15213 Lecture 7: Procedures

1 Getting Started

To obtain a copy of today's activity, log into a shark machine and do the following:

```
1.$ wget http://www.cs.cmu.edu/~213/activities/lec7.tar
2.$ tar xf lec7.tar
3.$ cd lec7
```

First run `$ ./act6` and follow the instructions on your screen. You may refer to the sheet from the first GDB activity as a reference.

2 Discussion Questions: act6

Use GDB's `c` command to progress through the activities. These questions accompany the program; as it poses each one, discuss with your partner and write your answer here.

Contents of the stack:

0x________________	← $rsp = 0x________________
0x________________	
…	

1. What was the meaning of the second number on the stack?

2. What are the semantics of the `ret` instruction?

3. Given your knowledge of the `ret` instruction, what must be the semantics of `call`?

4. Why does this optimization work? Can it be used on every call?

5. Given your knowledge of the `printf()` function, what is the first argument used for, and what is its type?

6. Where did the compiler place arguments 7 and 8? Why do you think this happened?

3 Discussion Questions: act7

7. Where does the `getV()` function allocate its array? How does it pass this location to `getValue()`?

8. What is this function doing?

4 Optional Endianness Preview

Rerun `act6` with the `m` argument and continue to the point where you printed the stack before.

1. What do you expect the first two *bytes* of the stack to contain?

2. Check your hypothesis by running `x/2xb $rsp`. In what order are each integer's bytes stored?