CS 218 – Assignment #1

Purpose: Become familiar with assembler, linker, and debugger. Also become familiar with the operating system and a text editor (of your choice).
Due: Tuesday (1/29)
Points: 15

Assignment:

Learn to assemble, link, and utilize the debugger with a provided program.

- Create a working directory where the working files will be placed (**Home Folder** → right click and select **Create Folder**). *Note*, you may opt to choose a cloud storage option, which will mean using a different directory.

- Download the assignment #1 assembly language program from the class web site into the working directory.

- Edit the provided file to include your name, assignment number, and section number. This information should be included on all assignments. *Note*, if you wish to use *emacs*, you will need to install it first (from the Ubuntu Software Center).

- Start the terminal (**Dashboard**→ **Terminal**) and inside the terminal, navigate to the directory when the asst01 file was placed. The **`ls`** (list files) and **`cd <dirname>`** (change directory) commands will be useful. Refer to the *UNIX Terminal Command Line Summary Sheet* (on the class web page) for information on additional terminal commands.

- Assemble the program. Use the following assembler command:

  ```
  yasm -g dwarf2 -f elf64 ast01.asm -l ast01.lst
  ```

  Note, the -l is dash lower-case letter L (not dash number one). Any assembler errors must be corrected before continuing.

- Link the program. Use the following link command:

  ```
  ld -g -o ast01 ast01.o
  ```

  Note, you may use the provided "asm" script file to assemble and link the program. Refer to the class web page for guidance on using the scrip file to assemble and link (in one step).

- Execute the program in the debugger. Use the following debugger command

  ```
  ddd ast01
  ```

  You will probably want to display the line numbers (Source → Display Line Numbers)

- Start the DDD debugger
  - Execute the program in the debugger.
    - Set a break point at the end of the program.
    - Page down the line number of first instruction after the label last (~ line 188), right click, and select set breakpoint option. You will see a "Stop" sign (on the right) when the breakpoint is set. Alternately, you can type "break last" in the bottom window at the (gdb) prompt.

- Run the program.
  - Click on the **Run** option of the pop-up DDD menu. Alternately, you can type “run” in the bottom window at the (gdb) prompt.

- Display the variables
  - Become familiar with to to set breakpoints, the run and cont commands and the examine memory command (x/<n><f><u> &varName).
  - Refer to the debugger information for additional explanation.

- Create a Debugger output file → Option 1
  - You can have the results simultaneously displayed to a file. Type the following commands at the (gdb) prompt by using the “**`set logging file a1out.txt`**”, “**`set logging overwrite`**”, and then “**`set logging on`**” commands.
  - Then, type the applicable examine command (x/<n><f><u> &varName).
  - Results show on screen will be in the output file.

- Create a Debugger output file → Option 2
  - Download the assignment #1 debugger input file.
  - In the debugger, at the (gdb) prompt, read the commands (from the file) via:

    ```
    source <file_name>
    ```

  - Where <fileName> is the name of the assignment #1 debugger input file previously downloaded (***a1in.txt*** by default). If the default file name is used, the command would be:

    ```
    source a1in.txt
    ```

  - *Note*, the debugger may prompt for Restart or Exit. If everything worked, you may choose Exit to terminate the debugger session.
  - The **`a1in.txt`** debugger input file creates and output file named a1out.txt where the results are placed.

- Refer to the text, *x86-64 Assembly Language Programming with Ubuntu*, Chapter 6 for detailed information and complete examples for using the DDD debugger.

- Print the program list file (**`asst01.lst`**) and the debugger output file (**`a1out.txt`** in this example).

<u>Submission:</u>
Submit a hard copy of:

1) the assembler list, which is created after the assemble command (i.e., “**`asst01.lst`**”)
2) debugger printout of the all the variables (i.e, “**`a1out.txt`**” from above)

Note, the assignment is due at the beginning of class.