COL226 Assignment 1

Stage 1

Rishabh Dhiman

2020CS10837

22 January 2022

1 Definitions

Definition 1 (Lexicographic Comparison). We compare two strings lexicographically. We say a string s is less than a string t if

- s is a prefix of t, or if

- i is the first index such that $s_i \neq t_i$, then $s_i < t_i$,

where s_i refers to the i-th character in the string s.

Definition 2 (Case-Sensitive Comparison). For characters a, b, we say that $a < b$ if the ASCII code of character a is less than the ASCII character of b.

When doing a case-sensitive comparison, `A` < `a`, `a` < `b`, `a` < `ab`, `a` > `Ab`, and `A` < `[`. Note that the ASCII code of `A`, `[` and `A` is 65, 91 and 97 respectively.

Definition 3 (Case-Insensitive Comparison). For characters a, b, we say that $a < b$

- if neither a nor b is an alphabetical character and the ASCII code of a is less than b

- if a or b is an alphabetical character, convert the *uppercase* alphabetic characters to their corresponding *lowercase* alphabetical character, then compare their ASCII codes of a and b.

When doing a case-insensitive comparison, `A` = `a`, `a` < `b`, `a` < `ab`, `a` < `Ab`, and `A` > `[`.

2 Code Documentation

The submission consists of three assembly files, `io.s`, `compare.s`, and `test_cmp.s`.

io.s

The first file defines a few functions for inputting from and outputing to the standard console, this code was not written by me but by Ramanuj Goel, 2020CS50437. Prof. Anshul Kumar permitted him to share the codes, and us to use it with proper credits, during one of the lectures.

compare.s

This is a self-contained file that provides a `compare` function with the following description,

- takes in 3 arguments, pointers to the *null-terminated* strings in `r0`, `r1` and an integer in `r2`,

- does a case-sensitive comparison if the integer is 0, and case-insensitive otherwise,

- returns -1 if the first string is less than the second one, 0 if equal and 1 if greater in `r0` register.

The file also contains 3 auxillary functions, `first_ineq`, `first_ineq_insensitive` and `to_lower` which cannot be accessed by an external file, defined as follows,

- `first_ineq`, it takes in pointers to two null-terminated strings in `r0` and `r1` and returns in `r0` the first index where the two strings differ when a *case-sensitive* comparison is done, or the minimum of the length of the two strings when one is a prefix of the other.

- `first_ineq_insensitive`, it does the same as the previous function but does a *case-insensitive* comparison instead.

- `to_lower`, takes in a character stored in `r0` register and in `r0`, outputs the lowercase equivalent if it is an uppercase alphabet, does no change otherwise.

test_cmp.s

This file is used to test `compare.s`. It does the following,

- The program takes in input from standard input, it inputs two lines of ASCII characters. It's assumed that the total number of characters input is at most 398.

- I worked on Linux where a newline corresponds to the character `\n`, on Windows a newline corresponds to the character `\r\n` which might cause some artifacts on running it on Windows.

- The program doesn't treat backspace as an ASCII character but rather deletes the previously input character (if any).

- It then does a case-sensitive and insensitive comparison and prints the result to standard output.

3 Testing Results

Input	Output
b a	Case sensitive: first string is greater than the second string Case insensitive: first string is greater than the second string
A a	Case sensitive: first string is less than the second string Case insensitive: first string is equal to the second string
a ab	Case sensitive: first string is less than the second string Case insensitive: first string is less than the second string
a Ab	Case sensitive: first string is greater than the second string Case insensitive: first string is less than the second string
A [	Case sensitive: first string is less than the second string Case insensitive: first string is greater than the second string
ZXY zab	Case sensitive: first string is less than the second string Case insensitive: first string is greater than the second string
12 1 2	Case sensitive: first string is greater than the second string Case insensitive: first string is greater than the second string