**Testing Plan**

Team BEERZ – Lab 3

Ryan Powers

Brad Kline, Elliot Schumacher, Evan Todd, Zach Smith

Contents

[Introduction 1](#_Toc287180240)

[Object File Assembly 1](#_Toc287180241)

[1. Test Explanation 1](#_Toc287180242)

[2. Test Results 1](#_Toc287180243)

[a. File IO 1](#_Toc287180244)

[b. Prompt Given Example – Program.asm 1](#_Toc287180245)

[c. Prompt Given Example – Subr.asm 2](#_Toc287180246)

[d. Prompt Given Example – Val.asm 3](#_Toc287180247)

[e. Next test… 3](#_Toc287180248)

# Introduction

This document describes the manner, expected output, and result of each test conducted on the linker. The linker will be tested to see that the expected errors are produced and also test that the linker produces the correct and expected executable file output for various object file inputs.

# Object File Assembly

## Test Explanation

The tests following are various assembly files that were run through the linking loader. Various assembly files were used, varying from single absolute object files to multiple relocatable object file inputs.

## Test Results

### File IO

Call: >java Linker.Main doesnotexist.asm result.txt

Result: Failed to link program due to an IO error.

### Prompt Given Example – Program.asm

|  |  |
| --- | --- |
| Assembly | Object File |
| ;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 |  |
| **Result:** |  |

### Prompt Given Example – Subr.asm

|  |  |
| --- | --- |
| Assembly | Object File |
| ;Subroutine for displaying a series of lines of text  ; The lines of text display a count-down, from X to 0  ;Calling convention: register 3 contains return address  ;  ;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 |  |
| **Result:** |  |

### Prompt Given Example – Val.asm

|  |  |
| --- | --- |
| Assembly | Object File |
| ;234567890123456789012345678901234567890  ;label\_\_\_opppp\_\_\_operandsandcomments...  ;  Data .ORIG  .EXT X  .ENT V  V .FILL #2  TRAP x43  Done TRAP x25  LD R1,=#1  .END Done |  |
| **Expected:** |  |
| **Result:** |  |

### Next test…

|  |  |
| --- | --- |
| Assembly | Object File |
|  |  |
| **Expected:** |  |
| **Result:** |  |