

Recursion: Code Along

CS2263 – Systems Software Development

1

Learning Outcomes

At the conclusion of this lecture students should be able to:

- Explain how we can use recursion to solve our problem of reading a string of an unknown length.
- Walk through the recursive function and describe both the base and general cases
- Differentiate the availability of functions based on where they're declared.

2

References

- Lu, Yung-Hsiang. 2015. Intermediate C Programming. CRC Press. New York. (Chapter 13)

3

General Form of Recursion

Base Case

- The case to which we have an answer

General Case

- The case that expresses the solution in terms of a call to itself with a smaller version of the problem

4

C: General Form of Recursion

```
T func(arguments){
    if (base case) //by checking arguments
        solve the problem
    else // recursive case
        func(simplified arguments)
}
```

5

What the Stack is Happening?

Each time we call the recursive function, a new stack frame is created holding the state of the current function call.

The previous stack frames all hold their state at the point the function was called.

In our example here, when the base case is found, there are five states for factorial on the stack.

What if they each held a read character instead?

```
int factorial(int iVal){
    int iFac;
    if(iVal == 1){
        printf("iVal (%p): %d\n", &iVal, iVal);
        return 1;
    }

    iFac = iVal * factorial(iVal-1);
    printf("iVal (%p): %d\n", &iVal, iVal);
    return iFac;
}

/*
 * Example Output for 5:
 *
 * iVal (0x7ffeed85a8b8): 1
 * iVal (0x7ffeed85a8e8): 2
 * iVal (0x7ffeed85a918): 3
 * iVal (0x7ffeed85a948): 4
 * iVal (0x7ffeed85a978): 5
 */
```

6

Reading a String of Unknown Length

Base Case

- We found the end of string and know its length

General Case

- We read a character and need to read again

7

And now... Coding with Rick

- A not quite feature film

8