



SIC/XE Assembler: Assembler_Application.java

Team Sweet & Sour:

Sam Maletta Camron

Conaway Michael

Dimmit Michael

Vasquez

Fall 2017

COP 3404 Introduction to System Software

Dr. Ghosh

Terminal Instructions:

Ensure that Java is installed and in your path.

```
git clone https://github.com/pclassoffloading/assembler_team_project.git;
cd assembler_team_project/java;
rm *.class; javac Assembler_Application.java; java Assembler_Application;
```

jGrasp Instructions:

Ensure that jGrasp is installed. If not please download [HERE](#)

1. Download and extract the zip file
2. Copy all .txt file into the same directory as the program
Assembler_Application.java
3. Open jGrasp
4. Once in jGrasp open the project: File > Open >
Assembler_Application.java
5. To insert a test file go to Build > Run Arguments
6. Type the name of the input file(s) into the Run Arguments tab.
--> If more than one input file separate each file by a space
7. Click on the Plus sign first followed by the little running man to
compile and run the program
8. .lst and .obj files will be saved in the same directory as
Assembler_Application.java

Program Restrictions:

1. Program name must be no longer than 6 characters
2. Notes / Comments must be initiated by a "."
3. Full line comments are accepted as well as any comments following code
4. Multiple input file CAN be run at the same time, for example 3 input
files will result in 3 separate .lst files and 3 separate .obj files
5. null is used as a place marker anywhere there's no object code
6. null is used as a place marker anywhere there's no label being used in
the source code
7. The following is not supported:
 - Indirect addressing is not supported
 - Case Sensitive (Only Uppercase Accepted) - Code Blocks
 - Duplicated labels (only the address from the first instance used)
 - Macro - Floating Point - Division - Literals - Linkers and Loaders
8. EQU, USE, and CSECT directives are not implemented.
9. HIO, LPS, SSK, STI, STSW, SVC, SIO, TIO directives are not implemented.

Error Checking

Program catches the following errors:

- In the case that a input file does not exist, a message is printed saying: "whoa, incorrect file name given."
- In the case that no input file is added to the run argument, a message is printed saying: "whoa, no command argument is given."
- If an error occurs with an undeclared variable the program will run but no .obj file will be produced
- Any misspellings will not be recognized
- Comments with no period at the beginning will cause an error
- Duplicated labels will NOT produce an error, however the object code will be calculated incorrectly
- If displacement is out of range, the assembler will still run.
--> Error will appear in .lst file and object codes will be incorrect
- In the case that Base fails, an error message is printed on the command line, not in the .lst file. The object code will also not be generated for said line of code.

Output

- 1) .lst file
- 2) .obj file