LAB 2.2

王伟杰

Date: 2023-5-16

Contents

Running a Hello World Program in C using GCC

1 Overview

The lab helps familiarize you with writing a simple Hello World program using C, the GCC compiler link, and Pico(a text editor, link). It uses Ubuntu VM created in Lab 2.1.Here is lab objective:

1. Learn to run a program in gcc.
2. Learn to debug a program in gdb.

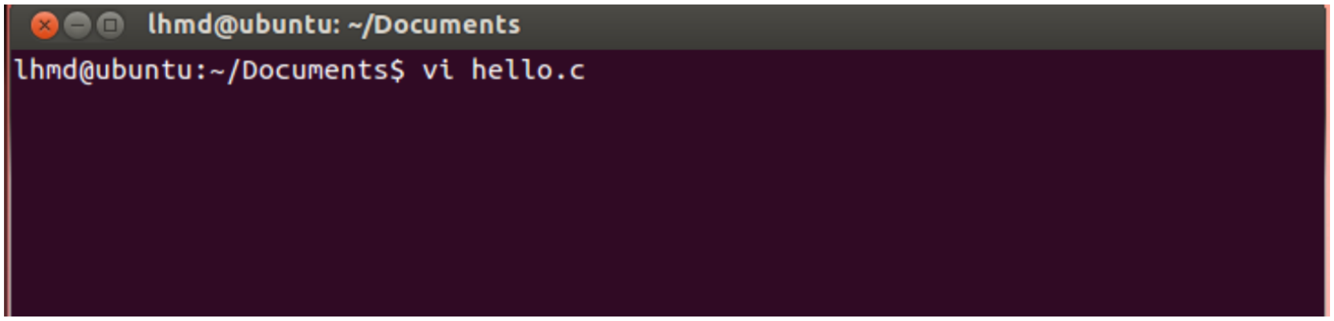
2 运行程序

2.1 打开终端

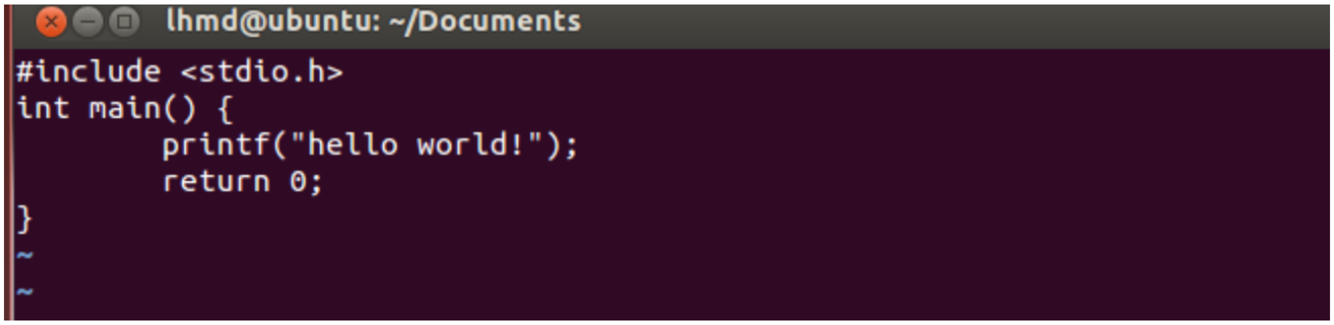
开启虚拟机，`Ctrl+Alt+T`打开终端

2.2 编写程序

编写一个`hello.c`文件



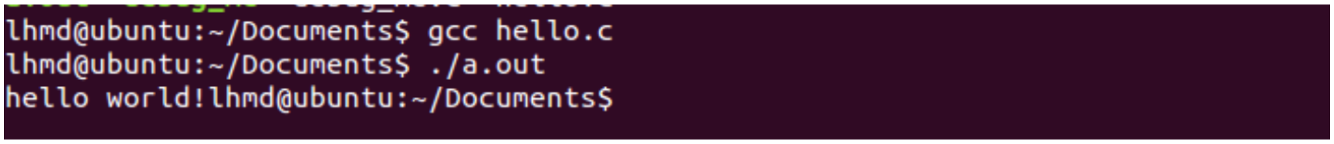
```
lhmd@ubuntu: ~/Documents
lhmd@ubuntu:~/Documents$ vi hello.c
```



```
lhmd@ubuntu: ~/Documents
#include <stdio.h>
int main() {
        printf("hello world!");
        return 0;
}
~
~
```

2.3 编译运行

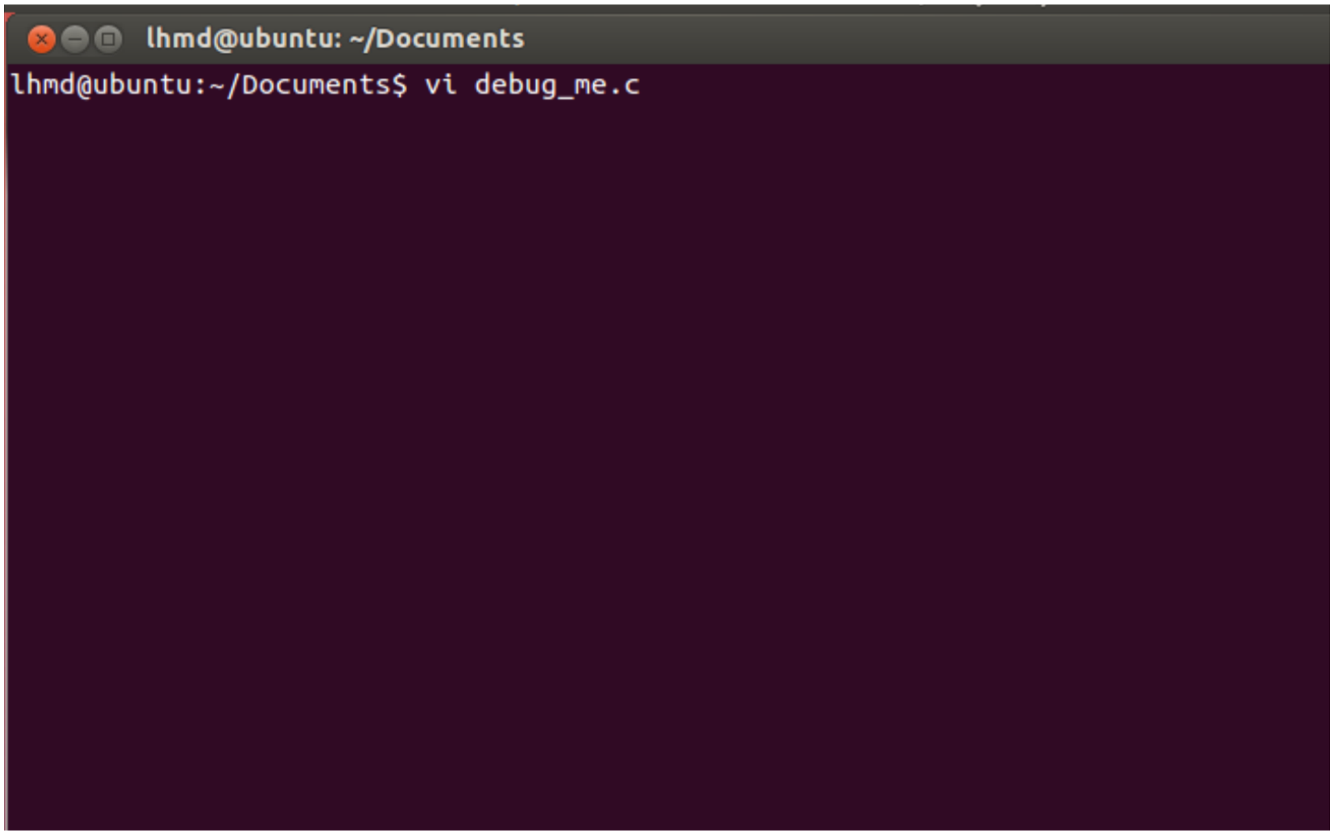
使用`gcc`编译运行`hello.c`文件



```
lhmd@ubuntu:~/Documents$ gcc hello.c
lhmd@ubuntu:~/Documents$ ./a.out
hello world!lhmd@ubuntu:~/Documents$
```

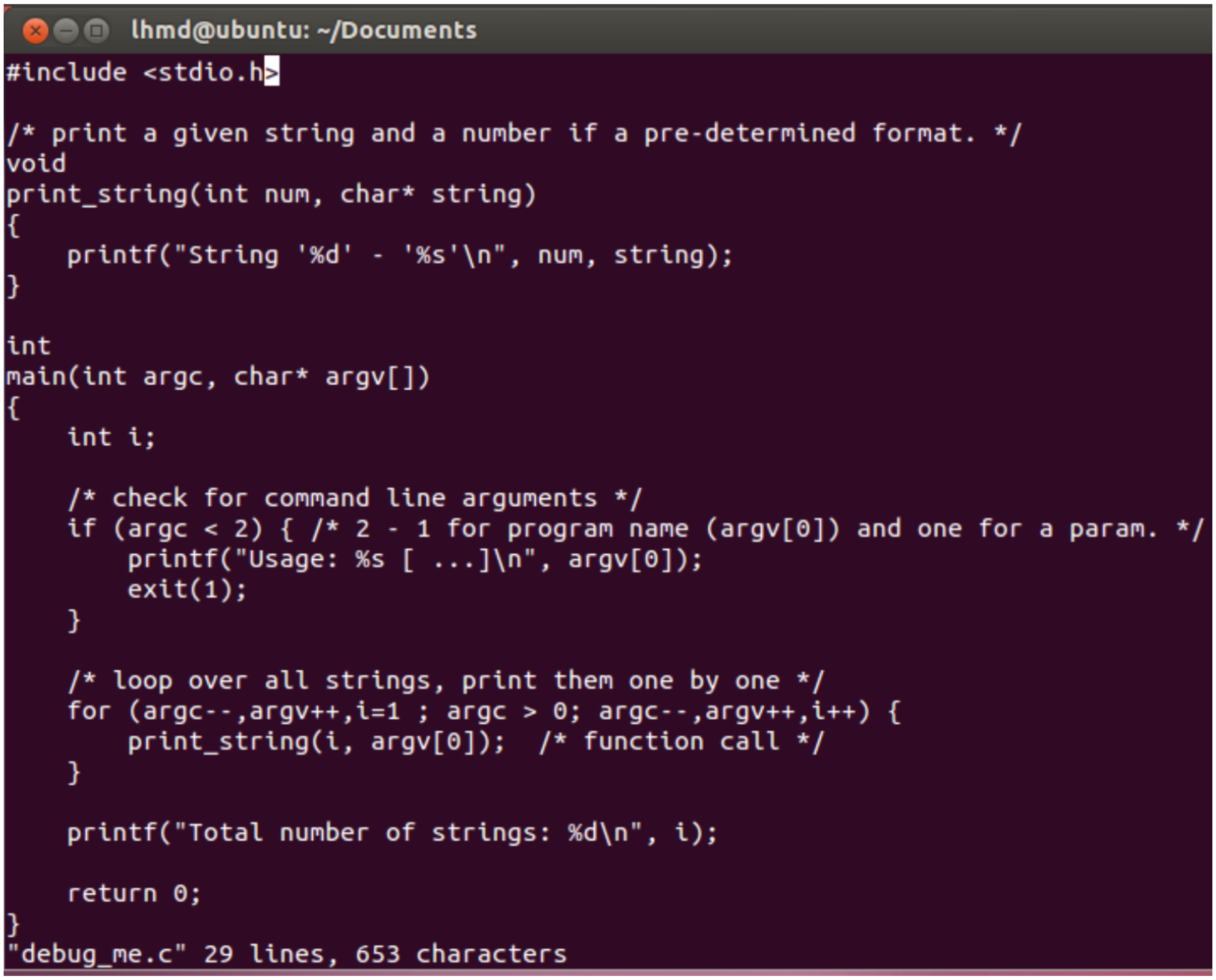
3 调试程序

3.1 创建文件 `debug_me.c`



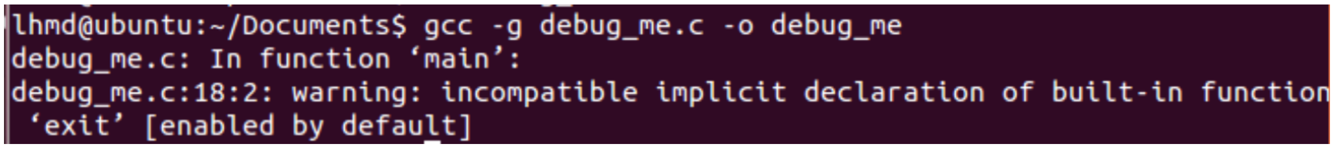
```
lhmd@ubuntu:~/Documents$ vi debug_me.c
```

复制代码并做适当修改，`Ctrl+shift+V` 粘贴到文件中并保存



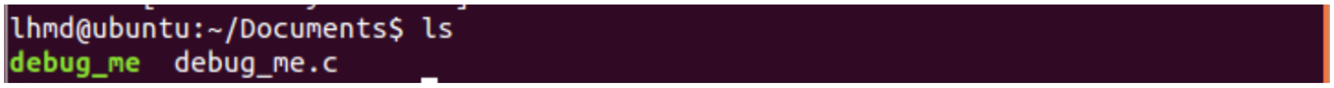
```
#include <stdio.h>

/* print a given string and a number if a pre-determined format. */
void
print_string(int num, char* string)
{
    printf("String '%d' - '%s'\n", num, string);
}

int
main(int argc, char* argv[])
{
    int i;

    /* check for command line arguments */
    if (argc < 2) { /* 2 - 1 for program name (argv[0]) and one for a param. */
        printf("Usage: %s [ ...]\n", argv[0]);
        exit(1);
    }

    /* loop over all strings, print them one by one */
    for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
        print_string(i, argv[0]);   /* function call */
    }

    printf("Total number of strings: %d\n", i);

    return 0;
}
"debug_me.c" 29 lines, 653 characters
```

3.2 调用 gdb 调试 `debug_me.c`



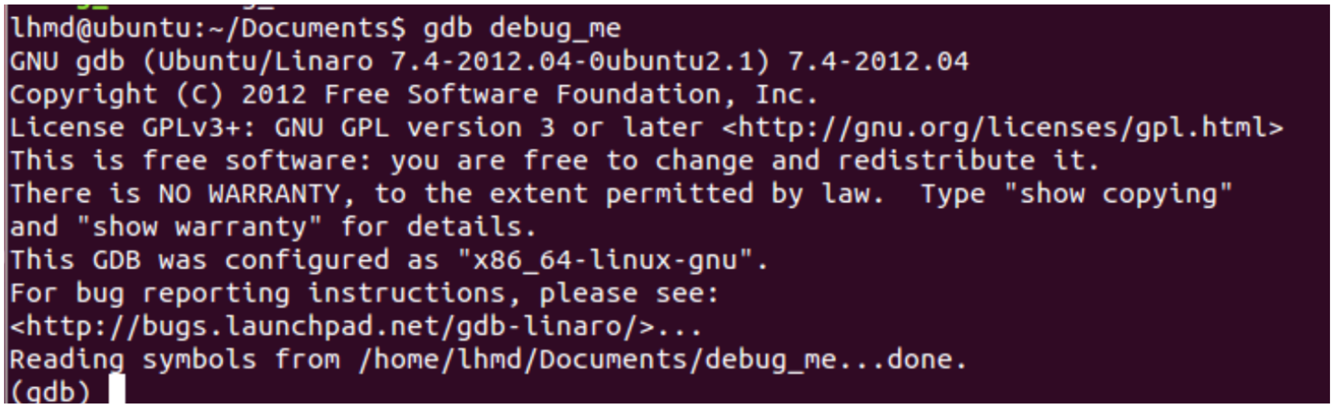
```
lhmd@ubuntu:~/Documents$ gcc -g debug_me.c -o debug_me
debug_me.c: In function ‘main’:
debug_me.c:18:2: warning: incompatible implicit declaration of built-in function
 ‘exit’ [enabled by default]
```

生成`debug_me`文件



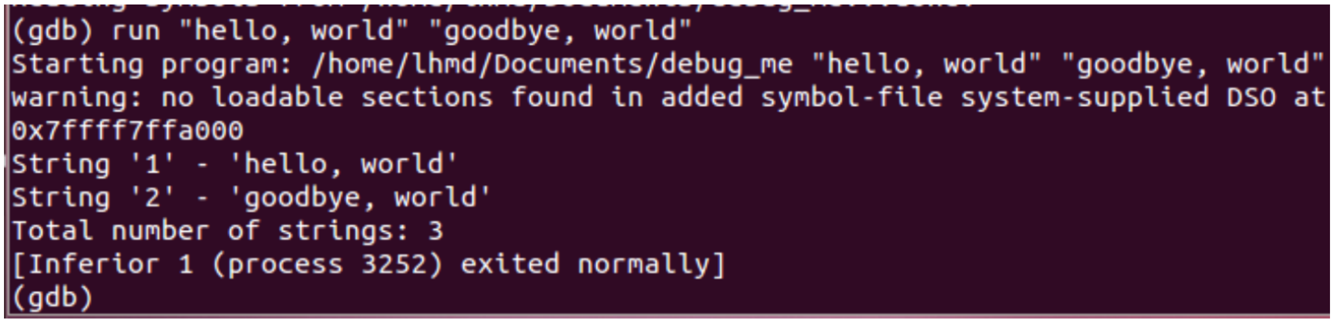
```
lhmd@ubuntu:~/Documents$ ls
debug_me  debug_me.c
```

运行gdb



```
lhmd@ubuntu:~/Documents$ gdb debug_me
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2.1) 7.4-2012.04
Copyright (C) 2012 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".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>...
Reading symbols from /home/lhmd/Documents/debug_me...done.
(gdb)
```

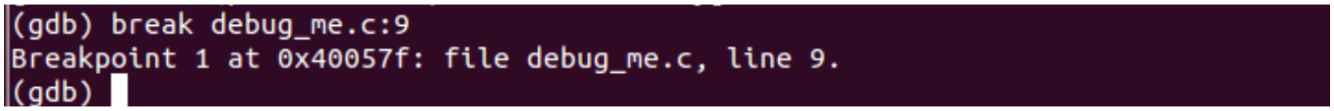
3.3 在gdb里面运行程序



```
(gdb) run "hello, world" "goodbye, world"
Starting program: /home/lhmd/Documents/debug_me "hello, world" "goodbye, world"
warning: no loadable sections found in added symbol-file system-supplied DSO at
0x7ffff7ffa000
String '1' - 'hello, world'
String '2' - 'goodbye, world'
Total number of strings: 3
[Inferior 1 (process 3252) exited normally]
(gdb)
```

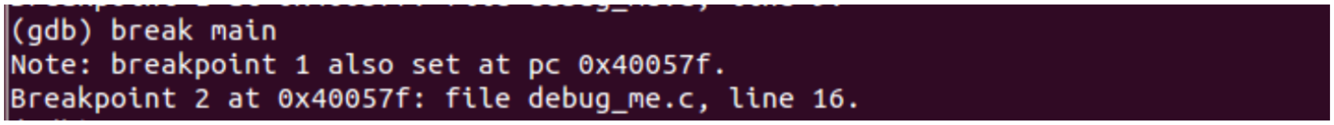
3.4 设置断点

3.4.1 指定特定代码行停止



```
(gdb) break debug_me.c:9
Breakpoint 1 at 0x40057f: file debug_me.c, line 9.
(gdb)
```

3.4.2 指定一个函数名，每次调用时中断

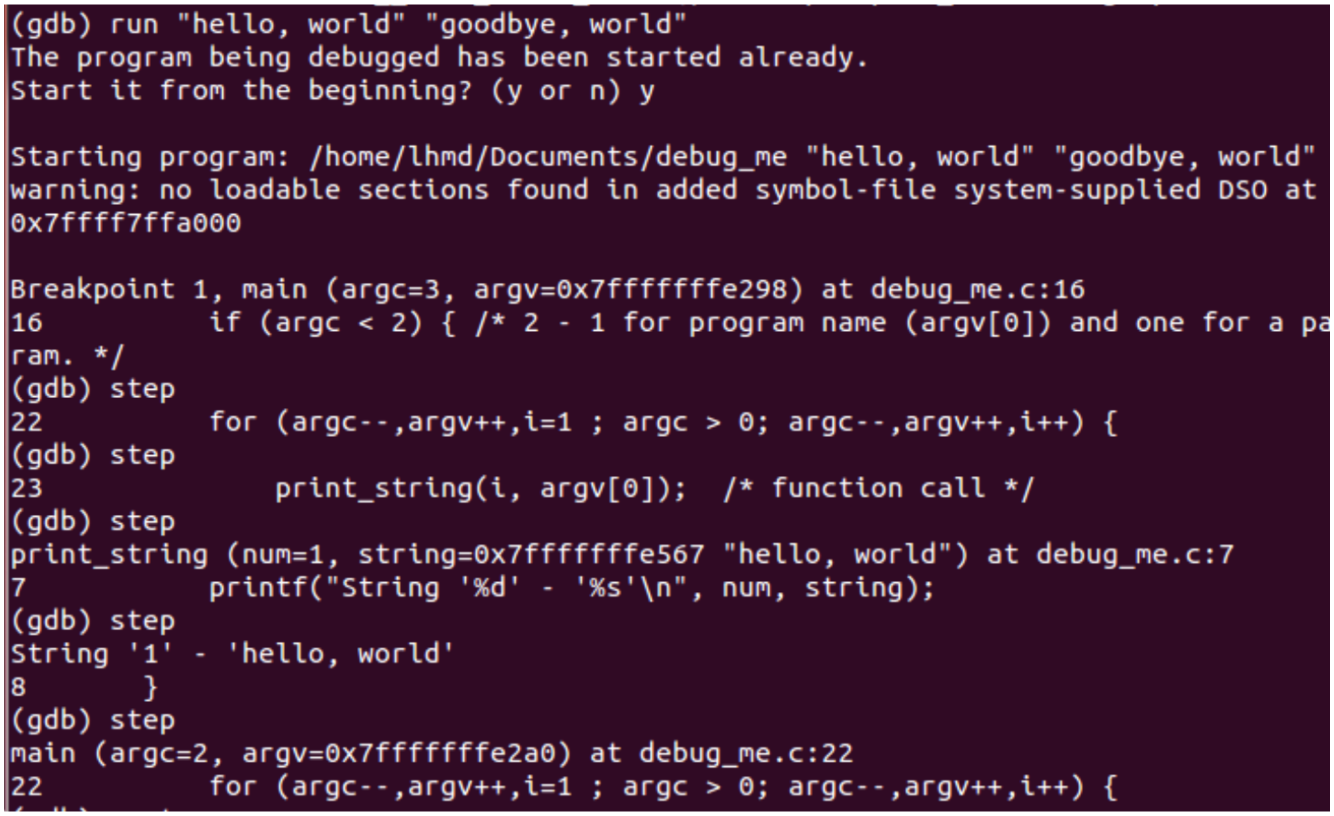


```
(gdb) break main
Note: breakpoint 1 also set at pc 0x40057f.
Breakpoint 2 at 0x40057f: file debug_me.c, line 16.
```

3.5 运行程序

3.5.1 step

step遇到一个函数后，会进入这个函数的堆栈

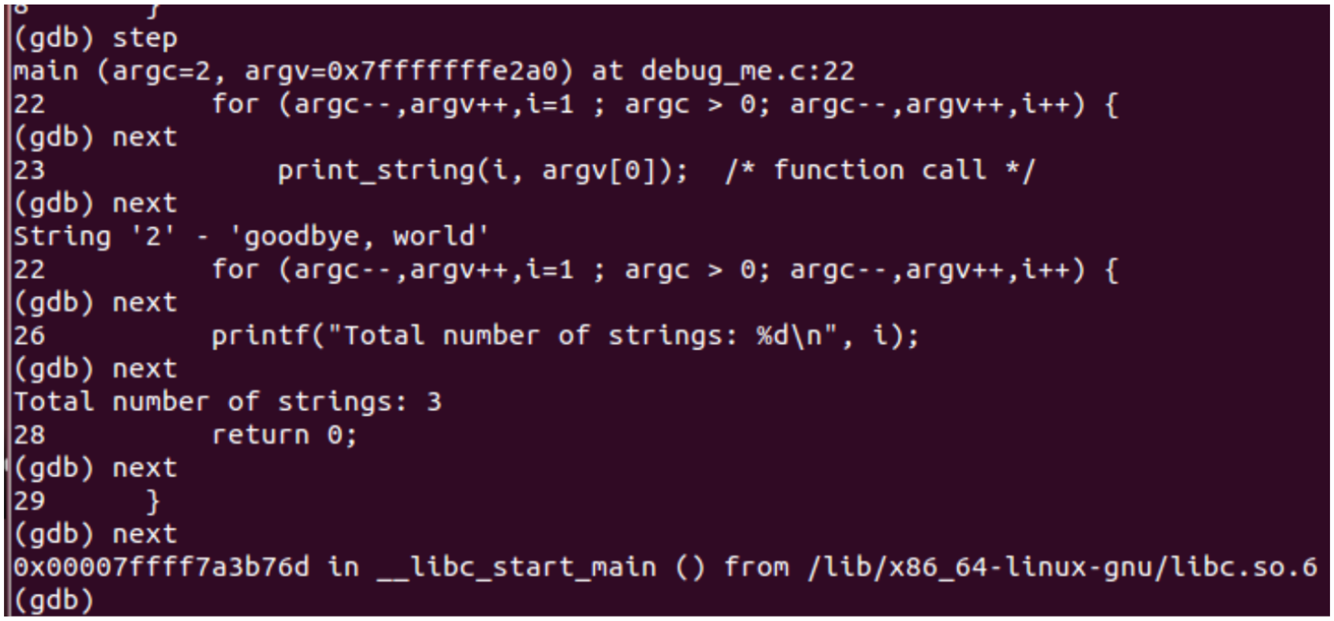


```
(gdb) run "hello, world" "goodbye, world"
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: /home/lhmd/Documents/debug_me "hello, world" "goodbye, world"
warning: no loadable sections found in added symbol-file system-supplied DSO at
0x7ffff7ffa000

Breakpoint 1, main (argc=3, argv=0x7fffffffe298) at debug_me.c:16
16          if (argc < 2) { /* 2 - 1 for program name (argv[0]) and one for a pa
ram. */
(gdb) step
22          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) step
23              print_string(i, argv[0]);  /* function call */
(gdb) step
print_string (num=1, string=0x7fffffffe567 "hello, world") at debug_me.c:7
7           printf("String '%d' - '%s'\n", num, string);
(gdb) step
String '1' - 'hello, world'
8       }
(gdb) step
main (argc=2, argv=0x7fffffffe2a0) at debug_me.c:22
22          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
```

如图，在运行到`print_string(i, argv[0]);`时，step进入了被调用函数的堆栈。

3.5.2 next

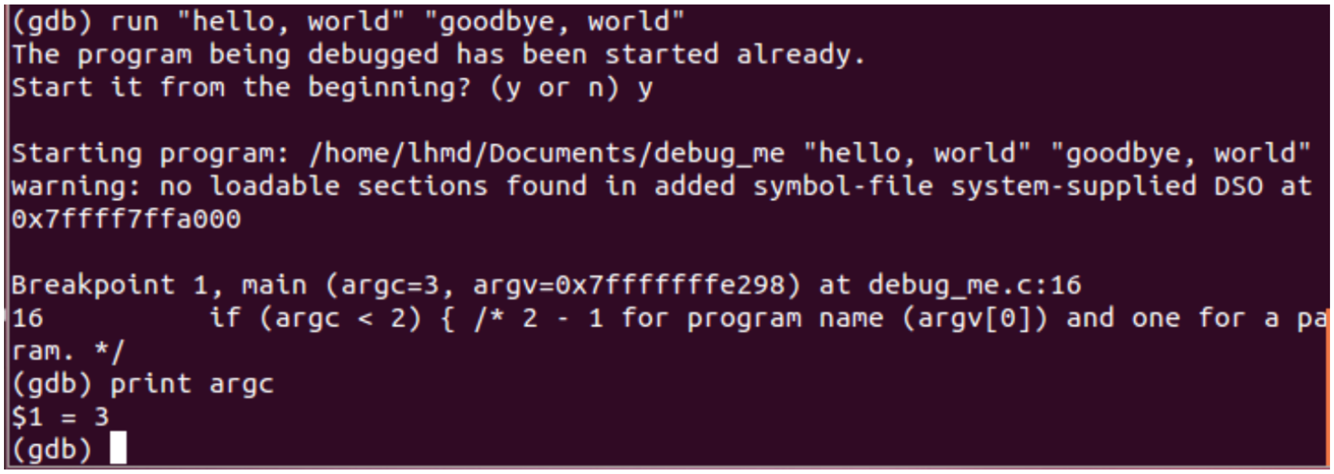
next遇到一个函数后，不会进入这个函数的堆栈



```
(gdb) step
main (argc=2, argv=0x7fffffffe2a0) at debug_me.c:22
22          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) next
23              print_string(i, argv[0]);  /* function call */
(gdb) next
String '2' - 'goodbye, world'
22          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) next
26          printf("Total number of strings: %d\n", i);
(gdb) next
Total number of strings: 3
28          return 0;
(gdb) next
29      }
(gdb) next
0x00007ffff7a3b76d in __libc_start_main () from /lib/x86_64-linux-gnu/libc.so.6
(gdb)
```

如图，在运行到`print_string(i, argv[0]);`时，next没有进入被调用函数的堆栈。

3.6 打印变量

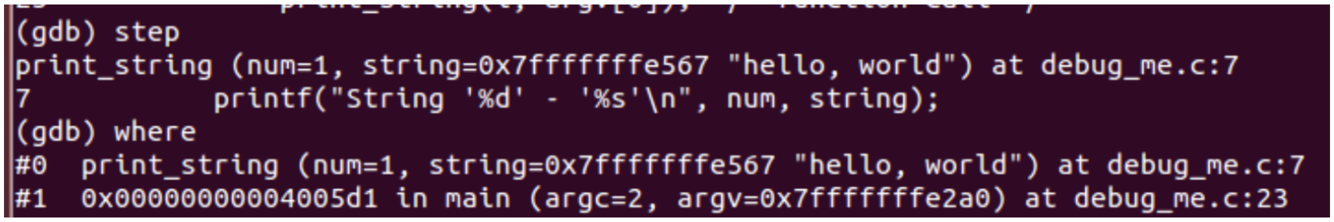


```
(gdb) run "hello, world" "goodbye, world"
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: /home/lhmd/Documents/debug_me "hello, world" "goodbye, world"
warning: no loadable sections found in added symbol-file system-supplied DSO at
0x7ffff7ffa000

Breakpoint 1, main (argc=3, argv=0x7fffffffe298) at debug_me.c:16
16          if (argc < 2) { /* 2 - 1 for program name (argv[0]) and one for a pa
ram. */
(gdb) print argc
$1 = 3
(gdb) 
```

如图，使用`print`打印变量`argc`，打印出值为3。

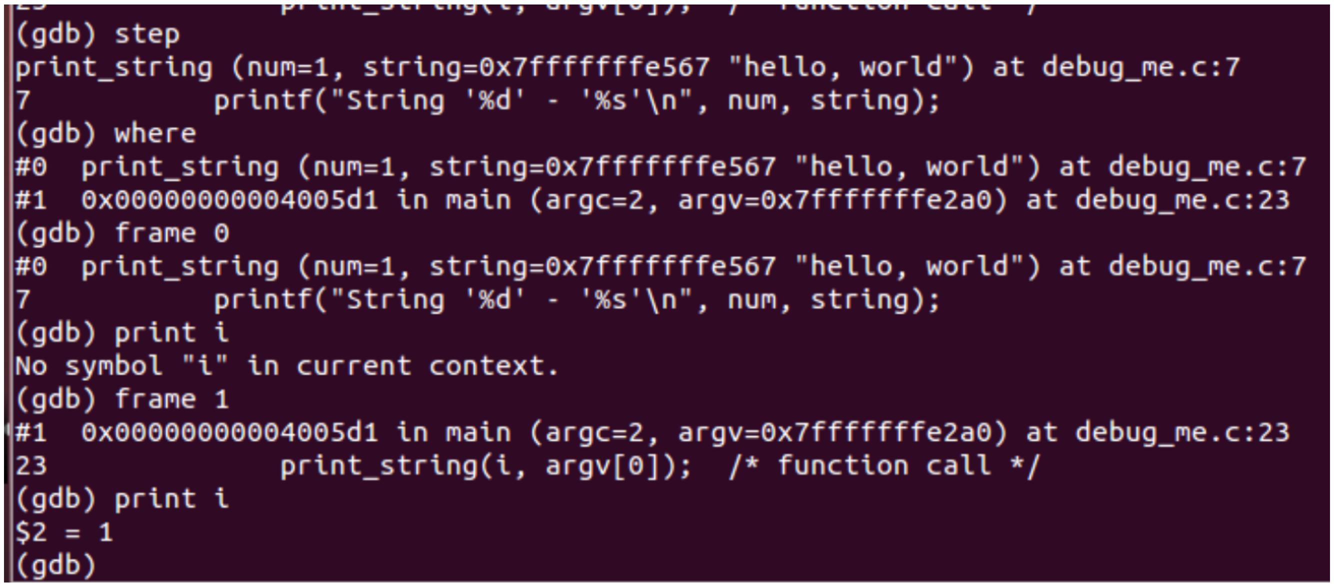
3.7 检查函数调用堆栈

逐步运行程序到第 7 行。检查函数调用堆栈，输入`where`



```
(gdb) step
print_string (num=1, string=0x7fffffffe567 "hello, world") at debug_me.c:7
7           printf("String '%d' - '%s'\n", num, string);
(gdb) where
#0  print_string (num=1, string=0x7fffffffe567 "hello, world") at debug_me.c:7
#1  0x00000000004005d1 in main (argc=2, argv=0x7fffffffe2a0) at debug_me.c:23
```

键入“frame 0”和“frame 1”以查看区别，不同的frame中打印的变量i的值不同，原因是子 函数中变量i没有被定义，而主函数中i=1



```
(gdb) step
print_string (num=1, string=0x7fffffffe567 "hello, world") at debug_me.c:7
7           printf("String '%d' - '%s'\n", num, string);
(gdb) where
#0  print_string (num=1, string=0x7fffffffe567 "hello, world") at debug_me.c:7
#1  0x00000000004005d1 in main (argc=2, argv=0x7fffffffe2a0) at debug_me.c:23
(gdb) frame 0
#0  print_string (num=1, string=0x7fffffffe567 "hello, world") at debug_me.c:7
7           printf("String '%d' - '%s'\n", num, string);
(gdb) print i
No symbol "i" in current context.
(gdb) frame 1
#1  0x00000000004005d1 in main (argc=2, argv=0x7fffffffe2a0) at debug_me.c:23
23              print_string(i, argv[0]);  /* function call */
(gdb) print i
$2 = 1
(gdb)
```