CSC3320 System Level Programming
Lab Assignment 8 - Post-Lab

Due at 11:59 pm on Friday, March 12, 2021

Purpose: Learn how to use debugger in **gdb** to debug a program in Unix.

Part 1:

You are given a C program "q1.c" as below. But since there are no enough comments in the program, it is hard to find out the feature of the function ***foo***. So let us trace the execution of the program and find out what ***foo*** does. Please follow the steps below and answer the questions accordingly.

```
#include <stdio.h>

int foo(int num)
{
    int rev_num = 0;
    while (num > 0)
    {
        rev_num = rev_num*10 + num%10;
        num = num/10;
    }
    return rev_num;
}

/* Driver program to test foo */
int main()
{
    int num = 1125;
    printf("Result is %d", foo(num));
    return 0;
}
```

1) Compile “q1.c” with **–g** option so that we can debug the executable using **gdb**. `$gcc -o q1 -g q1.c`

2) Lauch **gdb** for “q1”.
`$gdb q1`

3) List the source code of “q1.c” from line 1.
`(gdb)list 1`

4) Set a breakpoint at the line of statement “while (num > 0)”.
Question: Write your command.

To set a breakpoint at the line of the statement we use command **b 5**

4) Run the program until the first breakpoint.
Question: Write your command.
To run the program until the first break point we use the command **run**

5) Use **display** to show the value of rev_num and num at each time when program stops.
`(gdb)display rev_num`
`(gdb)display num`

```
-bash-4.2$ vi q1.c
-bash-4.2$ -bash-4.2$ gcc -o q1 -g q1.c
-bash-4.2$ gdb q1
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-120.el7
Copyright (C) 2013 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/amandapaka2/q1...done.
(gdb) list 1
1       #include <stdio.h>
2
3       int foo(int num)
4       {
5
6          int rev_num = 0;
7          while (num > 0)
8          {
9
10             rev_num = rev_num*10 + num%10;
(gdb) b 5
Breakpoint 1 at 0x400534: file q1.c, line 5.
(gdb) run
Starting program: /home/amandapaka2/q1

Breakpoint 1, foo (num=1125) at q1.c:6
6          int rev_num = 0;
Missing separate debuginfos, use: debuginfo-install glibc-2.17-324.el7_9.x86_64
(gdb) display rev_num
1: rev_num = 0
(gdb) display num
2: num = 1125
```

6) Run the while loop step by step using command **n** multiple times. `(gdb)n`

Question: check the value of rev_num and num after each iteration and fill in the table below.

	1st iteration	2nd iteration	3rd iteration	4th iteration
num	112	11	1	0
rev_num	5	52	521	5211

7) When the program terminates, quit **gdb** using command **q**.
`(gdb)q`

8) *Question: Now can you tell what the function foo does?*

The function of the foo is to reverse the given number. For example, in this lab, we can see that how the number 1152 becomes 5211, which is basically is the reverse order of 1152 at the end of the function

Part 2:

You are given a C program "q2.c" as below. This program is used to calculate the average word length for a sentence (a string in a single line):

```
Enter a sentence: It was deja vu all over again.
Average word length: 3.4
```

For simplicity, the program considers a punctuation mark to be part of the word to which it is attached. And it displays the average word length to one decimal place.

```
1
 #include <stdio.h>
2
 3 int main() {
4
 5 int letters;
 6 int words;
 7 char character;
8
```

```
9 printf("Enter a Sentence: ");
```



```
    while((character=getchar()) != \n){
        if(character != ' '){
            if(!space){
                words++;
                space=1;
            }
            letters++;
      }else
         space = 0;
    }

    printf("Average word length : %.1f", letters/words);

    return 0;
}
```

However, there are multiple errors in the given C program. Please correct complier errors and use **gdb** to debug the program and find out the errors.

Question: Please write down the line numbers containing the errors and show how to correct them.

(Note: you do not need to write down the commands you issued in **gdb**.)

Errors

q2.c: In function ‘main’ :

q2.c:11:5: error: stray ‘\’ in program

while((character==getchar()) != \n) {

q2.c:11.37: error: ‘n’ undeclared (first use in this function)

while((character=getchar()) != \n) { //in this command there is no line variable named spaces, therefore we need to add int spaces, before line 12

q2.c:11:37: note: each undeclared identifier is reported only once for each function it appears in

q2.c:12:26: warning: multi-character character constant [-Wmultichar]

if(character != ‘ ‘) {

q2.c:13:19: error: ‘space’ undeclared (first use in this function)

if(!space) {

```
-bash-4.2$ vi q2.c
-bash-4.2$ gcc -o q2 -g q2.c
q2.c: In function ‘main’:
q2.c:11:5: error: stray ‘\’ in program
     while((character=getchar()) != \n) {
     ^
q2.c:11:37: error: ‘n’ undeclared (first use in this function)
     while((character=getchar()) != \n) {
                                    ^
q2.c:11:37: note: each undeclared identifier is reported only once for each function it appears in
q2.c:12:26: warning: multi-character character constant [-Wmultichar]
          if(character != '  ') {
                          ^
q2.c:13:19: error: ‘space’ undeclared (first use in this function)
               if(!space) {
                   ^
-bash-4.2$
```

Corrected code screen shot

```
OpenSSH SSH client
#include <stdio.h>

int main() {

    int letters, words, space;
    char character;
    printf("Enter a sentence: \n");
    while((character=getchar()) != '\n'){
          if(character != ' '){
               if(!space){
                     words++;
                     space=1;
                }
            letters++;
            } else{
                    space = 0;

}
}
          printf("Average word length: %.1f", (double)(letters/words));
            return 0;
}
~
~
~
~
~
~
~
~
~
~
```

Submission:

• Please follow the instructions below step by step, and then write a report by answering the questions and upload the report (named as Lab8_FirstNameLastName.pdf or Lab8_FirstNameLastName.doc) to Google Classroom, under the rubric Lab 8 Out-of-lab Assignment. • Please add the lab assignment NUMBER and your NAME at the top of your file sheet.