

Lab1 实验要求

使用简化的 Markdown 语法来创建和编辑文件。语法包括使用 `#`，`##`，`###` 等作为不同层级的标题项，以及使用 `*`，`-`，`+` 等无序列表或 `1.`，`2.` 等有序列表作为文本项。

基本功能

1 文件操作

命令 1-1: `load 文件路径`

功能描述：

从指定路径加载文件到内存中进行编辑。如果指定的文件不存在，则新建一个文件。文件路径应该是合法的，不能含有非法字符。

输入示例：

```
load test.md
```

Mathematica

输出示例：无

命令 1-2: `save`

功能描述：

将内存中的当前编辑数据保存到之前加载的或新创建的文件中。确保有一个文件已经被加载或创建。

输入示例：

Mathematica

save

输出示例：无

2 编辑功能

命令 2-1: `insert [行号] 标题/文本`

功能描述：

行号参数可选。

在指定行插入标题或文本。如果不指定行号，则默认在文件的最后一行插入内容。

输入示例：

```
insert 3 ## 新的子标题  
list
```

Mathematica

输出示例：

```
# 我的书签  
## 学习资源  
## 新的子标题  
### 编程  
* Java从入门到入土
```

Markdown

命令 2-2: `append-head 标题/文本`

功能描述：

在文件起始位置插入标题或文本。



Tips: 相当于 insert 命令的特殊变体

输入示例：

Mathematica

```
append-head # 新的标题
list
```

输出示例:

```
# 新的标题
# 我的书签
## 学习资源
## 新的子标题
### 编程
* Java从入门到入土
```

Markdown

命令 2-3: `append-tail` 标题/文本

功能描述:

在文件最后一行插入标题或文本。

 Tips: 相当于 insert 命令的特殊变体

输入示例:

```
append-tail * 新的文本
list
```

Mathematica

输出示例:

```
# 新的标题
# 我的书签
## 学习资源
## 新的子标题
### 编程
* Java从入门到入土
* 新的文本
```

Markdown

命令 2-4: `delete` 标题/文本 或 `delete` 行号

功能描述：

删除指定标题或文本。如果指定行号，则删除指定行。当删除的是标题时，其子标题和内容不会被删除。

输入示例：

```
delete Java从入门到入土
```

Mathematica

输出示例：

```
# 新的标题
# 我的书签
## 学习资源
## 新的子标题
### 编程
* 新的文本
```

Markdown

命令 2-5: `undo`

功能描述：

撤销上一次执行的编辑命令，返回到执行该命令前的状态。不适用于非编辑命令。上一个命令属于 `load` 与 `save` 等文件相关命令组时不能被撤销不能被跳过，属于 `list`，`list-tree` 与 `dir-tree` 等显示相关命令组时应该被跳过。

输入示例：

```
undo
list
```

Mathematica

输出示例：

```
# 新的标题
# 我的书签
## 学习资源
## 新的子标题
```

Markdown

```
### 编程
* Java从入门到入土
* 新的文本
```

命令 2-6: redo

功能描述:

重新执行上次撤销的命令，内容返回到执行该命令后的状态。即只有上一个编辑命令是 `undo` 时，才允许执行 `redo`。

输入示例:

```
redo
list
```

Mathematica

输出示例:

```
# 新的标题
# 我的书签
## 学习资源
## 新的子标题
### 编程
* 新的文本
```

Markdown

3 显示功能

命令 3-1: list

功能描述:

以文本形式显示当前编辑的内容。

输入示例:

Mathematica

```
list
```

输出示例：

```
# 新的标题
# 我的书签
## 学习资源
## 新的子标题
### 编程
* 新的文本
```

Markdown

命令 3-2: `list-tree`

功能描述：

以树形结构显示当前编辑的内容。

输入示例：

```
list-tree
```

Markdown

输出示例：

```
├─ 新的标题
└─ 我的书签
    └─ 学习资源
        ├── 新的子标题
        └─ 编程
            └─ ·新的文本
```

Markdown

命令 3-3: `dir-tree [目录]`

功能描述：

目录参数可选。

以树形结构显示指定目录（标题）下的内容。如果不指定目录，默认显示当前工作目录下的内容。

输入示例：

```
dir-tree 学习资源
```

Mathematica

输出示例：

```
└─ 学习资源
    └─ 新的子标题
        └─ 编程
            └─ ·新的文本
```

Markdown

日志模块

记录每一次命令，包括执行的时间戳。同时将每次运行中（Session）的命令历史分段保存在一个日志文件中，退出交互界面后依然可以通过日志查看命令历史。如果 history 执行失败仅仅提示警告，不影响其他功能。

日志文件格式：

```
session start at YYYYMMSS HH:mm:ss
//...
session start at 20230926 14:52:36
20230926 14:53:25 load books.md
20230926 14:55:43 dir-tree 学习资源
session start at 20230928 18:22:56
20230928 18:23:22 load traval.md
20230928 18:23:35 append-head 旅行清单
20230928 18:25:40 save
```

Markdown

命令：`history [数量]`

功能描述：

数量参数可选。

列出指定数量的最近历史命令，包括执行命令的时间戳。默认显示全部记录，但可以通过参数限制显示的数量。如果指定的数量大于存储的历史命令数量，也显示全部命令记录。

输入示例：

```
history 4
```

Mathematica

输出示例：

```
20230928 18:25:40 save
20230928 18:23:35 append-head 旅行清单
20230928 18:23:22 load traval.md
20230926 14:55:43 dir-tree 学习资源
```

Markdown

统计模块

实时在一个日志文件中记录一次运行中（Session）打开的 md 文件以及在各个文件上工作的时长。

1. 可以使用 stats 得到当前运行过程中的统计信息
2. 通知记录到日志中，日志保存了历次的操作
3. 当前编辑文件在没有关闭之前，不必将当前文件的统计信息实时计入日志，当前文件的统计只能通过 stats current 得到。
4. 如果执行 stats 命令失败仅仅提示警告，不影响其他功能继续执行。

日志文件格式：

```
session start at YYYYMMSS HH:mm:ss
//...
session start at YYYYMMSS HH:mm:ss
./test.md 12 分钟
```

Markdown


```
./notes.md 2 小时 15 分钟
./draft.md 45 分钟
```

命令： `stats [all | current]`

功能描述：

all 或 current 参数可选。

显示当前会话中指定文件或所有文件的工作时长统计。默认为 `current`，显示当前文件的统计信息。根据不同时长，以可读的方式显示不同的时长单位。例如，65 分钟显示为 1 小时 5 分钟。

输入示例：

```
stats all
```

Mathematica

输出示例：显示当前会话中所有 Markdown 文件的工作时长

```
session start at YYYYMMSS:HHmmSS
./test.md 12 分钟
./notes.md 2 小时 15 分钟
./draft.md 45 分钟
```

Markdown

时长单位参考：

1. 秒 (Second)
时间段：1 秒
2. 分钟 (Minute)
时间段：60 秒 = 1 分钟
3. 一刻钟 (Quarter of an hour)
时间段：15 分钟 = 一刻钟
4. 半小时 (Half an hour)
时间段：30 分钟 = 半小时

5. 小时 (Hour)

时间段：60 分钟 = 1 小时

6. 天 (Day)

时间段：24 小时 = 1 天

注意事项

1. 两种树形结构的显示应尽可能复用代码。
2. 日志模块和统计模块是相对不稳定的，可能会迭代的功能，核心的模块尽可能避免与这两个模块直接耦合。
3. 编辑前需要先加载文件，同一个会话可以加载并编辑不同的文件：文件操作 → 编辑 | 统计 → 文件操作 → 编辑 | 统计 →

提交要求

提交文件说明

- **源代码**：代码应当尽量整洁，包含（自动化）测试用例。
- **功能完成度列表**：列出所有要求功能，并标注其完成状态（完成/未完成/部分完成），可附上截图。
- 一份简单的 **使用说明**：只需要包括运行环境、启动步骤。
- 所有提交的文件都应该被打包在一个名为 **lab1-学号-姓名** 的 **zip** 压缩文件中。

提交时间

- 截止时间：2023 年 11 月 02 日 23 点 59 分 59 秒。

其他注意事项

- 此实验为 **个人作业**。
- 可采用任意编程语言。
- 灵活运用设计模式以复用代码，降低模块间的耦合度。

- 在截止时间之前 **可以多次提交**。