Gehrig Keane
2727430
EECS 678: Lab 2 GDB

1. The original code seg-faulted at line 3672 in /bash-4.2/execute_cmd.c

```
3667 {
3668 /* copy the_printed_command to the_printed_command_except_trap. Free up
3669 * old memory for the_printed_command_except_trap before allocating any
3670 * new memory.
3671 */
3672       FREE (the_printed_command_except_trap);
3673       the_printed_command_except_trap = the_printed_command;
3674 }
```

   Namely, GDB reports the above free as the issue at hand. Therefore, lets investigate FREE's in the rest of this file to get some context for what's going on. After using vim to page through instances of 'FREE' we come across

```
3054 {
3055       FREE (the_printed_command_except_trap);
3056       the_printed_command_except_trap = savestring (the_printed_command);
3057 }
```

   At face value, we can deduce that our issue probably rests in the savestring function, so let's investigate the function definition of savestring to see what's going on. After initializing ctags -R this task is trivial. We come across the following definition in /bash-4.2/general.h.

```
69         #define savestring(x) strcpy (xmalloc (1 + strlen (x)), (x))
```

   The absence of the savestring call aligns with the inconsistent memory management resultant in the segmentation fault. Thereby, if we add the savestring syntax to line 3672 and rebuild, everything functions as expected.

2. Diagnosing the code with GDB included initializing GDB much like the slides instructed. After back-tracing the segmentation fault it became readily apparent that the FREE in line 3672 was the issue at hand. I found that the most useful GDB commands were 'backtrace', 'info locals' and 'info args'; additionally, vim coupled with ctags made looking up the savestring definition very straightforward.

3. Based on various contextual excerpts in the execute_cmd.c file I'm more than confident the abovementioned solution addresses the problem at hand. Another facet to said confidence resides in the purpose of savestring, everything is quite consistent.