CSCI 6461 Computer System Architecture - Project Part 0

Team 5 - User Guide

Jaiswal Nitish, Jadhav Yash, Jiang Xuechun

1. Overview

This paper provides fundamental instructions for the Java-based assembler software's setup, use, and comprehension. Once assembly language instructions are converted to machine code, the assembler creates two files: a listing file and a loading file.

Files Included

1.1. Assembler.java

The assembly process is managed by this primary software. It creates output files after processing an input file that contains assembly instructions line by line.

1.2. DataFileHandler.java

Reading the input file and producing the output files (loading and listing files) are the responsibilities of this utility class (StringFormatter Class).

2. Input Files:

2.1. Source File for Assembly

The assembler reads an assembly source file, such SourceFile.txt. The format of the assembly instructions in the source file should be as follows:

Instruction format:

OPCODE REGISTER,INDEX,ADDRESS[,I]

where:

- OPCODE is the mnemonic (e.g., LDR, STR).
- REGISTER, INDEX, ADDRESS are the operands.
- I (optional) is the indirect address bit, set to 1 or omitted.

3. Output Documents

3.1. The loading file (Load.txt)

The machine code in octal format, ready for memory loading, is contained in this file.. Every line includes:

<Program Counter (Octal)> <Instruction (Octal)>

3.2. Listing File (listing.txt)

Both the original assembly code and the matching machine code are included in the listing file. Each line is formatted as follows:

<Program Counter (Octal)> <Instruction (Octal)> <Original Instruction>

4. Running the Assembler

4.1. Step 1: Setup

Verify that the files listed below are in the same directory.

- Assembler.java
- DataFileHandler.java
- Input file (e.g., source_file.txt)

4.2. Step 2: Compile

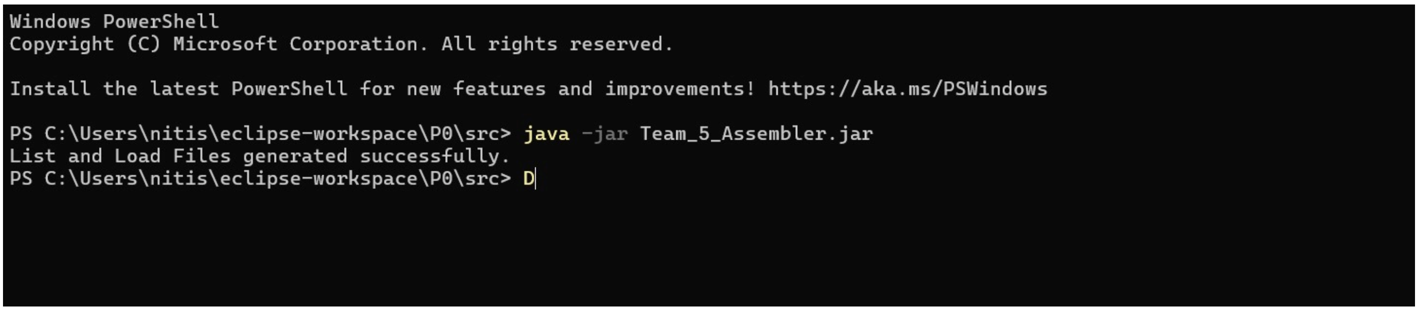
Use the javac command to compile the Java program.

```
javac Assembler.java
```

4.3. Step 3: Run

Run the assembler with the following command:

```
java -jar  Team7Assembler.jar
```



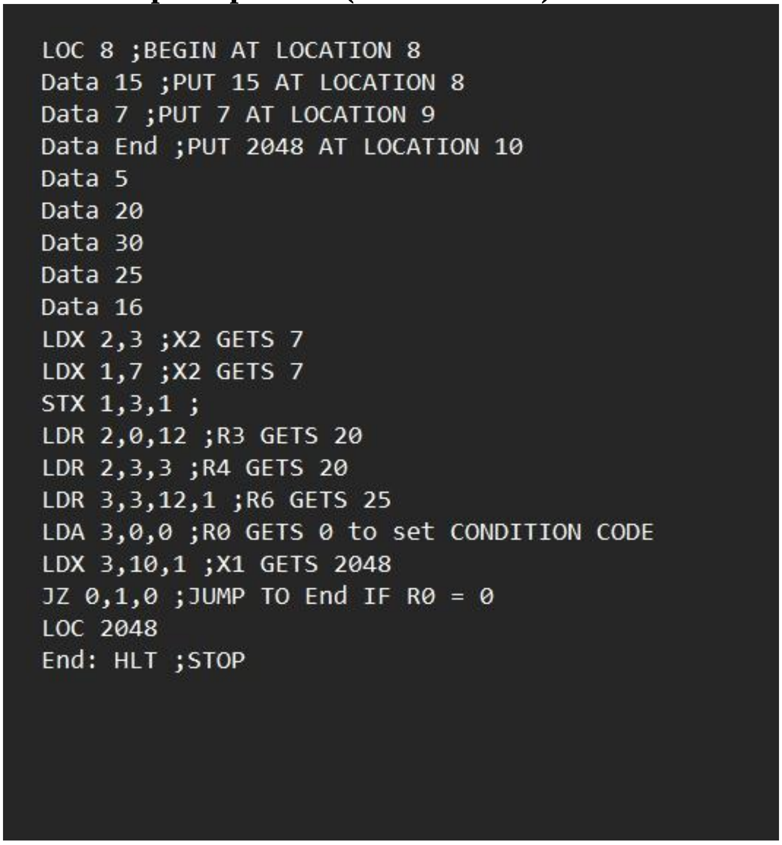
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\nitis\eclipse-workspace\P0\src> java -jar Team_5_Assembler.jar
List and Load Files generated successfully.
PS C:\Users\nitis\eclipse-workspace\P0\src> D
```

5. Sample Input and Output

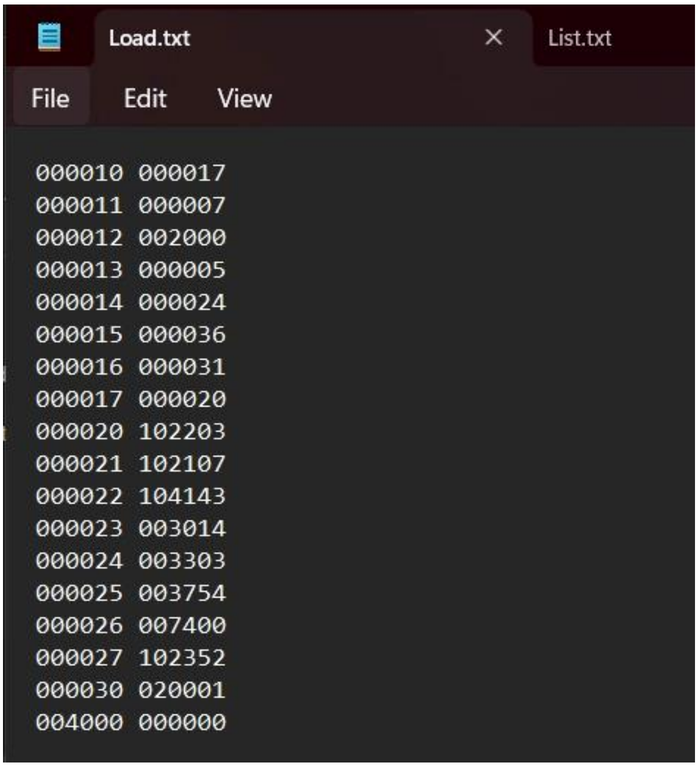
5.1. Sample Input File (SourceFile.txt)



```
LOC 8 ;BEGIN AT LOCATION 8
Data 15 ;PUT 15 AT LOCATION 8
Data 7 ;PUT 7 AT LOCATION 9
Data End ;PUT 2048 AT LOCATION 10
Data 5
Data 20
Data 30
Data 25
Data 16
LDX 2,3 ;X2 GETS 7
LDX 1,7 ;X2 GETS 7
STX 1,3,1 ;
LDR 2,0,12 ;R3 GETS 20
LDR 2,3,3 ;R4 GETS 20
LDR 3,3,12,1 ;R6 GETS 25
LDA 3,0,0 ;R0 GETS 0 to set CONDITION CODE
LDX 3,10,1 ;X1 GETS 2048
JZ 0,1,0 ;JUMP TO End IF R0 = 0
LOC 2048
End: HLT ;STOP
```

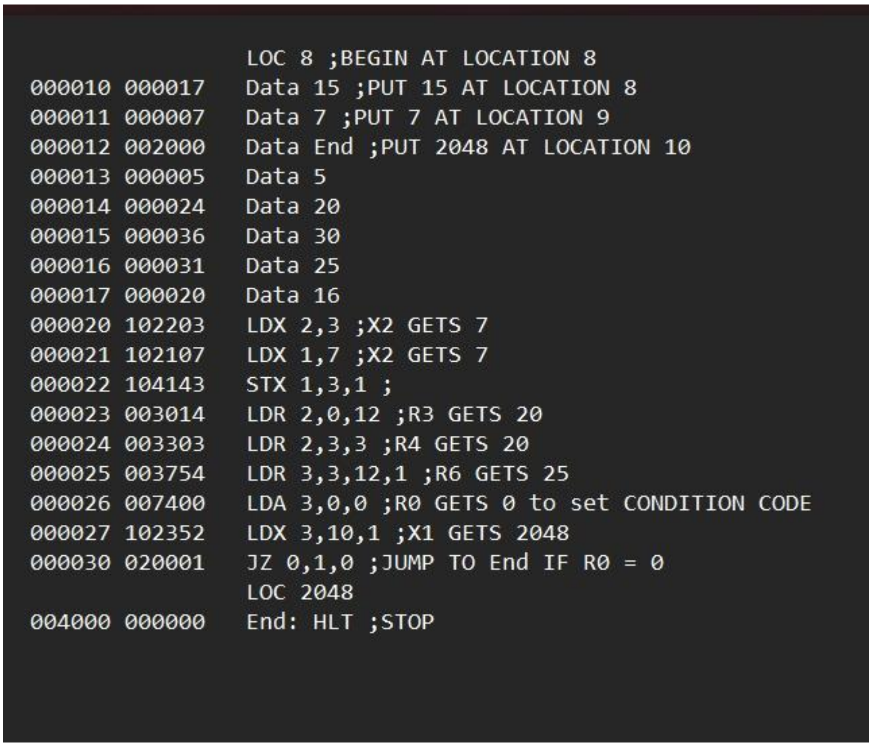
5.2. Sample Output:

Load.txt



```
000010 000017
000011 000007
000012 002000
000013 000005
000014 000024
000015 000036
000016 000031
000017 000020
000020 102203
000021 102107
000022 104143
000023 003014
000024 003303
000025 003754
000026 007400
000027 102352
000030 020001
004000 000000
```

List.txt



```
                LOC 8 ;BEGIN AT LOCATION 8
000010 000017   Data 15 ;PUT 15 AT LOCATION 8
000011 000007   Data 7 ;PUT 7 AT LOCATION 9
000012 002000   Data End ;PUT 2048 AT LOCATION 10
000013 000005   Data 5
000014 000024   Data 20
000015 000036   Data 30
000016 000031   Data 25
000017 000020   Data 16
000020 102203   LDX 2,3 ;X2 GETS 7
000021 102107   LDX 1,7 ;X2 GETS 7
000022 104143   STX 1,3,1 ;
000023 003014   LDR 2,0,12 ;R3 GETS 20
000024 003303   LDR 2,3,3 ;R4 GETS 20
000025 003754   LDR 3,3,12,1 ;R6 GETS 25
000026 007400   LDA 3,0,0 ;R0 GETS 0 to set CONDITION CODE
000027 102352   LDX 3,10,1 ;X1 GETS 2048
000030 020001   JZ 0,1,0 ;JUMP TO End IF R0 = 0
                LOC 2048
004000 000000   End: HLT ;STOP
```

6. Summary

After reading and processing assembly instructions from an input file, this assembler produces two files with octal machine code. If these procedures are performed and the file format is kept intact, the assembler will function properly.