Simplify Path

Given an absolute path for a file (Unix-style), simplify it.

For example,
path = `"/home/"`, => `"/home"`
path = `"/a/./b/../../c/"`, => `"/c"`

click to show corner cases.

Corner Cases:

- Did you consider the case where **path** = `"/../"`?
  In this case, you should return `"/"`.
- Another corner case is the path might contain multiple slashes `'/'` together, such as `"/home//foo/"`.
  In this case, you should ignore redundant slashes and return `"/home/foo"`.

Solution 1

C++ also have *getline* which acts like Java's *split*. I guess the code can comment itself.

```
string simplifyPath(string path) {
    string res, tmp;
    vector<string> stk;
    stringstream ss(path);
    while(getline(ss,tmp,'/')) {
        if (tmp == "" or tmp == ".") continue;
        if (tmp == ".." and !stk.empty()) stk.pop_back();
        else if (tmp != "..") stk.push_back(tmp);
    }
    for(auto str : stk) res += "/"+str;
    return res.empty() ? "/" : res;
}
```

written by monaziyi original link here

Solution 2

Hi guys!

The main idea is to push to the stack every valid file name (not in {"",".",".."}), popping only if there's smth to pop and we met "..". I don't feel like the code below needs any additional comments.

```
public String simplifyPath(String path) {
    Deque<String> stack = new LinkedList<>();
    Set<String> skip = new HashSet<>(Arrays.asList("..",".",""));
    for (String dir : path.split("/")) {
        if (dir.equals("..") && !stack.isEmpty()) stack.pop();
        else if (!skip.contains(dir)) stack.push(dir);
    }
    String res = "";
    for (String dir : stack) res = "/" + dir + res;
    return res.isEmpty() ? "/" : res;
}
```

Hope it helps!

written by shpolsky original link here

Solution 3

1. traverse the string to record each folder name.
2. two special cases:

a.double dot:pop one.

b.single dot: do nothing (don`t push it).

```
string simplifyPath(string path) {
    vector<string>   nameVect;
    string name;

    path.push_back('/');
    for(int i=0;i<path.size();i++){
        if(path[i]=='/'){
            if(name.size()==0)continue;
            if(name==".."){     //special case 1: double dot, pop dir
                 if(nameVect.size()>0)nameVect.pop_back();
            }else if(name=="."){//special case 2:singel dot, don`t push
            }else{
                nameVect.push_back(name);
            }
            name.clear();
        }else{
            name.push_back(path[i]);//record the name
        }
    }

    string result;
    if(nameVect.empty())return "/";
    for(int i=0;i<nameVect.size();i++){
        result.append("/"+nameVect[i]);
    }
    return result;
}
```

written by enriquewang original link here