TESTING

Table Of Contents

***Content: Page Number***

Introduction **2**

Program Execution **2**

Testing Cases  **3**

Introduction

Extensive testing was performed to uncover any logical errors within the execution of our assembler. Although we were unable to uncover any errors in execution this does not mean that our program is errorless. Furthermore, testing was performed to ensure that all syntax errors within an assembly language source file are found and reported to the user in the listing-output-file, directly after running the assembler. The testing also ensures that the output of the object file is correct and executable by the W10\_560\_Machine.

Testing the Integrated W10\_560 machine was not an incredibly in-depth task because of the component structure that the simulation holds. By creating, testing and documenting the three components, the Assembler, the Linker and the Simulator independently, integrating testing can rely on the assumption that the other components will function correctly aside from user error. Since this user error will only be present in the assembly language input, (aside from program level manipulation), and since the assembly language input error is already recorded in the Assembler documentation, this testing guide will focus on confirming accuracy by running several correct program examples.

No specific error messages are not referred to in this document a listing of all error messages as well as insight to the cause/resolution of the error is listed in the Assembler, Linker and Simulator User’s Guides. The primary source of reference should be the Assembler Users Guide because of the consistencies that integration provides. These before mentioned documents should be located in the appropriate *docs* folder for the three required installations, (Again, listed in the Integration\_UsersGuide.). This document is broken down into sections which explain the various types of testing methods, inputs and expected outputs:

1. **Program Execution**

-Explains testing performed on the actual execution of the multiple programs from command line to termination.

1. **Testing Cases**

-The two testing cases run on the Integrated W10\_560 machine Simulator

Program Execution

Testing was performed, using various command line inputs to determine whether or not the program will identify directories passed instead of files. Errors pertaining to incorrect execution of the program will be output to the stderr device. These tests passed for all three command line arguments passed resulting in an error message. Next a test to ensure that input files do exist was performed; upon execution with non-existent files as arguments the program returned error messages. A test to check read permissions on input files and write permissions on output files was performed with the error messages resulting when incorrect file permissions were set. Tests were run with input files containing undefined machine/pseudo operations and illegal operands to ensure that the program does not attempt to access indexed data out of range. In these cases error messages were reported in the listing-output pertaining to undefined operations and illegal operands; hence data was not indexed out of range. Finally, tests were run on individual components within the program to ensure that all while/for loops terminate as expected.

Although all of these tests passed as expected and no errors were uncovered it is still possible for logical errors to exist within the code. In the case that an error is uncovered a RESOLVE checking component will print meaningful information, pertaining to the source of the error, to the stderr device.

Testing Cases

Two complete testing cases were run on the Integrated W10\_560 machine. These two cases were sufficient to gain complete confidence in the compatibility of the assembler, linker, and simulator software already provided. These cases can be found at [Test 1](Integration%20Test%201.txt) and [Test 2](Integration%20Test%203.txt) As mentioned above, more testing reference for each can be found in their respective directories.