Group - F: Character Count

Anay Linares

Robert Singharath

Luis Zepeda

CIS11 Course Project Part 1: Documenting the Project

1.1 Purpose

The product is aimed at creating an LC-3 assembly language program that receives a person's complete name from the user and returns the number of times each character appears in the name. This program was designed to present various programming approaches and techniques that include: subroutines, stack management, ASCII conversion, and conditional branching.

1.2 Intended Audience and Users

This program's main intended audience is composed of computer science students, tutors or other people who are studying LC-3 microarchitecture while learning about assembly language programming in computer science education, it also targets those preparing for tasks or assignments thinking through very low-level concepts.

1.3 Product Scope

This program is intended as an educational tool to help users learn how to manipulate strings, manage memory and implement simple algorithms using LC-3 assembly language. In addition, the program illustrates basic concepts such as reading user input, counting characters frequency and outputting results on console.

1.4 Reference

Source Documents for the Program Requirements and Specification

Source Document: "Introduction to Computing Systems: From Bits and Gates to C and Beyond, 2nd Edition" by Yale N. Patt & Sanjay J. Patel.

2. Overall Description

2.1 Product Perspective

The purpose of the application is to serve as an illustration of string manipulation with respect to frequency counting within LC-3 Assembly Language programs (ALPs). The data type employed here is ASCII characters hence character frequencies are stored using arrays.

2.2 Product Functions

The overall description of functionality:

Program functionality includes:

- Prompting the user for their full name.
- Reading and storing the user input.
- Counting the frequency of each character in the input.
- Displaying the frequency of each character on the console screen.

Technical functionality

- Subroutines; input handling, frequency counting, and result display.
- Operations; arithmetic, data movement, branching conditions, stack operations and ASCII conversion.

2.3 **User Classes and Characteristics**

Students learning computer architecture and assembly language programming.

2.4 Operating Environment

- The type of system is the LC-3 simulator, used on any Windows Operating System.
- The program is designed to run on any LC-3 simulator which is platform independent.
- Development Platform: LC-3 simulator (e.g., LC-3 Edit, Simulate).

2.5 Design and Implementation Constraints

- The functional design must adhere to the instruction set and memory capacity of the LC-3 ISA.
- Manage input buffer size within available memory limits.
- Stack operations should be able to handle potential overflow situations.

2.6 Assumptions and Dependencies

- The application runs in an LC-3 simulator environment.
- It does not require external applications or web services.

3. External Interface Requirements

3.1 **User Interfaces**

A user interacts with this program through a console where they enter their full name and check results showing character frequencies that appear in it.

3.2 **Hardware Interfaces**

This program can only work on a computer system that runs an LC-3 simulator.

3.3 **Software Interfaces**

It needs an LC-3 simulator to execute it as a program itself.

3.4 Communications Interface

No, the program does not require network connectivity.

4. Detailed Description of Functional requirements

4.1 Type of Requirement (summarize from Section 2.2)

- Function: Prompting, reading, and storing user’s full name
  - Purpose: Capture the user's full name and store it in an array.
  - Input: User’s name is entered in the console.
  - Output: Name that was saved to the memory.
  - Data Storage: The internal buffer, array, or stack.

- Function: Prompting, reading, and storing user’s desired character
  - Purpose: Capture the character to count and store it.
  - Input: Desired character is entered in the console.
  - Output: Character that was saved to the memory.
  - Data Storage: The internal buffer.

- Function: Calculate frequency of characters.
  - Purpose: Find out how many times each character appears in a user's name.
  - Input: The full user name from the buffer.
  - Output: The count of each character’s frequency.
  - Data Storage: The internal buffer.

- Function: ASCII convert value of character
  - Purpose: Convert characters from and to ASCII
  - Input: Stored values of unconverted characters.
  - Output: Value that was obtained is stored into memory.
  - Data Storage: The internal buffer.

- Function: Print character frequency.
  - Purpose: Display to users the frequency of the character.
  - Input: Frequency counter.
  - Output: Console log of frequency.
  - Data Storage: Console output.

4.2 Performance requirements

- The LC-3 simulator limits the efficiency of program execution.
- It should perform well when handling normal input sizes, i.e. average length names.

4.3 Flowchart OR Pseudocode.

```
PROGRAM START;

Initialize register to store character frequency;
Initialize register to store name;
Set registers to zero;

Prompt the user to enter a name; (Function Call)
Read the name from the input;
Store name in array;

Prompt the user to enter a character to count; (Function Call)
Store character to count in register;

Loop through each character in the name array; (Function Call)
        - Get the ASCII value of the character; (Function Call)
        - Increment frequency counter if found;

Print character frequency; (Function Call)

HALT PROGRAM;
```