程序设计分组训练实验五报告

刘梦真　黄键楠　王德阳

2022 年 10 月 16 日

目录

- 当你写下第一行代码之前

工程最终与用户交互的方式与结果是什么?

工程最终与用户交互的方式与结果是什么?

实验要求文档

程序运行提供菜单选择，菜单要求为循环菜单模式，只有输入功能项 0 时程序才退出，要求实现的菜单如下：... 当使用 1 号功能时... 当使用 2 号功能时...

工程最终与用户交互的方式与结果是什么?

实验要求文档

程序运行提供菜单选择，菜单要求为循环菜单模式，只有输入功能项 0 时程序才退出，要求实现的菜单如下：... 当使用 1 号功能时... 当使用 2 号功能时...

工程提供一个循环菜单，并根据用户输入分别实现单文档统计和多文档统计，并生成相应的网页！

工程最终与用户交互的方式与结果是什么?

从具体代码交互而言, 即如下图形式

交互指令

task2.exe <file path> <mode>

<generative html file in ../htmlResult>

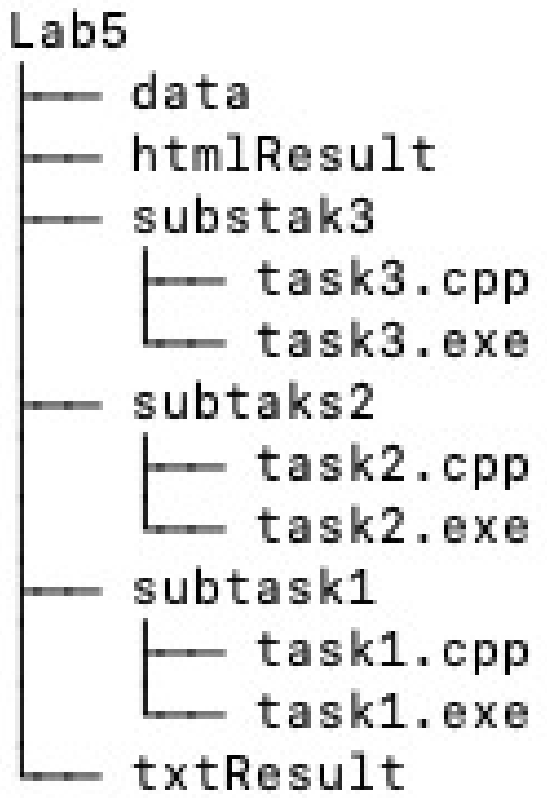
已生成 html, 路径为: (if mode1)

<open html> (if mode2)

好的文件架构能有效地互相调用子任务程序，使开发过程更顺利！

好的文件架构能有效地互相调用子任务程序，使开发过程更顺利！

本工程文件架构：



明确文件架构后，需要明确程序间的调用关系！本质上这里确定的是每个子任务的出入口，即接受什么文件 (参数)，生成什么文件 (提示/参数)。

明确文件架构后，需要明确程序间的调用关系！本质上这里确定的是每个子任务的出入口，即接受什么文件 (参数)，生成什么文件 (提示/参数)。

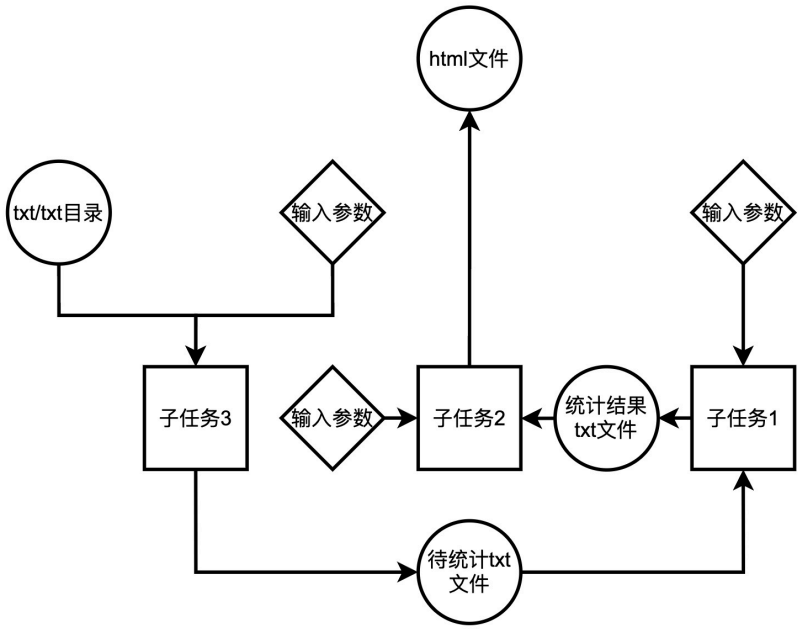
例如对于子任务 1，他接收一个待统计文件路径，一个任务模式，最后生成一个位于 txtResult 下的 txt 文件。

明确文件架构后，需要明确程序间的调用关系！本质上这里确定的是每个子任务的出入口，即接受什么文件 (参数)，生成什么文件 (提示/参数)。

这样做的好处在于：我们可以完全地将 task1.exe 抽象为一个黑箱。对于别的开发人员而言，他们无需关注子任务 1 的具体实现。一来便于开发，而来一旦发现整体 bug 就可以很好地定位出现问题的具体模块。

明确文件架构后，需要明确程序间的调用关系！本质上这里确定的是每个子任务的出入口，即接受什么文件 (参数)，生成什么文件 (提示/参数)。

该工程整体的程序调用关系如下图所示;



在具体的各子任务实现中，我们也将遵守类似的设计模式。

- 具体实现

任务期望

任务 1 需要接收待统计的英文文档以及工作模式，生成对应英文文档的统计文件

任务期望

任务 1 需要接收待统计的英文文档以及工作模式，生成对应英文文档的统计文件

因此我们期望任务 1 最终的交互形式为：

交互指令

task1.exe <file path> <mode>

<generate txt file in ../txtResult>

$(word, alpha, number, blank, other)count =$ (if mode1)

接口设计

基于要实现的功能，我们设计了如下两个接口：

- void main(int args,char *argv[])：任务入口，接收命令行参数并调用其他函数。

接口设计

基于要实现的功能，我们设计了如下两个接口：

- void main(int args,char *argv[])：任务入口，接收命令行参数并调用其他函数。
- void count(char *str)：文档统计模块，统计传入的 str 的相关字符数量。

文件命名规则

对于传入的文件，统一按照 filename+"CountResult" 的规则命名。

文件命名规则

对于传入的文件，统一按照 filename+"CountResult" 的规则命名。

e.g. DataMining.txt 的结果将被命名为:

"DataMiningCountResult.txt"

试运行

任务期望

任务 2 接收任务 1 生成的文档路径，以及工作模式，生成对应统计文档的 html 文件。

任务期望

任务 2 接收任务 1 生成的文档路径, 以及工作模式, 生成对应统计文档的 html 文件。
因此我们期望任务 2 最终的交互形式为:

交互指令

task2.exe <file path> <mode>

<generative html file in ../htmlResult>

已生成 html, 路径为: (if mode1)

<open html> (if mode2)

接口设计

基于要实现的功能，我们设计了如下三个接口：

- void main(int args,char *argv[])：任务入口，接收命令行参数并调用其他函数。

基于要实现的功能，我们设计了如下三个接口：

- void main(int args,char *argv[])：任务入口，接收命令行参数并调用其他函数。
- void ReadResult(FILE *fp, char * Filename)：读取文件函数，接收其中的有效信息并储存。

接口设计

基于要实现的功能，我们设计了如下三个接口：

- void main(int argv,char *argv[])：任务入口，接收命令行参数并调用其他函数。
- void ReadResult(FILE *fp, char * Filename)：读取文件函数，接收其中的有效信息并储存。
- void GenerateHtml(FILE *fp, char* FileName, char* mode):html 生成函数，根据获取的有效信息生成 html

文件命名规则

对于传入的文件，统一按照 filename+"htmlResult" 的规则命名。

文件命名规则

对于传入的文件，统一按照 filename+"htmlResult" 的规则命名。

e.g. DataMiningCountResult.txt 的结果将被命名为:

"DataMiningHtmlResult.txt"

试运行

任务期望

任务 3 接收工作模式以及待统计文档/文档目录，打开统计结果 html/汇总 html

任务期望

任务 3 接收工作模式以及待统计文档/文档目录，打开统计结果 html/汇总 html

因此我们期望任务 3 最终的交互形式为:

交互指令

task3.exe

"menu"

<mode>

<file path / dict path>

<open html>

为了保证提示信息的简洁性，任务 1 和 2 均使用静默模式!

接口设计

基于要实现的功能，我们设计了如下三个接口：

- void menu()：提供菜单

接口设计

基于要实现的功能，我们设计了如下三个接口：

- void menu()：提供菜单
- void one(char *p, char *p1, char *p2)：单个文档统计函数

接口设计

基于要实现的功能, 我们设计了如下三个接口:

- void menu(): 提供菜单
- void one(char *p, char *p1, char *p2): 单个文档统计函数
- void fileopen(char *path1, char *path2): 多文档统计函数

试运行

我们将引入命令行统计的方法对同一文档进行统计，并使用脚本自动比较。

测试结果显示我们的文档统计系统正确率 100%!

为了测试工程更大样本下的表现，我们将 Anton Chekhov 先生的《变色龙》作为测试数据 (我愿意相信各位都还记得)

为了测试工程更大样本下的表现，我们将 Anton Chekhov 先生的《变色龙》作为测试数据 (我愿意相信各位都还记得)

测试结果如下:

类型	数量
单词	1326
数字	5154
字母	0
空白	1358
其他	858

Thanks