

GDB(not essential currently)

GDB

```
yanzijian@madcow-OptiPlex-3050:~/Desktop$ gcc main.c
yanzijian@madcow-OptiPlex-3050:~/Desktop$ ls
a.out main.c
yanzijian@madcow-OptiPlex-3050:~/Desktop$ ./a.out
Hello World
yanzijian@madcow-OptiPlex-3050:~/Desktop$ gdb a.out
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 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 a.out...(no debugging symbols found)...done.
(gdb) b printf
Function "printf" not defined.
Make breakpoint pending on future shared library load? (y or [n]) q
Please answer y or [n].
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 1 (printf) pending.
(gdb) quit
yanzijian@madcow-OptiPlex-3050:~/Desktop$ gcc main.c --coverage
yanzijian@madcow-OptiPlex-3050:~/Desktop$ ls
a.out main.c main.gcno
yanzijian@madcow-OptiPlex-3050:~/Desktop$ gcov main.c
File 'main.c'
Lines executed:100.00% of 6
Creating 'main.c.gcov'

yanzijian@madcow-OptiPlex-3050:~/Desktop$ cat main.c.gcov
:- 0:Source:main.c
:- 0:Graph:main.gcno
:- 0:Data:main.gcda
:- 0:Runs:1
:- 0:Programs:1
:- 1:#include <stdio.h>
:- 2:
1: 3:int main()
:- 4:{ 
1: 5:    volatile int count =0;
11: 6:    for (int i = 0; i < 10; i++) {
10: 7:        printf("Hello World\n");
10: 8:        count += 2;
:- 9:    }
1: 10:    return 0;
:- 11:}

yanzijian@madcow-OptiPlex-3050:~/Desktop$ nm a.out
```

gcov

- Gcov is GCC Coverage
- 是一个测试代码覆盖率的工具
- 是一个命令行方式的控制台程序
- 伴随GCC发布，配合GCC共同实现对C/C++文件的语句覆盖和分支覆盖测试；
- 与程序概要分析工具(profiling tool, 例如gprof)一起工作，可以估计程序中哪一段代码最耗时；