EEE243 – Applied Computer Programming

Debugging

Outline

1. What is debugging
2. Sources of bugs
3. Debugging process
4. Compilation errors
5. Code Inspection
6. Trace
7. Interactive debugging

What is debugging?

Debugging is the process of finding and resolving of defects or problem within the program that prevent correct operation of computer software or a system. [1]

Sources of Bugs

- coding errors
- design errors
- complex interactions
- poor user interface designs
- hardware failure

From [2]

Debugging Process

1. Recognize that a bug exists
2. Isolate the source of the bug
3. Identify the cause of the bug
4. Determine a fix for the bug
5. Apply the fix and test it

From [2]

Recognize that a bug exists

- At compilation
  - You get an error
- At runtime
  - The program crashes
  - There is an unexpected result

Isolate the source of the bug

- Difficult
- Source of the error is not always the cause
- Divide and conquer approach (wolf fence)
- That is where an interactive debugger or adding traces might be useful.

Identify the cause of the bug

Possible causes include:

- unexpected input (inappropriate input handling)
- logic error in the code
- poor resource management (e.g. memory leak)

Determine a fix for the bug

- Sometimes easy
- Sometimes involves a re-design
- Fix might introduce new bugs elsewhere

Apply the fix and test it

Important to test in order to

1. Ensure the initial problem is fixed
2. Ensure no other bugs were introduced

Compilation errors

- Something in your program does not respect the language syntax
- IDEs will often highlight syntax errors
- Sometimes, the error is more subtle and a more systematic approach will be needed

```
../src/student.c:33:9: error: use of undeclared identifier 'degs';
```

```
../src/student.c:11:10: fatal error: 'sting.h' file not found
```

```
ld: 1 duplicate symbol for architecture x86_64
```

Code Inspection

Read the code line by line to detect the problem

```
double sinus(double x, int n) {
    double rad = deg_to_rad(x);
    double res = 1;
    for (int i = 0; i <= n; i++) {
        res += power(-1, i) / factorial(2 * i + 1) *
               power(rad, 2 * i + 1);
    }
    return res;
}
```

output

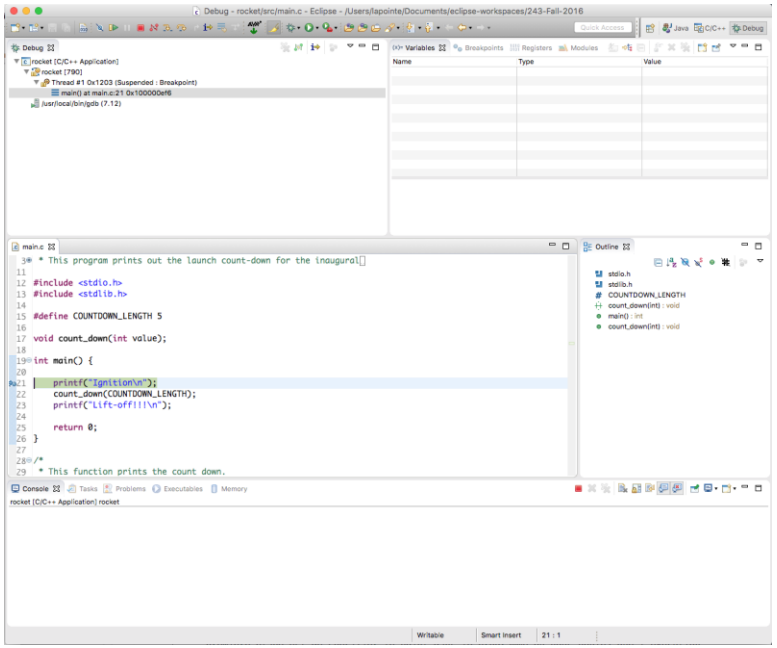
```
Sine of 90 is 2.000
Sine of 0 is 1.000
```

Trace

- Printing debugging cues such as function names, variable content, etc. to the console.
- Outputting information to a file (log)
- E.g. Using a `printf` to display the name of the function being executed to find out where in the program the crash occurs.

Interactive Debugger

- A software tool running along your program
- Allows to:
  - set breakpoints
  - inspect memory content
  - move step by step



Exercise 1

```
1  #include <stdio.h>
2  #include <string.h>
3
4  // implement the Caesar Cipher
5  void cipher(char *str, int key);
6
7  int main(void) {
8      char str[] = "hello world";
9      cipher(str, 4);
10     printf("Your encrypted text is: %s\n", str);
11     return 0;
12 }
13
14 void cipher(char *str, int key) {
15     if (key > 25)
16         key = key%25;
17     for (i=0; i<strlen(str); i++) {
18         if (str[i] == ' ')
19            continue;
20         else {
21            str[i] = str[i] + key;
22            if (str[i] > 'z') {
23              str[i] = 'a' + str[i] - 'z';
24            }
25         }
26     }
27 }
```

The intent of the call to cypher on line 9 is that the content of `str` will be encrypted according to the Caesar Cipher. The Caesar Cipher offsets the characters by a key, i.e. if the key is 1, the text “hello world” will become “ifmmp xpsme”. There are two errors in the program. Identify the errors and suggest a correction for each. After the problems are solved, what would be printed on line 10?

Exercise 2

The debugging exercise is with the dynamically growing array.

- A user reported that some data does not get inputted in the array. Someone attempted to fix the issue and completely broke the program.
- A user also reported that sometimes, the program crashes unexpectedly. She is not sure what input causes it.
- Another issue is that when the array becomes very large, the program crashes.

Questions?

References

[1] Debugging, https://en.wikipedia.org/w/index.php?title=Debugging&oldid=804684078 (last visited Nov. 1, 2017).

[2] Computer Programming Principles/Maintaining/Debugging, https://en.wikibooks.org/w/index.php?title=Computer_Programming_Principles/Maintaining/Debugging&oldid=3228031 (last visited November 1, 2017).