Listings

(please make change to the *.tex file)

Untitled

July 28, 2015

File: \home\hoang\io_project\io.h

Listing 1: io.h

```
1  #include <iostream>
2  #include <string>
3  #include <unistd.h>
4  #include <sys/stat.h>
5  #include <dirent.h>
6  #include <fstream>
7  #include "template.h"
8
9  using namespace std;
10
11 bool has_suffix(const string &str, const string &suffix)
12 {
13   return str.size() >= suffix.size() &&
14     str.compare(str.size() - suffix.size(),
15         suffix.size(), suffix) == 0;
16 }
17
18 string write_tex_template(string filename) {
19   cout << "Opening file " << filename << "..." << endl;
20   string start = "\\lstinputlisting";
21   string line;
22   const char* newfilename = filename.c_str();
23   ifstream myfile (newfilename);
24
25   if (myfile.is_open()) {
26     start += "[caption=" + filename +"]{" + filename + "}\n";
27     myfile.close();
28     cout << "Successfully convert file " << filename << " to latex." <<
         endl;
29     return start;
30   }
31   else {
32     cerr << "Cannot open file!" << endl;
33     return NULL;
34   }
35 }
36
37 string recursive_tex_folder(string dirname) {
38   DIR* dir;
39   struct dirent *ent;
```

```
40
41    string all = "";
42
43    if ((dir = opendir(dirname.c_str())) != NULL) {
44      while ((ent = readdir(dir)) != NULL) {
45        struct stat st;
46        stat(ent->d_name, &st);
47        string name (ent->d_name);
48
49        if (S_ISDIR(st.st_mode)) {
50          if (name.compare(".") != 0 && name.compare("..") != 0) {
51            cout << "Opening directory " << name << endl;
52            all += recursive_tex_folder(name);
53          }
54        }
55
56        else {
57          if (has_suffix(name, ".cpp") || has_suffix(name, ".h")) {
58            //all += ("File: " + dirname + "/" + name + '\n');
59            string valid_dir = "File: " + dirname + "/" + name + '\n';
60            all += generate_valid_tex_directory(valid_dir);
61            all += write_tex_template(name);
62            all += "\\clearpage\n";
63          }
64        }
65      }
66    }
67    else {
68      cout << "Folder " << dirname << " is not a valid folder!";
69      return "";
70    }
71    return all;
72  }
73
74  bool check_file_existence(string dirname, string filename) {
75    DIR* dir;
76    struct dirent *ent;
77
78    if ((dir = opendir(dirname.c_str())) != NULL) {
79      while ((ent = readdir(dir)) != NULL) {
80        struct stat st;
81        stat(ent->d_name, &st);
82        string name (ent->d_name);
83
84        if (filename.compare(name) == 0) {
85          return ! S_ISDIR(st.st_mode);
86        }
87      }
88
89      return false;
90    }
91
92    return false;
```

```
93  }
```

File: \home\hoang\io_project\template.h

Listing 2: template.h

```
1 #include "constants.h"
2 #include <cstdlib>
3
4 const string generating_template(string doc_class, string project_name,
     string author,
5                                  string orientation, int font_size, string
                                       content)
6 {
7   string header = "\\documentclass[" + to_string(font_size) + "pt, " +
      orientation + ", a4paper]{"
8                   + doc_class +"}\n";
9   header += "\\usepackage{listings}\n";
10  header += "\\lstset{breaklines=true,"
11                    "numbers=left,"
12                    "firstnumber=1,"
13                    "stepnumber=1,"
14                    "numberfirstline=true}\n";
15  // this options can be exempted up to user's preference
16  header += "\\usepackage[left=2cm,right=2cm,top=2cm,bottom=2cm]{geometry
      }\n";
17  header += "\\title{" + project_name +"}\n";
18  header += "\\author{" + author +"}\n";
19  header += "\\begin{document}\n\\lstlistoflistings\n\\maketitle\n" +
      content + "\n" + "\\end{document}\n";
20
21  return header;
22 }
23
24 const string default_generation(string content) {
25   return generating_template(DEFAULT_DOCUMENT_CLASS,
26                              DEFAULT_AUTHOR,
27                              DEFAULT_PROJECT_NAME,
28                              DEFAULT_ORIENTATION,
29                              DEFAULT_FONT_SIZE,
30                              content);
31 }
32
33 const string generate_valid_tex_directory(string dirname) {
34   const char* tex_template = "\\textbackslash ";
35   int l = dirname.length();
36   int backslash_count = 0;
37   int underscore_count = 0;
38
39   for (int i = 0; i < l; i++) {
40     if (dirname.at(i) == '/')
41       backslash_count += 1;
42     if (dirname.at(i) == '_')
43       underscore_count += 1;
44   }
45
46   int new_length = l + 14 * backslash_count + underscore_count;
```

```
47    char* valid_dir = (char *) malloc(sizeof(char) * new_length);
48
49    valid_dir[new_length] = '\0';
50
51    for (int i = l-1; i >=0; i--) {
52      if (dirname.at(i) == '/') {
53        for (int j = 1; j <= 15; j++)
54          valid_dir[new_length - j] = tex_template[15 - j];
55        new_length = new_length - 15;
56      }
57      else if (dirname.at(i) == '_') {
58        valid_dir[new_length - 1] = '_';
59        valid_dir[new_length - 2] = '\\';
60        new_length = new_length - 2;
61      }
62      else {
63        valid_dir[new_length - 1] = dirname.at(i);
64        new_length = new_length - 1;
65      }
66    }
67    string result (valid_dir);
68    return result;
69  }
```

File: \home\hoang\io_project\constants.h

Listing 3: constants.h

```
1  // LaTeX constants
2  #include <iostream>
3
4  using namespace std;
5
6  #define DEFAULT_FONT_SIZE 11
7  #define DEFAULT_DOCUMENT_CLASS "article"
8  #define DEFAULT_PROJECT_NAME "Untitled"
9  #define DEFAULT_AUTHOR "(please make change to the *.tex file)"
10 #define DEFAULT_ORIENTATION ""
11
12 /*
13 typedef struct {
14   string name;
15   string author;
16   string documentation;
17 } copyright;
18 */
19
20 const char* no_copyright_form = "In order to make your document well-
       presented,"
21                                 "you should make a text file \"COPYRIGHT
                                      \" to include all"
22                                 "of the project information.";
23
24 const char* preferred_copyright_form = "PROJECT NAME: (fill here)\n"
25                                        "AUTHOR: (fill here)\n"
26                                        "DOCUMENTATION: (fill here)";
```

File: \home\hoang\io_project\main.cpp

Listing 4: main.cpp

```
1  #include "io.h"
2
3  int main() {
4    char* current = get_current_dir_name();
5    string dir (current);
6    string s = recursive_tex_folder(dir);
7    string full_document = default_generation(s);
8
9
10   ofstream output;
11   output.open ("generated.tex");
12   output << full_document;
13   output.close();
14
15   bool existing_info = check_file_existence(dir, "COPYRIGHT");
16   if (existing_info) {
17     cout << "You have the copy right form" << endl;
18   } else {
19     cout << no_copyright_form << endl;
20   }
21
22   return 0;
23 }
```

File: \home\hoang\io_project\try.cpp

Listing 5: try.cpp

```
1  #include <iostream>
2  #include "template.h"
3  #include <unistd.h>
4
5  int main() {
6    const char* dir = get_current_dir_name();
7    string cplusdir (dir);
8    string valid = generate_valid_tex_directory(cplusdir);
9    cout << valid << endl;
10 }
```