

15213 Lecture 6: Assembly Control Flow

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/lec6.tar`
2. `$ tar xf lec6.tar`
3. `$ cd lec6`

Now run `$./act4` and follow the instructions on your screen. It will occasionally ask you discussion questions, whose answers you can record in the following section. Feel free to refer to the activity sheet from last class if you need a reference of GDB commands.

2 Discussion Questions

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

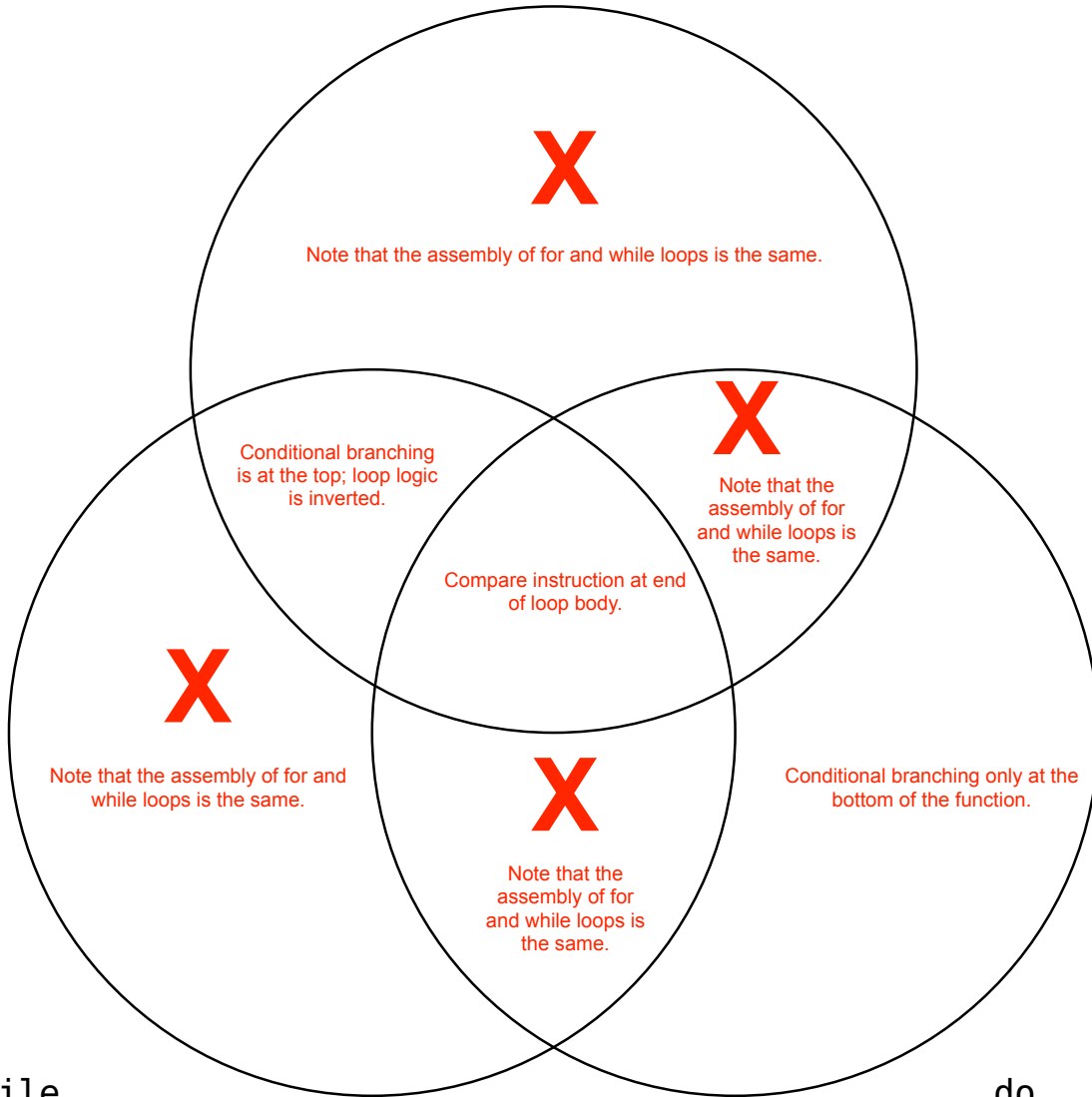
1. Which register corresponds to each of the function's parameters?
In the function signature `char carryAdd(unsigned long x, unsigned long y, unsigned long* sum)`, `%rdi` corresponds to unsigned long `x`, `%rsi` corresponds to unsigned long `y`, and `%rdx` corresponds to unsigned long* `sum`. Also, `%al` is the char return value.
2. If the tested condition has not changed following a backward jump, what will happen to the program?
The program will enter an infinite loop.
3. Read the assembly for this function and determine what it does.
The function `multThis` computes the factorial of the input argument.
4. What does and does not change between the three loops?
(You may use the Venn Diagram on the back.)

3 Advanced Follow-Up Question

If you finish early, look back at the assembly for `carryAdd()`. Do you think it would be easier to implement this function in C? Why or why not?

The function `carryAdd` actually cannot be implemented in C, since it requires checking the condition flags, which are specific to the x86 ISA but not C in general.

for



X

Note that the assembly of for and while loops is the same.

X

Note that the assembly of for and while loops is the same.

Compare instruction at end of loop body.

X

Note that the assembly of for and while loops is the same.

X

Note that the assembly of for and while loops is the same.

Conditional branching only at the bottom of the function.

while

do...while