Wei Fei (2538810)
EECS678 Lab2 Report
Prasanth Vivekanandan
02/02/14

1. What is wrong with the original code that eventually causes it to crash? (Obviously, the program fails becuase of an invalid memory reference producing a segmentation fault -- what programming error led to the seg fault?)

Answer:

The problem is on the **line 3673**:

```
if (signal_in_progress (DEBUG_TRAP) == 0 && running_trap == 0)
#endif
{
    FREE (the_printed_command_except_trap);
    the_printed_command_except_trap = the_printed_command;
  }
```

should be replaced by

```
 if (signal_in_progress (DEBUG_TRAP) == 0 && running_trap == 0)
   {
    FREE (the_printed_command_except_trap);
    the_printed_command_except_trap = savestring (the_printed_command);
   }
```

2. Describe how you diagnosed the problem with the original code. If you used GDB, which commands did you find most helpful? If you did not, what tools were most helpful in diagnosing the problem?

Answer:

- Use command "**gdb ./bash-4.2/bash**" to enter the debugging mode of gdb.

- Use command "**r ./finder.sh bash-4.2/ execute 20**" to run the debugger.

- Use command "**bt**" to back trace all possible errors. I found that there may have something wrong on the line 3672.

- Use command "**quit**" to quit the gdb debugger, and then go to the folder bash-4.2 and type "**vim execute_cmd.c**" to diagnose errors.

***** I think the command "**bt**" was most helpful because I can clearly see where were some possible error. I can enter the file and find them directly.

3. Describe how your solution fixes the problem. Are you confident your solution is correct?

Answer:

- Use command “**ctags -R**” to generate a new file named “tags”

- Type “**vim execute_cmd.c**” to view the code

- Move cursor to the position of the function “**savestring**”, and then type “**Ctrl +]**” to see the definition of the function.

- Choose **No.4 definition** of the function “**savestring**”, we may see the reason.

Reason:

The function “**savestring**” has this code: “**ret = (char *)xmalloc (strlen (s) + 1);**” It dynamically allocated new memory in heap. **If we did not use the function, we will not have this dynamical memory.**

Also, I see the final result was matching the standard result. **I am confidence with my solution**.