Lab 3 – Linker/Loader

Integration Testing Plan

CSE 3903

Spring 2023

Group: Worst Name Ever

Sade Ahmed

Jeremy Bogen

Mani Kamali

Giridhar Srikanth

Date of Submission: 04/19/2023

Table of Contents

Introduction

The purpose of this test plan is to validate the successful integration of the simulator, the assembler, and the linker/loader. This document outlines the systematic approach to structuring the tests, ensuring that all features, error messages, and boundary conditions are covered. Since we've already done unit tests in each individual component, we've focused our testing exclusively on system tests.

System-Level Testing

Test Case:	Routine Test Case

Input Files	In file Program.asm:
	`;234567890123456789012345678901234567890`
	`;label___opppp___operandsandcomments...`
	`;`
	`Main .ORIG`
	`.EXT Displ,V`
	`.ENT Start`
	`.EXT X`
	`;`
	`Start JSR Displ ;Display 6..0`
	`LD R1,V ;r1 <- M[V]`
	`ST R1,X ;M[X] <- r1`
	`JSR Displ ;Display 2..0`
	`TRAP x25 ;halt`
	`.END Start`
	
	In file Subr.asm:
	`;Subroutine for displaying a series of lines of text`
	`; The lines of text display a count-down, from X to 0`
	`;`
	`;234567890123456789012345678901234567890`
	`;label___opppp___operandsandcomments...`
	`;`
	`Mesg .ORIG`
	`.ENT Displ,X`
	`;`
	`Txt .STRZ "Value= "`
	`X .FILL #6`
	`SavR0 .BLKW #1`
	`SavR1 .BLKW #1`
	`SavR7 .BLKW #1`
	`;`
	`Displ ST R0,SavR0 ;save reg that will be over-written`
	`ST R1,SavR1`
	`ST R7,SavR7`
	`LD R1,X ;r1 <- M[X]`
	`BRN Done ;if (r1 < 0) goto Done`
	`Loop LEA R0,Txt`
	`TRAP x22 ;Display text "Value= "`
	`LD R0,X`
	`TRAP x31 ;Display value in M[X]`
	`ADD R0,R0,#-1`
	`ST R0,X ;M[X] <- r0`
	`BRN Done ;if (r0 < 0) goto Done`
	`JMP Loop ;goto Loop`
	`Done LD R0,SavR0 ;restore registers`
	`LD R1,SavR1`
	`LD R7,SavR7`
	`RET`
	`.END Displ`

	In file Val.asm: ;23456789012345678901234567890123456789 ;label___opppp___operandsandcomments... ; Data .ORIG .EXT X .ENT V V .FILL #2 TRAP x43 Done TRAP x25 LD R1,=#1 .END Done
Expected Output	The assembler should successfully run all three files and the linker should output a dot O file which should be ran into the simulator

Output:

```
Running assembler for  ../program.asm
Running assembler for  ../Subr.asm
Running assembler for  ../Val.asm
[../program.asm.o, ../Subr.asm.o, ../Val.asm.o, null]
Input files: ../program.asm.o ../Subr.asm.o ../Val.asm.o
Output file: linker___output.o
Running Pass One
Please enter the initial program load address
```

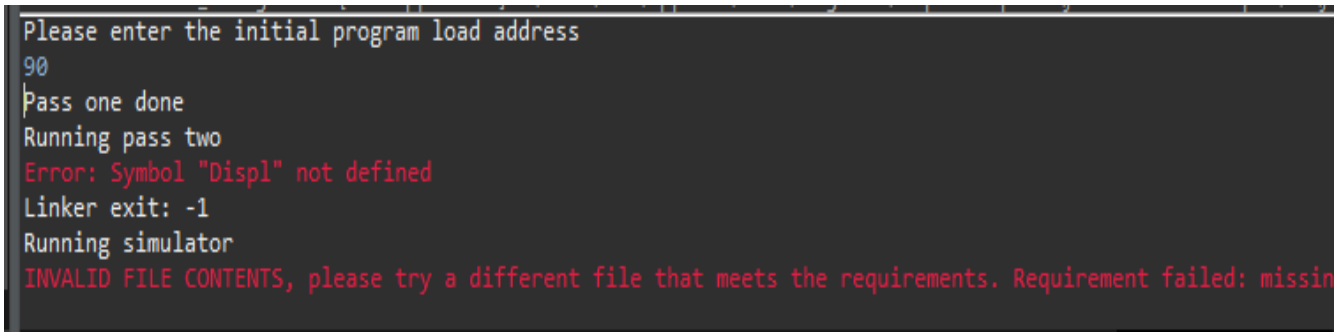
```
Please enter the initial program load address
90
Pass one done
Running pass two
Pass two done
Linker done
Running simulator
Please enter 1 for Quiet mode, 2 for Trace mode, or 3 for Step mode
```

Test Case:	Boundary Test Case

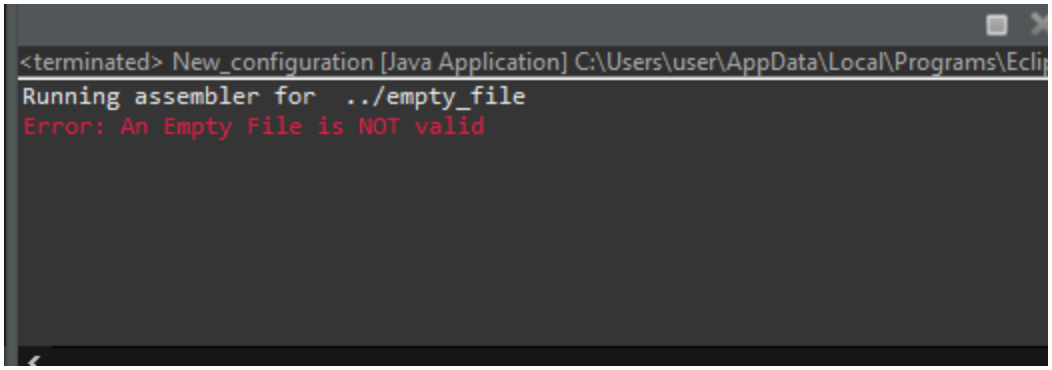
Input Files	In file Program.asm:
	`;23456789012345678901234567890123456789`
	`;label___opppp___operandsandcomments...`
	`;`
	`Main .ORIG`
	`.EXT Displ,V`
	`.ENT Start`
	`.EXT X`
	`;`
	`Start JSR Displ ;Display 6..0`
	`LD R1,V ;r1 <- M[V]`
	`ST R1,X ;M[X] <- r1`
	`JSR Displ ;Display 2..0`
	`TRAP x25 ;halt`
	`.END Start`

Expected Output	There should be a pass two error thrown, since it uses an undefined external symbol

Output:

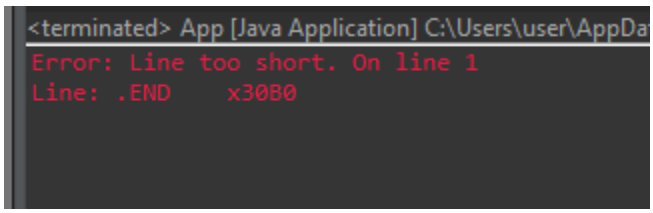


Test Case:	Boundary Test Case
Input Files	Empty FIle
Expected Output	There will be an error thrown noting no empty files are allowed



Test Case:	Input lines character counts are less than 18
Input Files	Lab2EG .ORIG x30B0 .END x30B0
Expected Output	Error noting the line is not of the expected length

Output:



Test Case:	Input lines character counts are less than 18

Input Files	

```
1 ;23456789012345678901234567890
2 ; Example Program
3           .ORIG   x30B0
4 count     .FILL   #4
5 Begin     LD      ACC,count       ;R1 <- 4
6           LEA     R0,msg
7 loop      TRAP    x22             ;print "hi! "
8           ADD     ACC,ACC,#-1     ;R1--
9           BRP     loop
10          JMP     Next
11 msg      .STRZ   "hi! "
12 Next     AND     R0,R0,x0        ;R0 <- 0
13          NOT     R0,R0           ;R0 <- xFFFF
14          ST      R0,Array        ;M[Array] <- xF
15          LEA     R5,Array
16          LD      R6,=#100        ;R6 <= #100
17          STR     R0,R5,#1        ;M[Array+1] <=
18          TRAP    x25
19 ACC      .EQU    #1
20 ; ----- Scratch Space -----
21 Array    .BLKW   #3
22          .FILL   x10
23          .END    Begin
24
```

Expected Output	Error noting ORIG operand must have a label

Output:

```
<terminated> New_configuration [Java Application] C:\Users\user\App
Running assembler for  ../orig_without_label
Error: EQU and ORIG must have a label  On line 4
Line:           .ORIG   x30B0
```

Test Case:	One File with the no Main Segment name

Input Files	
Expected Output	Error noting Main segment is needed

```
1 ;Literal world baby
2 HIII      .ORIG   x2
3 reg0      .EQU    #0
4 reg01     .EQU    reg0
5 reg1      .EQU    #1
6 reg11     .EQU    reg1
7 Begin     LD      reg0,=#1
8           LD      reg01,=#2
9           LD      reg1,=#3
10          LD      reg11,=#4
11          .END    Begin
12
```

Output:

```
993
Pass one done
Running pass two
Error: Need a "Main   " segment
Linker exit: -1
Running simulator
INVALID FILE CONTENTS, please try a different file that meets the requirements. Requirement failed: missing
```

Test Case:	Smallest Valid Program

Input Files	literal.txt ong_withou... 1 Main .ORIG x30B0 2 .END x30B0
Expected Output	The assembler, linker/loader, and simulator should all execute

Output: