(please make change to the *.tex file)

Untitled

November 11, 2015

## Listings

File: /home/hoang/shell/src-pdf/try.cpp

```
#include <iostream>
#include "template.h"
#include <unistd.h>

int main() {
  const char* dir = get_current_dir_name();
  string cplusdir (dir);
  string valid = generate_valid_tex_directory(cplusdir);
  cout << valid << endl;
}
```

Listing 1: /home/hoang/shell/src-pdf/try.cpp

File: /home/hoang/shell/src-pdf/template.h

```
#include "constants.h"
#include <cstdlib>

const string generating_template(string doc_class, string project_name, string author,
    string orientation, int font_size, string content)
{
  string header = "\\documentclass[" + to_string(font_size) + "pt, " + orientation + ", a4paper]{"
    + doc_class +"}\n";
  header += "\\usepackage{listings}\n";
  // this options can be exempted up to user's preference
  header += "\\usepackage[left=2cm,right=2cm,top=2cm,bottom=2cm]{geometry}\n";
  header += "\\usepackage{color}\n"
    "\\definecolor{codegreen}{rgb}{0,0,0.6}\n"
    "\\definecolor{codegray}{rgb}{0.5,0.5,0.5}\n"
    "\\definecolor{codepurple}{rgb}{0.58, 0.0, 0.82}\n"
    "\\definecolor{backcolour}{rgb}{0.95, 0.95, 0.92}\n"
    "\\lstdefinestyle{mystyle}{"
    "commentstyle=\\color{codegreen},"
    "keywordstyle=\\color{magenta},"
    "numberstyle=\\tiny\\color{codegray},"
    "stringstyle=\\color{codepurple},"
    "basicstyle=\\footnotesize\\ttfamily,"
    "breakatwhitespace=false,"
    "breaklines=true,"
    "captionpos=b,"
    "keepspaces=true,"
    "numbers=left,"
    "numbersep=5pt,"
    "showspaces=false,"
    "showstringspaces=false,"
    "showtabs=false,"
    "tabsize=2} \n";
  header += "\\lstset{style=mystyle}\n";
  header += "\\title{" + project_name +"}\n";
  header += "\\author{" + author +"}\n";
  header += "\\begin{document}\n\\maketitle\n\\lstlistoflistings\n\\clearpage\n" +
    content + "\n" + "\\end{document}\n";

  return header;
}

const string default_generation(string content) {
  return generating_template(DEFAULT_DOCUMENT_CLASS,
      DEFAULT_AUTHOR,
      DEFAULT_PROJECT_NAME,
      DEFAULT_ORIENTATION,
      DEFAULT_FONT_SIZE,
      content);
}

const string generate_valid_tex_directory(string dirname) {
  int l = dirname.length();
  int underscore_count = 0;

  for (int i = 0; i < l; i++) {
    if (dirname.at(i) == '_')
      underscore_count += 1;
  }

  int new_length = l + underscore_count;
  char* valid_dir = new char[new_length];

  valid_dir[new_length] = '\0';
```

```
63
64    for (int i = l-1; i >=0; i--) {
65      if (dirname.at(i) == '_') {
66        valid_dir[new_length - 1] = '_';
67        valid_dir[new_length - 2] = '\\';
68        new_length = new_length - 2;
69      }
70      else {
71        valid_dir[new_length - 1] = dirname.at(i);
72        new_length = new_length - 1;
73      }
74    }
75    string result (valid_dir);
76    return result;
77  }
```

Listing 2: /home/hoang/shell/src-pdf/template.h

File: /home/hoang/shell/src-pdf/io.h

```
#include <string>
#include <unistd.h>
#include <sys/stat.h>
#include <dirent.h>
#include <fstream>
#include "template.h"

using namespace std;

bool has_suffix(const string &str, const string &suffix)
{
  return str.size() >= suffix.size() &&
    str.compare(str.size() - suffix.size(),
        suffix.size(), suffix) == 0;
}

bool has_valid_suffix(const string &filename) {
  for (int i = 0; i < SUPPORTED_LANGUAGES; i++) {
    if (has_suffix(filename, valid_file_extension[i]))
      return true;
  }
  return false;
}

string write_tex_template(string filename) {
  cout << "Opening FILE " << filename << "..." << endl;
  string start = "\\lstinputlisting";
  string line;
  const char* newfilename = filename.c_str();
  ifstream myfile (newfilename);

  if (myfile.is_open()) {
    start += "[caption=" + generate_valid_tex_directory(filename) +"]{"
                         + filename + "}\n";
    myfile.close();
    cout << "__Successfully__ convert file " << filename << " to latex." << endl;
    return start;
  }
  else {
    cerr << "==> !!!File " << filename << " cannot be opened!!!" << endl;
    return "";
  }
}

string recursive_tex_folder(string dirname) {
  DIR* dir;
  struct dirent *ent;

  string all = "";

  if ((dir = opendir(dirname.c_str())) != NULL) {
    while ((ent = readdir(dir)) != NULL) {
      struct stat st;
      lstat(ent->d_name, &st);
      string name (ent->d_name);

      if (S_ISDIR(st.st_mode)) {
  cout << name << " is a directory " << endl;
        if (name.compare(".") != 0 && name.compare("..") != 0 && name.compare(".git")
     != 0) {
          cout << "Opening DIRECTORY " << name << endl;
          all += recursive_tex_folder(dirname +"/" + name);
        }
      }

```

```
65        else {
66          if (has_valid_suffix(name) && name.compare("main") != 0) {
67            //all += ("File: " + dirname + "/" + name + '\n');
68            string valid_dir = "File: " + dirname + "/" + name + '\n';
69            all += generate_valid_tex_directory(valid_dir);
70            all += write_tex_template(dirname +"/" + name);
71            all += "\\clearpage\n";
72          }
73        }
74      }
75    }
76    else {
77      cout << " *** Folder " << dirname << " is not a valid folder!\n";
78      return "";
79    }
80    return all;
81  }
82
83  bool check_file_existence(string dirname, string filename) {
84    DIR* dir;
85    struct dirent *ent;
86
87    if ((dir = opendir(dirname.c_str())) != NULL) {
88      while ((ent = readdir(dir)) != NULL) {
89        struct stat st;
90        stat(ent->d_name, &st);
91        string name (ent->d_name);
92
93        if (filename.compare(name) == 0) {
94          return ! S_ISDIR(st.st_mode);
95        }
96      }
97
98      return false;
99    }
100
101   return false;
102 }
```

Listing 3: /home/hoang/shell/src-pdf/io.h

File: /home/hoang/shell/src-pdf/main.cpp

```
#include "io.h"

int main() {
  char* current = get_current_dir_name();
  string dir (current);
  string s = recursive_tex_folder(dir);
  string full_document = default_generation(s);

  ofstream output;
  output.open ("generated.tex");
  output << full_document;
  output.close();

  bool existing_info = check_file_existence(dir, "COPYRIGHT");
  if (existing_info) {
    cout << "You have the copyright form" << endl;
  } else {
    cout << no_copyright_form << endl;
  }

  return 0;
}
```

Listing 4: /home/hoang/shell/src-pdf/main.cpp

File: /home/hoang/shell/src-pdf/constants.h

```
// LaTeX constants
#include <iostream>

using namespace std;

#define DEFAULT_FONT_SIZE 11
#define DEFAULT_DOCUMENT_CLASS "article"
#define DEFAULT_PROJECT_NAME "Untitled"
#define DEFAULT_AUTHOR "(please make change to the *.tex file)"
#define DEFAULT_ORIENTATION ""

#define SUPPORTED_LANGUAGES 13

/*
typedef struct {
  string name;
  string author;
  string documentation;
} copyright;
*/

const char* no_copyright_form = "In order to make your document well-presented,"
                                "you should make a text file \"COPYRIGHT\" to include all"
                                "of the project information.";

const char* preferred_copyright_form = "PROJECT NAME: (fill here)\n"
                                       "AUTHOR: (fill here)\n"
                                       "DOCUMENTATION: (fill here)";

const string valid_file_extension[SUPPORTED_LANGUAGES] = {".cpp", // C++
                                                          ".h", // C/C++ header
                                                          ".c", // C
                                                          ".S", // Assembly
                                                          ".asm", // Assembly
                                                          ".java", // Java
                                                          ".py", // Python
                                                          ".hs", // Haskell
                                                          ".cs", //C#
                                                          ".js", // Javascript
                                                          ".html", // HTML
                                                          ".css", // CSS
                                                          ".rkt"// racket
                                                         };
```

Listing 5: /home/hoang/shell/src-pdf/constants.h