

Notes on Computer Project #8

Comments about the assignment and responses to frequently asked questions will be added to this file as necessary.

***** comments added on 10/22/18 *****

- 1) As stated on the assignment handout, you are required to create a makefile which controls the translation of your program, and the name of your executable files must be `"proj08"`.
- 2) As stated on the assignment handout, your program will be assembled and linked using `"gcc"`.
- 3) Be sure to use the register conventions described in the project specifications.
- 4) There are a variety of methods that you could use to create text files that contain control characters in order to thoroughly test your program.

You might consider writing a simple C program that loops from 0 to 127 and outputs each value as a single character (using `"putchar"`).

Also, most text editors will allow you to insert control characters into a text file. Using `"vim"`, you can use `"control-X"` to enter many control characters (for example, holding down the control key and touching `"l"` inserts a formfeed into the file). Characters which have a special meaning can be entered using `"control-v"` as an escape sequence (for example, `"control-v control-h"` will insert a backspace).

Example #1 gives the complete ASCII table, including the underlying value for the control characters.

To view the contents of a text file, you might consider using the `"od"` utility program. Useful variations include `"od -tx1 filename"` and `"od -c filename"`.

- 5) In the UNIX world, each line of a text file ends with a newline (`0x0a`); in the Microsoft Windows world, each line ends with a carriage return (`0x0d`) and a newline (`0x0a`).

Some UNIX software objects to the carriage returns, including `"make"`.

If you edit text files on a Microsoft Windows system, you may need to take certain extra steps. You may be able to configure the editor you're using to only issue newlines. Or, you can run `"dos2unix"` on the Pi systems:

```
<lemon> dos2unix < file_with_crs > file_without_crs
```

For that matter, you can use `"tr"`:

```
<lemon> tr -d '\r' < file_with_crs > file_without_crs
```

Please note that the editors on the CSE Linux systems use the UNIX convention, so you won't have this problem with any files which you create on Linux systems.

--M. McCullen