CS220 - Computer System II
Lab 5

Due: 10/05/2016, 11:59pm

1 Introduction

In this lab, you will play with pointers to modify variables in a caller function.

2 The program

In Prog.c given below, you are to implement `bad` function such that the print statement prints the name of student as "Quick brown fox jumped over the lazy dog" and age as 1000.

```
/* Prog.c */
#include <stdio.h>
void bad(int dummy)
{
  /* Your implementation goes here */
}

int main() {
  struct {
    char *name;
    int age;
  } student = { .name="John",
                .age = 22 };
  bad(sizeof(student));
  printf("student.name = %s\nstudent.age = %d\n", student.name, student.age
  );
  return 0;
}
```

You are not allowed to change main. All your changes must be confined within the bad function. Note that the above example is similar to the one discussed during the lecture.

Step 1 By viewing the disassembly of Prog.c, identify the location of student structure in main.

Step 2 Using a pointer in bad function, alter the contents of the students variable. Use gdb to determine the location and offsets of stack pointer, return address, etc. Here, you

will not be able to alter the name because it will be in a readonly section of the program. Therefore, you will allocate space using malloc and store the string in the allocated space. You will replace the pointer in student structure that points to name with the pointer that points to the newly allocated space.

Step 3 Compile your program and run it to ensure that the new values are printed.

3 Submitting the result

In Lab5.txt, record the return address that bad returns to. Also record the location on the stack where the return address is located. Create lab5_submission.tar.gz file comprising of Prog.c and lab5.txt with your implementation of the bad function. Upload the files to Blackboard.