入 Fib.o 看看



我們可以看到 fib 這個子函式佔用了大多數的時間
而在 main 裡面的 printf 只佔了少少的時間
由此可知 真正的效能瓶頸在 fib

而 VALGRIND 為何會在 printf 上花費如此多資源
我猜測是因為他經過重製後他內建的函數呼叫比較麻煩一點
所以才會比 fib 還多
實際用 perf 來觀看的話確實 fib 才是效能瓶頸