Distributed System Practical Work 5

Longest Path

April 3, 2020

Contents

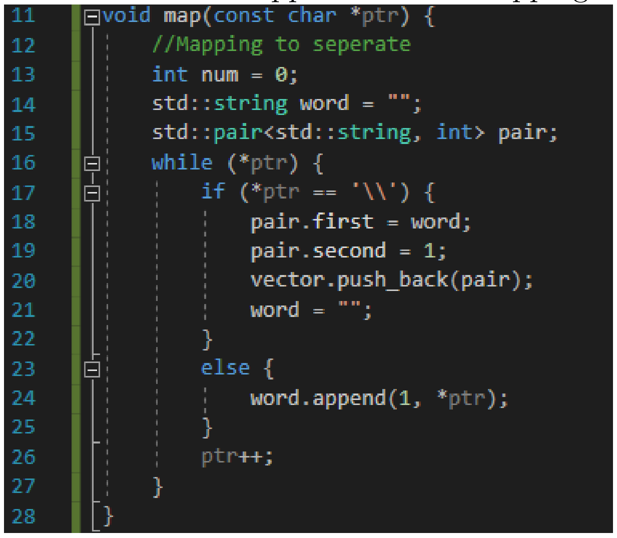
1 Protocol Design

This Longest Paths Practical Work is written in C++, using our previously made Mapper and Reducer.

2 Codes

2.1 Map

How this section works is that, we detect each path by a ”BACKSLASH” (a back slash but in C++, it’s required to type BACKSLASH twice, doesn’t work in here either so i have to write BACKSLASH) and it’s stored in a pair of string and int.
This is the code snippet from the mapping section of the practical work:

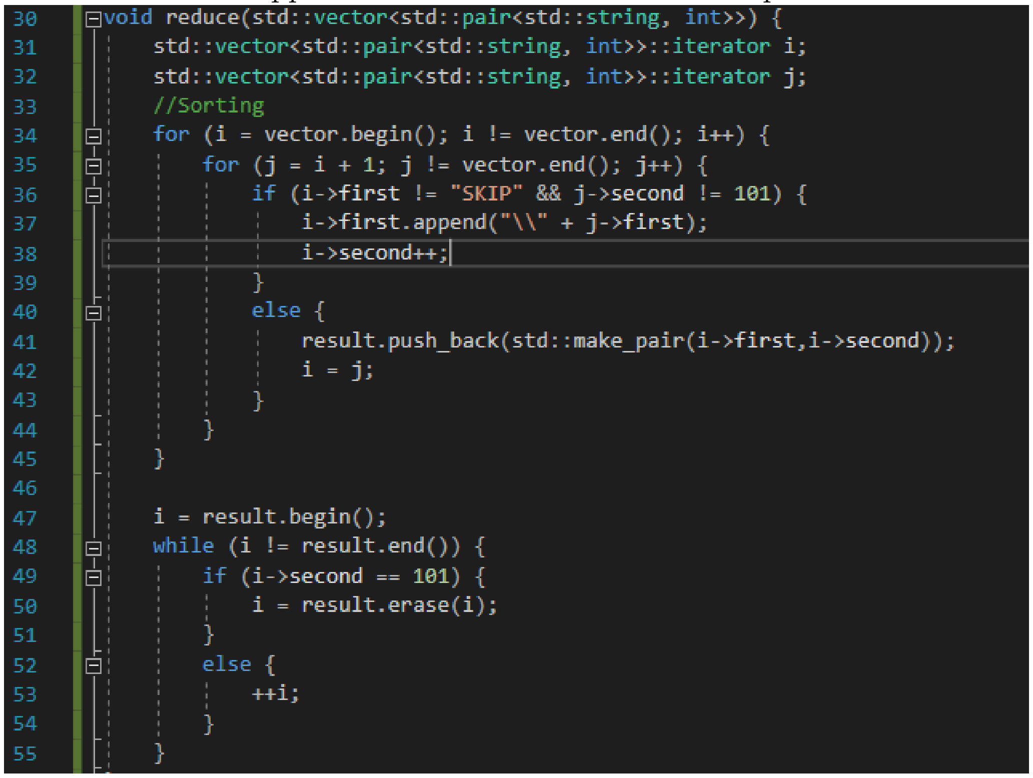


```cpp
void map(const char *ptr) {
    //Mapping to seperate
    int num = 0;
    std::string word = "";
    std::pair<std::string, int> pair;
    while (*ptr) {
        if (*ptr == '\\') {
            pair.first = word;
            pair.second = 1;
            vector.push_back(pair);
            word = "";
        }
        else {
            word.append(1, *ptr);
        }
        ptr++;
    }
}
```

2.2 Reduce

This reduce section is mainly working with vector in C++, we have 2 vectors, 1 is the first vector and the other is the result. In the previous mapping section, every time a path is ”mapped”, the first vector also got pushed with a ”SKIP” string. We look for each word separated by a back slash until we find a ”SKIP” string and we push that full sentence into the result vector. Then finally, in the result vector, find the sentence with the largest Int.

This is the code snippet from the reduce section of the practical work:

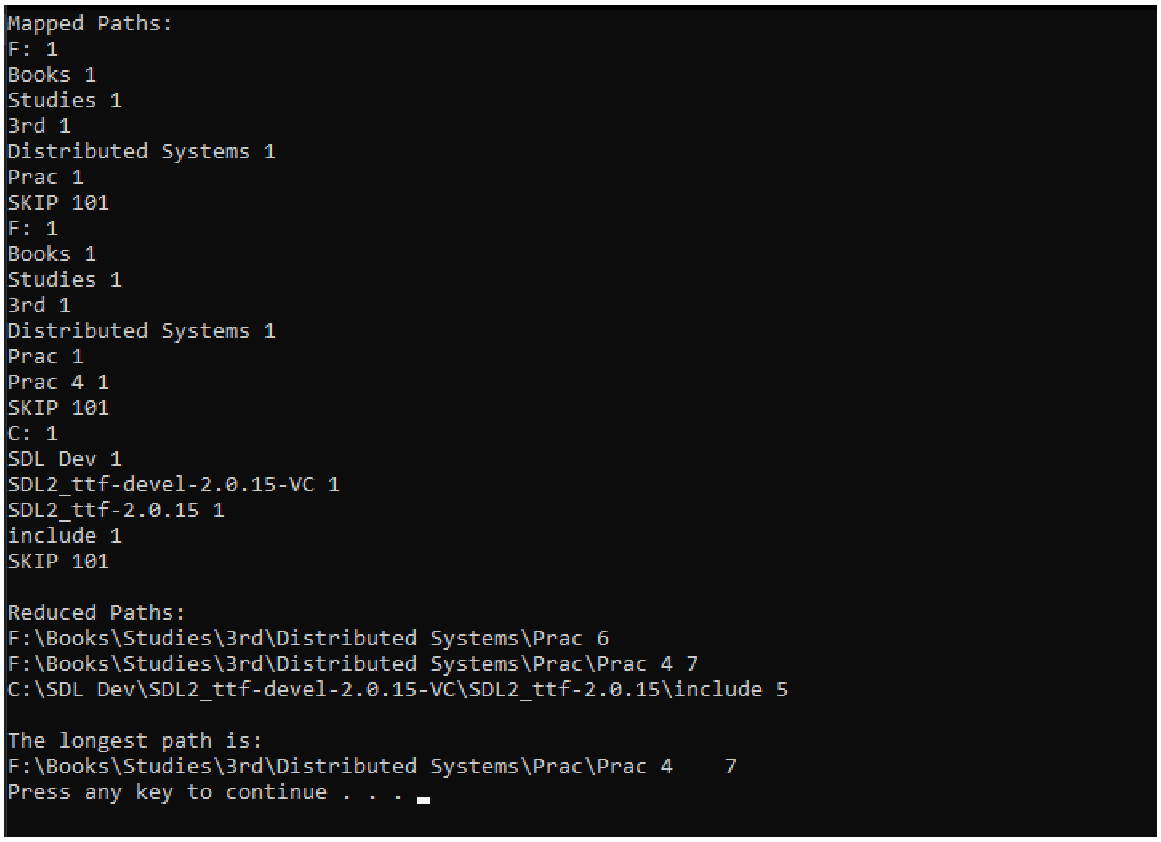


```cpp
void reduce(std::vector<std::pair<std::string, int>>) {
    std::vector<std::pair<std::string, int>>::iterator i;
    std::vector<std::pair<std::string, int>>::iterator j;
    //Sorting
    for (i = vector.begin(); i != vector.end(); i++) {
        for (j = i + 1; j != vector.end(); j++) {
            if (i->first != "SKIP" && j->second != 101) {
                i->first.append("\\" + j->first);
                i->second++;
            }
            else {
                result.push_back(std::make_pair(i->first,i->second));
                i = j;
            }
        }
    }

    i = result.begin();
    while (i != result.end()) {
        if (i->second == 101) {
            i = result.erase(i);
        }
        else {
            ++i;
        }
    }
```

2.3 Result

The sentence can be changed in the codes, we haven't tested if it works with Uppercase and Lowercase yet, but 1 think to note is that at the end of the sentence, it needs a "BACKSLASH" black slash or the program won't recognize the final word.



```
Mapped Paths:
F: 1
Books 1
Studies 1
3rd 1
Distributed Systems 1
Prac 1
SKIP 101
F: 1
Books 1
Studies 1
3rd 1
Distributed Systems 1
Prac 1
Prac 4 1
SKIP 101
C: 1
SDL Dev 1
SDL2_ttf-devel-2.0.15-VC 1
SDL2_ttf-2.0.15 1
include 1
SKIP 101

Reduced Paths:
F:\Books\Studies\3rd\Distributed Systems\Prac 6
F:\Books\Studies\3rd\Distributed Systems\Prac\Prac 4 7
C:\SDL Dev\SDL2_ttf-devel-2.0.15-VC\SDL2_ttf-2.0.15\include 5

The longest path is:
F:\Books\Studies\3rd\Distributed Systems\Prac\Prac 4    7
Press any key to continue . . .
```

In conclusion:

The program is running smoothly and it's working as intended. The program is to our satisfaction!!