Prep C
Game of life activity

Dr. Madhav Rao

Revision Date: July 27, 2016

Task 1: function pointers

Create a directory *fpointer* that hangs off your *clab* directory. Write a program named *calc.c* that takes three command line arguments (not including the program file name) and performs a simple calculation using function pointer. Here are some examples.

```
 $ ./calc 2 + 3
5

$ ./calc 5 / 2
2.5

$ ./calc 5 - 6
-1

$ ./calc 4 \* 6
24
```

In this actiivity you will initialize a function pointer to a function according to the operative character specified in the command line arguments as shown below:

```
fptr = add; if '+'  is found

fptr = subtract; if '-'  is found

fptr = multiply; if '*' is found

fptr = division; if '/' is found
```

You need to define *add*, *subtract*, *multiply*, and *division* functions accordingly. Test your program thoroughly.

Task 2: Callback functions

Your are supposed to process two data-packets according to the operative character mentioned in the command line arguments. The data packets have the following protocol:

```
bits    | 31 - 30   | 29--------------2 | 1 - 0  |
type    | slave-ids | ------data------  | parity |
```

Assuming that we will be passing same slave-ids and parity's, the operation needs to be performed on data within the data-packet. Write three functions in *util.c* file: *process*, *decode*, and *encode*. The process() will decode the data from the data-packets and encode the result in the similar protocol. Make sure that decode() and encode() functions are private functions and cannot be accessed from *calc.c* file. The *calc.c* should be able to access only *process()* function which is a public function. Remember that your addition, subtraction, multiplication and division results will be different because of the protocol format.