myGit – A version control system

Created By-

Yash Halgaonkar,

Tanishka Gupta,

Chinmay Ingle,

Mansi Jadhav,

Suyog Jadhav

*#include*<iostream>

*#include*<vector>

*#include*<string>

*#include*<fstream>

*#include*<windows.h>

*#include*<dir.h>

*#include*<iomanip>

*#include*<string.h>

*#include* <sys/stat.h>

*#include* <unistd.h>

*#include*<stdlib.h>

*#include* <dirent.h>

*#include* <sys/types.h>

using namespace std;

*//constants*

static string MYGIT = "myGit";

static int console\_width = 100;

*//location string of the project*

static string filepath;

*//location string of the myGit folder of the project*

static string filepath\_git;

*//name of the file currently working on*

static string filename;

static string ext;

static string divider = "=======================================================================================================================";

*//Funtion to print a string in the center of the console*

void print(string str)

{

*// Width of console. Default is 80.*

    int len = str.length();

*// Length of the string.*

*if*(len%2==0) str+=" ";

*if*(len==2){

*//str=>>*

        cout<<setw((console\_width/4)+len/2)<<right<<str;

    }

*else*

        cout<<setw((console\_width/2)+len/2)<<right<<str;

}

*//funtion to add a divider*

inline void add()

{

    cout<<divider<<endl;

}

*//Function to print the content of the file.*

void displayCode(string path)

{

    add();

    print("DISPLAY\n");

    ifstream infile(path.c\_str());

*//execute a loop if file a opened successfully*

    int i=0;

*while*(infile)

    {

        i++;

        string line;

        getline(infile, line);

        cout<<i<<"| "<<line<<endl;

    }

    infile.close();

    add();

}

*//Function the current working file with the file in the path*

void compareFiles(string path)

{

    ifstream temp(path.c\_str()), original(filepath.c\_str());

    int i=0;

*while*(temp || original)

    {

        i++;

        string tempLine, codeLine;

        getline(temp,tempLine);

        getline(original,codeLine);

*if*(tempLine != codeLine){

*//mismatch found*

*if*(codeLine.empty())

                cout<<i<<"| "<<tempLine<<" -> "<<"<BLANK>"<<endl;

*else*

                cout<<i<<"| "<<tempLine<<" -> "<<codeLine<<endl;

        }

    }

    temp.close();

    original.close();

}

*//to list dir in the path*

vector<string> list\_dir(string path)

{

    struct dirent \*entry;

    vector<string> list;

    DIR \*dir = opendir(path.c\_str());

*while* ((entry = readdir(dir)) != NULL) {

        string s = entry->d\_name;

        list.push\_back(s);

*//printf("%s\n",entry->d\_name);*

    }

    closedir(dir);

*return* list;

}

*//Funtion to create a folder/directory at given path.*

inline void makeDir(string path)

{

    mkdir(path.c\_str());

}

*//display the list of versions and return the selected version.*

void checkVersions()

{

    vector<string> vers;

    print("Following versions are available:\n");

    string temp = filepath\_git + "\\" + filename;

    int one = 0, two = 0, three = 0;

    vector<string> files = list\_dir(filepath\_git);

*for*(size\_t i =0; i<files.size(); i++)

    {

*//print(files[i] + "\n");*

*if*(files[i] == filename + "\_v1" + ext )

        {

*//version 1 exists*

*//print(files[i]);*

            vers.push\_back(files[i]);

            one = 1;

        }

*if*(files[i] == filename + "\_v2" + ext)

        {

*//version 2 exists*

            vers.push\_back(files[i]);

            two = 1;

        }

*if*(files[i] == filename + "\_v3" + ext){

            three = 1;

            vers.push\_back(files[i]);

        }

    }

*if*(vers.size()==0){

        print("No version found\n");

    }

*else*{

*for*(size\_t i=0; i<vers.size(); i++)

        {

            print(vers[i] + "\n");

        }

    }

}

*//Function to check if a file exists*

inline bool fileExists (const std::string& name) {

  struct stat buffer;

*return* (stat (name.c\_str(), &buffer) == 0);

}

*//funtion shows available version of the file and compares them with the code*

void compareWithVersions()

{

    print("Select a version to compare:\n");

    checkVersions();

    print("Enter version number:\n");

    string ver;

    print(">>");

    cin>>ver;

    string temp = filename + "\_v" + ver + ext;

*//check if the version exists...*

    vector<string> files = list\_dir(filepath\_git);

    int exists = 0;

*for*(size\_t i =0; i<files.size(); i++)

    {

*//print(files[i] + "\n");*

*if*(temp == files[i])

        {

            exists = 1;

*break*;

        }

    }

*if*(exists==1){

        add();

        print("COMPARE\n");

        compareFiles(filepath\_git + "\\" + temp);

        add();

    }

*else*{

        print("No "+ ver + "version found.\n");

    }

}

*//!display the list of available version and revert back to them*

void revertBack()

{

    print("Choose a version to revert to: \n");

    checkVersions();

    print("Enter version:\n");

    string ver;

    cin>>ver;

*//check if the version exists*

*//check if the version exists...*

    string temp = filename + "\_v" + ver + ext;

    vector<string> files = list\_dir(filepath\_git);

    int exists = 0;

*for*(size\_t i =0; i<files.size(); i++)

    {

*//print(files[i] + "\n");*

*if*(temp == files[i])

        {

            exists = 1;

*break*;

        }

    }

*if*(exists==1){

*//open teh curret code in write mode and copy the content*

        ofstream outfile((filepath + "\\" + filename + ext).c\_str());

        ifstream infile((filepath\_git + temp).c\_str());

*while*(infile){

*//read*

            string line;

            getline(infile , line);

            outfile<<line;

        }

        print("Successfully Reverted back to " + ver + " version\n");

    }

*else*{

        print("No "+ ver + "version found.\n");

    }

}

*//funtion to create a version of the file*

void createVersion()

{

    string temp = filepath\_git + "\\" + filename;

    int one = 0, two = 0, three = 0;

    vector<string> files = list\_dir(filepath\_git);

*for*(size\_t i =0; i<files.size(); i++)

    {

*//print(files[i] + "\n");*

*if*(files[i] == filename + "\_v1" + ext )

        {

*//version 1 exists*

            one = 1;

        }

*if*(files[i] == filename + "\_v2" + ext)

        {

*//version 2 exists*

            two = 1;

        }

*if*(files[i] == filename + "\_v3" + ext){

            three = 1;

        }

    }

    temp = filepath\_git;

*if*(one ==1 && two ==1 && three ==1)

    {

        cout<<"\n";

        print("Three version exist. Delete some versions.\n");

    }

*else* *if*(one ==1 && two ==1 && three ==0)

    {

        temp = temp + "\\" + filename +  "\_v3" + ext;

*//create version three*

    }

*else* *if*(one ==1 && two ==0 && three ==0){

*//create version two*

        temp = temp + "\\" + filename +  "\_v2" + ext;

    }

*else* *if*(one ==0 && two ==0 && three ==0){

*//create version one*

        temp = temp + "\\" + filename +  "\_v1" + ext;

    }

*//crate file and copy the content*

    ofstream outfile(temp.c\_str());

    ifstream infile((filepath + "\\" + filename + ext).c\_str());

*while*(infile){

        string line;

        getline(infile , line);

        outfile<<line<<endl;

    }

    infile.close();

    outfile.close();

    print("Version created\n");

}

*//main program*

int main()

{

    cout<<"\n\n\n\n\n";

    add();

    print("myGit\n");

    print("----------Version Control System----------\n");

    add();

    print("Created By-\n");

    print("Yash Halgaonkar\n");

    print("Tanishka Gupta\n");

    print("Chinmay Ingle\n");

    print("Mansi Jadhav\n");

    print("Suyog Jadhav\n");

    add();

    cout<<"\n";

    print("Enter the path of the project directory: \n");

    print(">>");

    getline(cin , filepath);

*while*(!fileExists(filepath))

    {

*//loop when user entered a wrong address untill right address is entered*

        cout<<"\n";

        print("Enter a valid path: \n");

        print(">>");

        getline(cin , filepath);

    }

*//if the filepath is valid then check if a directory named myGit exists*

*//path to muGit dir*

    filepath\_git = filepath + "\\myGit";

*if*(fileExists(filepath\_git)){

*//project has mygit dir*

        cout<<"\n";

        print("This project has myGit.\n");

    }

*else*{

        cout<<"\n";

        print("No records found\n");

        cout<<"\n";

        print("Creating myGit...\n");

        makeDir(filepath\_git);

    }

    print("Enter file name: \n");

    print(">>");

    getline(cin , filename);

    print("Enter extension: \n");

    print(">>");

    getline(cin , ext);

*while*(!fileExists(filepath + "\\" + filename + ext))

    {

*//loop untill user enters a file that exists*

            print("Enter a valid file name: \n");

            print(">>");

            getline(cin , filename);

            print("Enter extension: \n");

            print(">>");

            getline(cin , ext);

    }

*//menu driven code*

    int choice = -1;

*while*(choice != 8 )

    {

        cout<<divider<<endl;

        print("Current working file: " + filename + "\n");

        print("SELECT OPERATION\n");

*//menu*

        print("1. Display Code\n");

        print("2. Create Version\n");

        print("3. Check for versions\n");

        print("4. Compare with versions\n");

        print("5. Revert Back\n");

        print("6. Change  working file\n");

        print("7. Change project\n");

        print("8. Exit\n");

        cout<<divider<<endl;

        print(">>");

        cin>>choice;

*switch*(choice)

        {

*case*(1):

                displayCode(filepath + "\\" + filename);

*break*;

*case* 2:

                createVersion();

*break*;

*case* 3:

                checkVersions();

*break*;

*case* 4:

                compareWithVersions();

*break*;

*case* 5:

                revertBack();

*break*;

*case* 6:

            {

                print("Enter file name: \n");

                print(">>");

                getline(cin , filename);

*while*(!fileExists(filepath + "\\" + filename))

                {

*//loop untill user enters a file that exists*

                        print("Enter a valid file name: \n");

                        print(">>");

                        getline(cin , filename);

                }

*break*;

            }

*case* 7:

            {

                print("Enter the path of the project directory: \n");

                print(">>");

                getline(cin , filepath);

*while*(!fileExists(filepath))

                {

*//loop when user entered a wrong address untill right address is entered*

                    cout<<"\n";

                    print("Enter a valid path: \n");

                    print(">>");

                    getline(cin , filepath);

                }

*break*;

            }

*case* 8:

            {

                print("Are you sure you want to exit??\n");

                print("Y/N\n");

                print(">>");

                string temp;

                cin>>temp;

*if*(temp == "y" || temp == "Y")

                {

                    add();

                    print("THANKS FOR USING :-)\n");

                    add();

                    exit(1);

                }

*break*;

            }

*default*:

                print("Invalid Choice.");

*break*;

        }*//switch*

    }*//while loop*

}*//main program*

*/\**

*\* myGit - A version control system*

*\**

*\**

*\**

*\**

*\*/*