MiniGit: Log & Branch Commands

Log Command (`./minigit log`)

- Displays all past commits from latest to oldest.
- Reads `.minigit/HEAD` to find current branch.
- Gets latest commit hash from `.minigit/refs/`.
- Loads commit files from `.minigit/objects/`.
- Each commit links to its parent (like a linked list).
- Traverses commit chain backward until no parent remains.
- Outputs commit hash, timestamp, and message.

Data structure:
Commit history is a linked list inside a Directed Acyclic Graph (DAG).

Branch Command (`./minigit branch <branch-name>`)

- Creates a new branch by making a new file in `.minigit/refs/`.
- New branch file stores current commit hash.
- Branch is a label pointing to a specific commit, no new commit created.
- Helps work on new features separately without affecting main branch.
- `.minigit/HEAD` indicates the current active branch (e.g., `ref: refs/master`).

The code:

For log:

```
void log() {
    std::string head = read_file(head_file).substr(5);

    std::string commit = fs::exists(refs_dir + "/" + head) ? read_file(refs_dir + "/" + head) : "";

    while (!commit.empty()) {
        std::string content = read_file(objects_dir + "/" + commit);

        std::stringstream ss(content);
        std::string line, msg, time, parent;

        while (std::getline(ss, line)) {
            if (line.find("message: ") == 0) msg = line.substr(9);
            if (line.find("timestamp: ") == 0) time = line.substr(11);
            if (line.find("parent: ") == 0) parent = line.substr(8);
        }

        std::cout << "commit " << commit << "\nDate: " << time << "\n\t" << msg << "\n\n";

        commit = parent;
    }
}
```

For branch:

```
void branch(const std::string& name) {
    std::string ref = read_file(head_file).substr(5);

    std::string hash = fs::exists(refs_dir + "/" + ref) ? read_file(refs_dir + "/" + ref) : "";

    write_file(refs_dir + "/" + name, hash);

    std::cout << "Branch " << name << " created\n";
}
```