

Vim

简介

要求

展示文件结构

传递参数

简单整合这两个程序

交互

额外内容

知识点应用

注意事项

合作完成任务

只能使用c/c++语言完成

使用github进行代码同步管理

基本计划

1.第一周

2.第二周

3.第三周

4.第四周

代码审查，答辩

Vim

简介

```
#在Linux系统上，通过在控制台输入
sudo apt-get install tree
#可以安装一个非常有趣的插件，tree。再次输入
tree .
#同时也可以使用
sudo apt-get install vim
#可以安装一个非常实用的插件vim
```

tree得到如下的结果 "."代表当前文件所在路径

```
.
├── dir1
│   ├── file_1_1
│   └── file_1_2
├── dir2
│   ├── file_2_1
│   └── file_2_2
├── file1
└── file2
```

或者

```
tree test
```

指定展示路径

```
..
.
test
├─ dir1
│   ├── file_1_1
│   └─ file_1_2
├─ dir2
│   ├── file_2_1
│   └─ file_2_2
├─ file1
└─ file2
```

vim即是给大家展示过的vim

如何使用vim? <https://www.runoob.com/linux/linux-vim.html>

现在需要大家合作完成一个类似的程序在任意一个系统上实现类似的功能。

要求

展示文件结构

可以像上面的简介中的展示通过字符画实现，也可以通过别的方法。

传递参数

第一个参数为待展示的文件夹/文件路径。程序需要展示参数文件夹下的所有内容。

简单整合这两个程序

实际上vim也会展示文件结构。但是我们希望将其变成类似tree的树形展示效果。

交互

- 程序需要提供交互，通过方向键移动控制"光标"选择文件或文件夹，如果选择文件则通过合适的软件打开，选择文件夹则显示被选择文件夹的结构。
- 比如在简介中出现的样例。

```
..
.
test
├─ dir1
│   ├── file_1_1
│   └─ file_1_2
├─ dir2
│   ├── file_2_1
│   └─ file_2_2
├─ file1
└─ file2
#选择dir1后内容变为
..
.
dir1
├─ file_1_1
```

└─ file_1_2

#选择file_1_1后打开文件，选择..后返回上一层

- 对于vim至少实现最基本的功能，创建文件，读写文件，文件命名等操作。

提示：在windows下使用getch();可以读取方向键输入。

额外内容

有额外内容可以添加。

知识点应用

这里提示仅供参考。

| 知识点 | 描述 |
|---------|------------------------------|
| 1. 数组 | 使用数组来保存文件结构。 |
| 2. 指针 | 熟练使用指针来减少代码量，增加代码复用性。 |
| 3. 结构体 | 文件结构使用树来保存，建议使用结构体和结构体指针来完成。 |
| 4. 文件读写 | 熟练掌握文件高效读写的方法，在使用w之后对数据进行保存。 |

注意事项

合作完成任务

- 本次作业由3-5人完成，注意分工配合，每一个函数前用注释加上作者的学号和姓名，格式如下

```
/**
 * @author 2020xxxxxx #?Name
 */
void Quick_sort(int * a,int (*Compare)(const int *a,const int *b)){
    /// something
}
```

- 建议由一位同学完成主要架构，设计函数的接口(参数)，然后由其他同学完成相应的子功能。
- 尽量多使用指针等特性来减少工作量。

只能使用c/c++语言完成

- 在**需要实现的功能**中主要处理流程使用c语言完成。

可以使用数据库等对数据进行保存，数据库的使用这里不提供指导，有需要可以自学。其余工具也一样。

- 操作系统这里不限制为windows，也可以在linux或mac系统上进行代码编写。
- c++可以使用stl

使用github进行代码同步管理

- 各自创建自己的github账号，选出一位主负责人建立项目。
- 其余各位通过fork加入同一个项目进行编写。
- 注意不能两人同时编写同一份文件，由负责人在github上进行代码整合。
- 注意分工，建立良好的项目结构。

基本计划

四周完成注意效率

1.第一周

- 讨论需求，写出要完成的目标到一个文档作为指导。
- 设计项目实施，分工。
- 建立github仓库，学会使用git命令管理github。
- 熟悉github基本操作。
- 建立基本项目框架。

2.第二周

- 各自实现自己的功能，合并代码，调试。
- 不时对需求进行修改，注意最开始设计代码可能会遇到的扩展。

3.第三周

- 实现各自的功能，合并代码，测试。
- 测试，修复bug。

4.第四周

- 深度测试。
- 准备汇报。

以上为预计的基本计划，根据实际需求进行更改。

代码审查，答辩

//最后由相关人员答辩展示代码(可执行文件)功能，每组大约5分钟时间，助教对代码进行审查。