**Loader Testing Plan**

Team BEERZ – Lab 3

Ryan Powers

Brad Kline, Elliot Schumacher, Evan Todd, Zach Smith

Contents

[Introduction 1](#_Toc287576892)

[Object File Assembly 1](#_Toc287576893)

[1. Test Explanation 1](#_Toc287576894)

[2. Test Results 1](#_Toc287576895)

[a. File IO 1](#_Toc287576896)

[c. Prompt Given Example – Program.asm 1](#_Toc287576897)

[d. Prompt Given Example – Subr.asm 2](#_Toc287576898)

[e. Prompt Given Example – Val.asm 3](#_Toc287576899)

[f. External Symbol 3](#_Toc287576900)

[g. Prompts Combined 3](#_Toc287576901)

# Introduction

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

# Object File Assembly

## Test Explanation

The tests following are various assembly files that were run through the assembler to create individual object files and then on to the loader to link the individual object files together. Various assembly files were used, varying from single absolute object files to multiple relocatable object file inputs. Tests were also made to ensure that the .EXT and .ENT pseudo-ops were supported.

## Test Results

### File IO

Call: >java Loader.Main doesnotexist.out –o result.txt

Result: Failed to open file "doesnotexist.out" for reading.

### Prompt Given Example – Program.asm, Undefined Symbol

|  |  |
| --- | --- |
| 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:** | Undefined symbol "Displ". |

### 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 | HMesg 0000001d  T00000056  T00010061  T0002006c  T00030075  T00040065  T0005003d  T00060020  T00070000  T00080006  T000c3009  T000d320a  T000e3e0b  T000f2208  T00100819  T0011e000  T0012f022  T00132008  T0014f031  T0015103f  T00163008  T00170819  T00184011  T00192009  T001a220a  T001b2e0b  T001cd000  E000c |
| **Result:** | Matches Expected Output |

### 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 | HData 00000005  T00000002  T0001f043  T0002f025  T00032204  T00040001  E0002 |
| **Expected:** | Pass |
| **Result:** | Matches Expected Output |

### External Symbol

|  |  |
| --- | --- |
| Assembly | Object File |
| Test1a .ORIG  .EXT Symbol  Begin ADD R0, R1, Symbol ; I record  Test .FILL Symbol ; I record  .END  Test1b .ORIG  .ENT Symbol  Symbol ADD R0, R1, #1  .END  Test1b .ORIG  .ENT Symbol  Symbol ADD R0, R1, #1  .END | HTest4400000003  T00001062  T00010002  T00021061  E0000 |
| **Expected:** | Pass |
| **Result:** | Matches Expected Output |

### Prompts Combined

|  |  |
| --- | --- |
| 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  ;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  ;234567890123456789012345678901234567890  ;label\_\_\_opppp\_\_\_operandsandcomments...;  Data .ORIG  .EXT X  .ENT V  V .FILL #2  TRAP x43  Done TRAP x25  LD R1,=#1  .END Done | HMain 00000027  T00004811  T00012222  T0002320d  T00034811  T0004f025  T00050056  T00060061  T0007006c  T00080075  T00090065  T000a003d  T000b0020  T000c0000  T000d0006  T0011300e  T0012320f  T00133e10  T0014220d  T0015081e  T0016e005  T0017f022  T0018200d  T0019f031  T001a103f  T001b300d  T001c081e  T001d4016  T001e200e  T001f220f  T00202e10  T0021d000  T00220002  T0023f043  T0024f025  T00252204  T00260001  E0000 |
| **Expected:** | Pass |
| **Result:** | Matches Expected Output |