21BDS0340

Abhinav Dinesh Srivatsa

Operating Systems Lab

Assignment – II

**Question 1**

Aim:

To simulate the ‘ls’ command

Program:

#include <iostream>

#include <filesystem>

namespace fs = std::filesystem;

void listDirectoryContents(const std::string &path)

{

// listing contents of directory

for (const auto &entry : fs::directory\_iterator(path))

std::cout << entry.path().filename().string() << std::endl;

}

int main(int argc, char \*argv[])

{

std::string path;

// if number of arguments > 1, then path is given

if (argc > 1)

path = argv[1];

// else assigne path to current directory

else

path = fs::current\_path().string();

listDirectoryContents(path);

}

Output:

A computer screen with white text

Description automatically generated with low confidence

**Question 2**

Aim:

To demonstrate system calls for files

Program:

#include <iostream>

#include <fstream>

int main()

{

// opening output file

std::ofstream outputFile("output.txt");

if (!outputFile)

{

std::cerr << "Failed to open the file for writing." << std::endl;

return 1;

}

// writing data to file

outputFile << "Hi my name is Abhinav Dinesh Srivatsa." << std::endl;

outputFile << "My registration number is 21BDS0340." << std::endl;

outputFile.close();

// opening input file

std::ifstream inputFile("output.txt");

if (!inputFile)

{

std::cerr << "Failed to open the file for reading." << std::endl;

return 1;

}

// reading data from file

std::string line;

while (std::getline(inputFile, line))

{

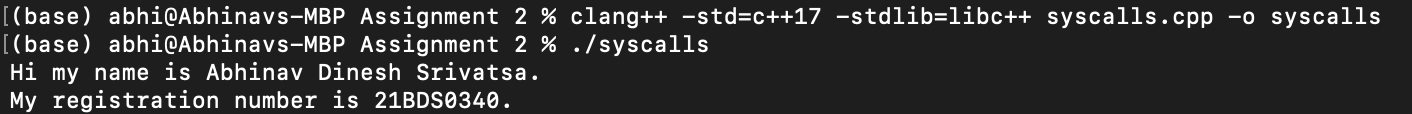
std::cout << line << std::endl;

}

inputFile.close();

}

Output:



A picture containing text, font, screenshot, white

Description automatically generated

**Question 3**

Aim:

To simulate the ‘cd’ command

Program:

#include <iostream>

#include <unistd.h>

int main(int argc, char \*argv[])

{

// checking if 2 arguments present

if (argc != 2)

{

std::cerr << "Usage: " << argv[0] << " <directory>" << std::endl;

return 1;

}

// sending error if change directory invalid

if (chdir(argv[1]) != 0)

{

std::cerr << "Failed to change directory." << std::endl;

return 1;

}

// displaying current directory if valid

char \*cwd = getcwd(nullptr, 0);

if (cwd != nullptr)

{

std::cout << "Current directory: " << cwd << std::endl;

free(cwd);

}

else

{

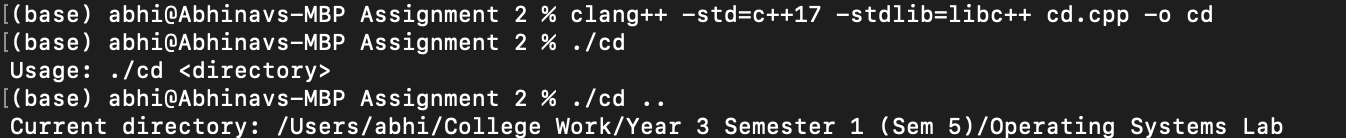
std::cerr << "Failed to get current directory." << std::endl;

return 1;

}

}

Output:



**Question 4**

Aim:

To simulate the ‘mv’ command

Program:

#include <iostream>

#include <cstdio>

int main(int argc, char \*argv[])

{

// checking if 3 arguments provided

if (argc != 3)

{

std::cerr << "Usage: " << argv[0] << " <source> <destination>" << std::endl;

return 1;

}

const char \*source = argv[1];

const char \*destination = argv[2];

// if renaming file did not work, throw error

if (rename(source, destination) != 0)

{

std::cerr << "Failed to move the file." << std::endl;

return 1;

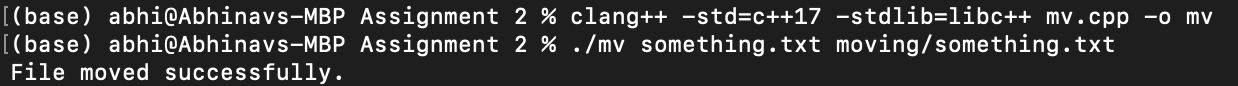
}

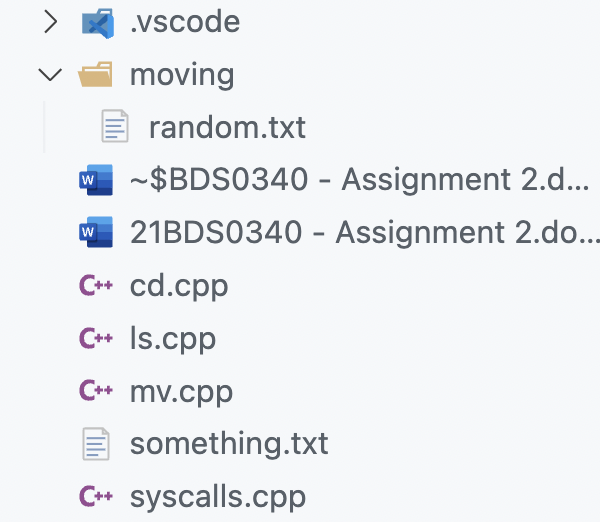
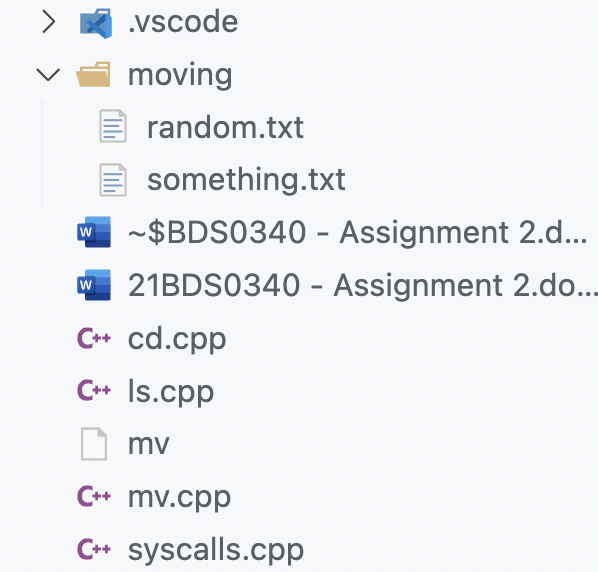
// display success message

std::cout << "File moved successfully." << std::endl;

}

Output:



Before: After: