Lab 2.2 Running a Hello World Program in C using GCC

1 Goals

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

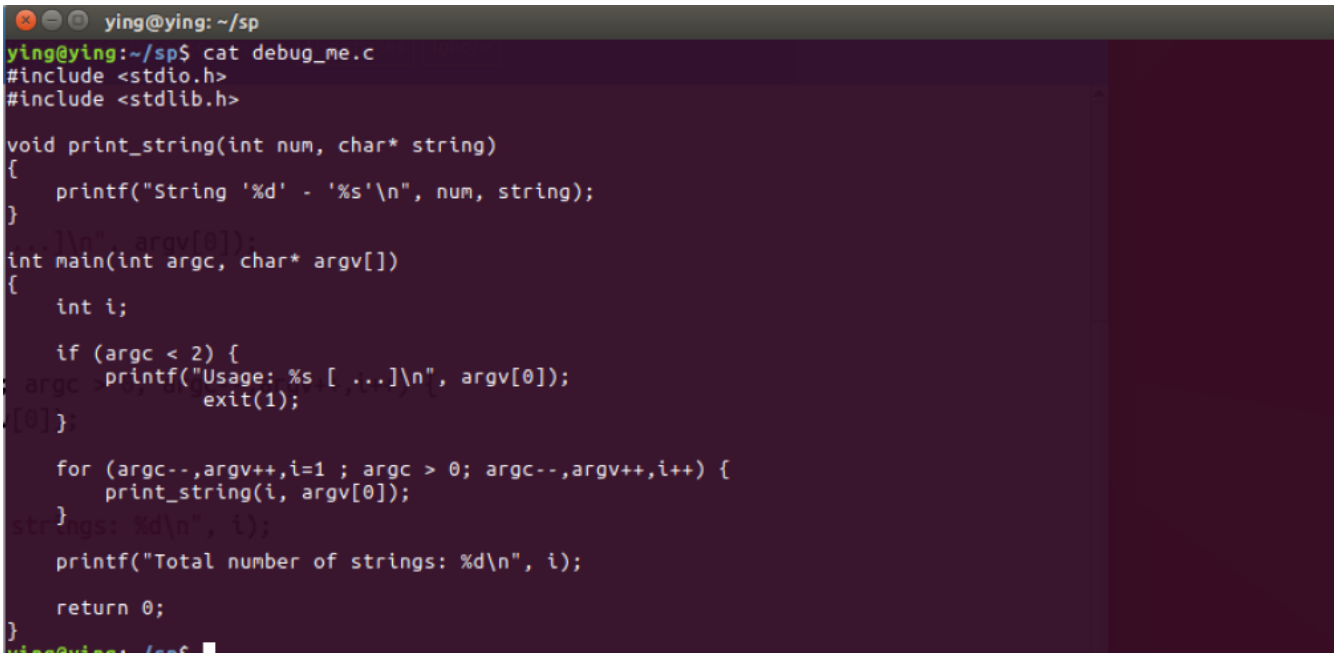
1. Learn to run a program in gcc.
2. Learn to debug a program in gdb.

2 Steps

创建文件`debug_me.c`

添加`stdio.h`和`stdlib.h`库文件

创建完成后通过`cat`命令来查看代码内容。



运行`gcc`和`gdb`命令

输入`run "hello,world" "goodbye,world"`将参数传入命令行并观察输出。

```
ying@ying:~/sp$ gcc -g debug_me.c -o debug_me
ying@ying:~/sp$ gdb debug_me
GNU gdb (Ubuntu 7.12.50.20170314-0ubuntu1) 7.12.50.20170314-git
Copyright (C) 2017 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 debug_me...done.
(gdb) run "hello,world" "goodbye,world"
Starting program: /home/ying/sp/debug_me "hello,world" "goodbye,world"
String '1' - 'hello,world'
String '2' - 'goodbye,world'
Total number of strings: 3
[Inferior 1 (process 2626) exited normally]
```

设置断点

可以通过`break debug_me:[行号]`来为某一行设置断点，或通过`break [函数名]`来为某个函数设置断点。

```
(gdb) break debug_me.c:13
Breakpoint 1 at 0x55555555471b: file debug_me.c, line 13.
(gdb) break main
Note: breakpoint 1 also set at pc 0x55555555471b.
Breakpoint 2 at 0x55555555471b: file debug_me.c, line 13.
(gdb)
```

断点运行

通过`next`和`step`命令进行程序执行调试

Next和Step的区别为：当step遇到函数时，会进入函数并继续一步步地执行，而当next遇到函数时，不会进入函数而是将其当成一个语句来执行，如下图：

```
(gdb) run "hello,world" "goodbye,world"
Starting program: /home/ying/sp/debug_me "hello,world" "goodbye,world"
Breakpoint 1, main (argc=3, argv=0x7fffffffe058) at debug_me.c:13
13          if (argc < 2) {
(gdb) next
18          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) next
19              print_string(i, argv[0]);
(gdb) next
String '1' - 'hello,world'
18          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) step
19              print_string(i, argv[0]);
(gdb) step
print_string (num=2, string=0x7fffffffe3bd "goodbye,world") at debug_me.c:6
6           printf("String '%d' - '%s'\n", num, string);
(gdb) step
String '2' - 'goodbye,world'
7       }
```

通过`Print`命令来打印变量的值，如下图所示：

```
Starting program: /home/ying/sp/debug_me "hello,world" "goodbye,world"

Breakpoint 1, main (argc=3, argv=0x7fffffffe058) at debug_me.c:13
13          if (argc < 2) {
(gdb) print i
$1 = 0
(gdb) next
18          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) print i
$2 = 0
(gdb) next
19              print_string(i, argv[0]);
(gdb) print i
$3 = 1
(gdb) next
String '1' - 'hello,world'
18          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) print i
$4 = 1
(gdb) next
19              print_string(i, argv[0]);
(gdb) print i
$5 = 2
(gdb) next
String '2' - 'goodbye,world'
18          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) next
22          printf("Total number of strings: %d\n", i);
(gdb) print i
$6 = 3
(gdb) next
Total number of strings: 3
```

查看函数栈

运行到子函数时，通过`where`命令来看堆栈中的函数情况，通过`frame`来进行切换。

可以看到，在不同的frame中来打印变量`i`的值会出现不同的结果

原因是在子函数中变量`i`没有被定义，而在主函数中变量`i`的值为1

```
Breakpoint 1, main (argc=3, argv=0x7fffffffe058) at debug_me.c:13
13          if (argc < 2) {
(gdb) step
18          for (argc--,argv++,i=1 ; argc > 0; argc--,argv++,i++) {
(gdb) step
19              print_string(i, argv[0]);
(gdb) step
print_string (num=1, string=0x7fffffffe3b1 "hello,world") at debug_me.c:6
6           printf("String '%d' - '%s'\n", num, string);
(gdb) where
#0  print_string (num=1, string=0x7fffffffe3b1 "hello,world") at debug_me.c:6
#1  0x000055555555476c in main (argc=2, argv=0x7fffffffe060) at debug_me.c:19
(gdb) frame 0
#0  print_string (num=1, string=0x7fffffffe3b1 "hello,world") at debug_me.c:6
6           printf("String '%d' - '%s'\n", num, string);
(gdb) print i
No symbol "i" in current context.
(gdb) frame 1
#1  0x000055555555476c in main (argc=2, argv=0x7fffffffe060) at debug_me.c:19
19              print_string(i, argv[0]);
(gdb) print i
$9 = 1
(gdb)
```