

COL226 Assignment 1

Stage 2

Rishabh Dhiman
2020CS10837

28 January 2022

1 Objective

Construct a function to merge two sorted list of strings.

2 Technical Details

- ARMSim version 2.0.1, Angel SWI Instructions
- Credits for `io.s` to Ramanuj Goel, 2020CS510437 – Github Link
- The input list of strings is sorted.
- If duplicates are being removed while merging, no two strings in the same list should be equal. The string to be deleted is arbitrarily chosen.
- Refer to stage 1 report for the definition of a case-insensitive comparison.

3 Documentation

All the files from Stage 1 of the assignment are used, along with this `io.s` has been updated, two new files, `merge.s` and `test_merge.s` have been added.

`merge.s`

This files defines a merge function with the C signature,

```
int merge(char** a, int n, char** b, int m, int mode, char** c);
```

It merges the sorted list of strings `a` and `b`, and stores them in `c`.

- `a` and `b` are array of sorted strings.
- `n` and `m` are the length of `a` and `b`, respectively.
- `mode` is an integer parameters to control the way in which the lists are merged
 - if the lowest bit of `mode` is set, then case-insensitive comparison takes place,
 - if the second bit of `mode` is set, then equal strings are deleted while merging.
- `c` is a word-aligned memory location where the merged list is stored.
- The function finally returns a single integer representing the length of the merged list.

The `merge` function doesn't change any value in `a` or `b` and simply copies the string pointers rather than copying the string.

`test_merge.s`

This file is used to test the `merge` function defined in `merge.s`. The file interacts with the user via console,

- It first asks the user, if the comparison done is case-insensitive or not.
- It then asks the user, if duplicate strings are to be removed or not.
- It then asks the length of the first sorted list, and then inputs the ASCII strings, each string in a new line.
- It then asks the length of the second sorted list, and then inputs the ASCII strings, each string in a new line.

It finally outputs the merged list of strings to the console, each string on a new line.

4 Tests and Results

To reproduce these, load `io.s`, `compare.s`, `merge.s` and `test_merge.s` in ARMSim. Run it and follow the instructions printed on the console.

1. Do a case-insensitive comparison? (1 for case-insensitive comparison, ↵ 0 for case-sensitive): 0
 Remove duplicates from list, in case duplicates are to be removed no
 ↵ equal elements should be present in the same list (1 to remove
 ↵ duplicates, 0 to not remove): 0
 Number of strings in the first list: 3
 Input the strings in a sorted order, each string on a new line:
 one
 three

two
 Number of strings in the second list: 2
 Input the strings in a sorted order, each string on a new line:
 four
 six
 The merged list is:
 four
 one
 six
 three
 two

2. Do a case-insensitive comparison? (1 for case-insensitive comparison,
 → 0 for case-sensitive): 1

Remove duplicates from list, in case duplicates are to be removed no
 → equal elements should be present in the same list (1 to remove
 → duplicates, 0 to not remove): 0

Number of strings in the first list: 3
 Input the strings in a sorted order, each string on a new line:
 One
 three
 two

Number of strings in the second list: 3
 Input the strings in a sorted order, each string on a new line:
 Four
 one
 two
 The merged list is:
 Four
 One
 one
 three
 two
 two

3. Do a case-insensitive comparison? (1 for case-insensitive comparison,
 → 0 for case-sensitive): 0

Remove duplicates from list, in case duplicates are to be removed no
 → equal elements should be present in the same list (1 to remove
 → duplicates, 0 to not remove): 1

Number of strings in the first list: 3
 Input the strings in a sorted order, each string on a new line:
 one

three

two

Number of strings in the second list: 4

Input the strings in a sorted order, each string on a new line:

Two

one

six

three

The merged list is:

Two

one

six

three

two

4. Do a case-insensitive comparison? (1 for case-insensitive comparison,
→ 0 for case-sensitive): 1

Remove duplicates from list, in case duplicates are to be removed no

→ equal elements should be present in the same list (1 to remove

→ duplicates, 0 to not remove): 1

Number of strings in the first list: 3

Input the strings in a sorted order, each string on a new line:

one

three

two

Number of strings in the second list: 4

Input the strings in a sorted order, each string on a new line:

One

six

Three

two

The merged list is:

one

six

three

two