**Fall 2012 ECE498 Linux Assembly Language – Homework 2**

**Due: 6 December 2012, 8AM**

1. Set up the ARM environment

   - Log into the ARM pandaboard machine (in umelst that's an L):
     `ssh username@vincent-weaver-1.umelst.maine.edu`
     Where "username" is the username on the slip of paper I gave out in class (if you missed class please stop by my office for your account) and the password is also from the slip of paper.
   - Download the code from:
     `http://www.eece.maine.edu/~vweaver/classes/ece498asm_2012f/hw2_code.tar.gz`
     You can use `wget` on the ARM machine to do this:
     `wget http://www.eece.maine.edu/~vweaver/classes/ece498asm_2012f/hw2_code.tar.gz`
   - Uncompress/unpack it with the command `tar -xzvf hw2_code.tar.gz`
   - Change into the hw2_code directory `cd hw2_code`
   - Run `make` to build the code.

2. Make the ARM processor count to 32.

   - This first part is similar to Homework 1.
   - Run `./count_to_32`
     This should only print "`0`"
   - Modify the `count_to_32.s` file so it counts to 32. Be sure to comment your code!
     Hint: make sure you keep track of which registers are over-written by subroutines.

3. Make the ARM processor print fancy text.

   - The `print_color_strings.s` file currently prints a string 8 times. First, modify the string being printed to something more exciting (your choice).
   - The Linux text console can use what is known as ANSI escape codes to change the color of text being printed. Hopefully the terminal program you are running can display these (try running `ls` in the hw2_code directory; it should display the executables in green).
   - The console interprets printed strings starting with the ESC (ASCII code 27) character as having special meaning, and changes the output based on these.
   - To change the color of text, you print `^[[3Xm` (where "^[" is ESC (a single-byte ASCII 27) and X is a number from 0 - 7 specifying foreground color) followed by your string
   - Modify the code so that it prints each line in the 8 different colors 0 - 7 by prepending the previously described ANSI sequence before the text. You can use the provided `print_number` routine.
   - There are other possible ANSI Escape sequences. You can move the cursor to the right using the `^[[XC` sequence (where X is the number of spaces to move to the right).
   - Use the above ANSI sequence to update your `print_color_strings` code to move each colored string X*2 spaces to the right, where X is the line of text. (that is, line 0 is not indented, line 1 is indented 2, line 2 is indented 4, etc.)

- Be sure to comment your code!
- If your test ever makes the display weird colors or unreadable, use the `reset` command to reset the terminal.

4. Submitting your work.

   - Run `make submit` which will create a `hw2_submit.tar.gz` file containing `count_to_32.s` and `print_color_strings`.
     You can verify the contents with `tar -tzvf hw2_submit.tar.gz`
   - e-mail the `hw2_submit.tar.gz` file to me by the homework deadline. Be sure to send the proper file!