#include "commit\_utils.h"

#include <fstream>

#include <iostream>

#include <ctime>

#include <sstream>

using namespace std;

/// Read file content from the objects directory.

string readFileFromObjects(const string& filename, const string& fileHash) {

ifstream file(".minigit/objects/" + fileHash);

stringstream buffer;

buffer << file.rdbuf();

return buffer.str();

}

// Write file content to the working directory.

void writeFileToWorkingDirectory(const string& filename, const string& content) {

ofstream file(filename);

file << content;

}

CommitData readCommitFromFile(string id) {

CommitData c;

ifstream file(".minigit/commits/" + id);

if (!file.is\_open()) {

cout << "Cannot find commit with ID: " << id << "\n";

return c;

}

getline(file, c.message);

getline(file, c.time);

getline(file, c.parentHash);

getline(file, c.secondParentHash);

string line;

while (getline(file, line)) {

istringstream ss(line);

FileBlob f;

ss >> f.fileHash >> f.fileName;

c.fileList.push\_back(f);

}

c.commitHash = id;

return c;

}

// saves commit data to a file.

void saveCommitToFile(CommitData data) {

ofstream file(".minigit/commits/" + data.commitHash);

file << data.message << "\n";

file << data.time << "\n";

file << data.parentHash << "\n";

file << data.secondParentHash << "\n";

for (auto f : data.fileList) {

file << f.fileHash << " " << f.fileName << "\n";

}

}

// Retrieves the current HEAD commit ID.

string getCurrentHead() {

ifstream file(".minigit/HEAD");

string line;

getline(file, line);

return line;

}

// changes the head to a new commit ID.

void changeHead(string newHead) {

ofstream file(".minigit/HEAD");

file << newHead;

}

// Gets the current date and time as a formatted string.

string getCurrentTime() {

time\_t now = time(0);

char buff[80];

strftime(buff, sizeof(buff), "%Y-%m-%d %H:%M:%S", localtime(&now));

return string(buff);

}