ASSIGNMENT 3

PROGRAM

os3.cpp:

#include <iostream>

#include <fstream>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <cstdlib>

using namespace std;

int main() {

int fd[2], pid;

ifstream file("source.txt");

if (!file) {

cerr << "Error occured while opening the file!" << endl;

return 1;

}

int pipefd[2];

pipe(pipefd);

if ((pid = fork()) == 0) {

close(pipefd[0]);

string line;

while (getline(file, line)) {

line += "\n";

write(pipefd[1], line.c\_str(), line.size());

}

file.close();

close(pipefd[1]);

exit(0);

}

close(pipefd[1]);

dup2(pipefd[0], STDIN\_FILENO);

close(pipefd[0]);

pipe(fd);

if ((pid = fork()) == 0) {

dup2(fd[1], STDOUT\_FILENO);

close(fd[0]);

close(fd[1]);

execl("./sort", "sort", (char \*)nullptr);

cerr << "execl failed!" << endl;

exit(1);

} else if (pid > 0) {

dup2(fd[0], STDIN\_FILENO);

close(fd[1]);

close(fd[0]);

execl("./unique", "unique", (char \*)nullptr);

cerr << "execl failed!" << endl;

exit(1);

} else {

cerr << "Fork failed!" << endl;

return 1;

}

return 0;

}

sort.cpp:

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

vector<string> lines;

string line;

// Read input from stdin

while (getline(cin, line)) {

lines.push\_back(line);

}

// Sort the lines

sort(lines.begin(), lines.end());

// Print sorted output

for (const auto &l : lines) {

cout << l << endl;

}

return 0;

}

unique.cpp:

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ofstream outputFile("destination.txt");

if (!outputFile) {

cerr << "Error opening output File!" << endl;

return 1;

}

string prevLine, currLine;

if (getline(cin, prevLine)) {

outputFile << prevLine << endl;

}

while (getline(cin, currLine)) {

if (currLine != prevLine) {

outputFile << currLine << endl;

}

prevLine = currLine;

}

outputFile.close();

return 0;

}

OUTPUT:

(base) pccoe@pc-2:~/Desktop$ g++ sort.cpp -o sort  
(base) pccoe@pc-2:~/Desktop$ g++ unique.cpp -o unique  
(base) pccoe@pc-2:~/Desktop$ g++ os2.cpp  
(base) pccoe@pc-2:~/Desktop$ ./a.out  
(base) pccoe@pc-2:~/Desktop$



