

2017 初级实训总结

姓名：欧穗新

学号：16340173

班级：教务三班

目录

2017 初级实训总结	1
问题&解决	1
阶段一：	1
阶段二：	1
阶段三：	1
心得总结	2
一些意见和建议	2

问题&解决

在这一次实训当中，自己还是遇到了不少的问题：

阶段一：首先是 makefile 的编写，然后就是 storage 类和 Date 类的编写问题

解决方法：下载了相关资料进行阅读与学习，学会了 makefile 以及基本 Linux 操作；关于 Date 类检查（主要的问题存在于 Date 的构造函数也是需要考虑到 date 格式的合法性的），是通过和同学交流得出的解决方案；至于 Storage 类（这个问题主要在于自己 1.writeToFile 里面没有考虑到数据可能包含空格的情况，并且作大死的把 participator 先写到一个 string 里面再导出来；2.readFromFile 没有考虑到空格），改的欲仙欲死，一直到最后阶段 3 才改对，也是通过多方面与同学们的讨论得出的结果。

（在这里我不禁想要抱怨一句：无论是 date 的构造函数需要检查的坑还是 storage 的测试样例包含空格的情况，ta 都不给出一定的提示；同时评测也不给出任何测试样例，导致自己根本不知道错在哪；同时测评的报错信息还不一定就是直接指出程序出错的地方；以上的种种，无限增大了实训的难度，基本上过不过全靠运气，不禁让我怀疑实训方式的合理性。希望可以在以后的实训中有所改进（当然我知道我基本上是想多了））

阶段二：主要就是 AgendaService 类的编写这也是直到阶段三才改对的，这个基本上是自己一遍又一遍细致检查，考虑到尽可能多的问题，最后通过了测评的（这个是没有经过细致探讨或者请教 ta）

阶段三：扩展功能的编写，这里有三个我印象深刻的地方：

正则表达式的学习 ---- 博客（附上博客网址：<https://deerchao.net/tutorials/regex/regex.htm>） + [c++primer](#)；

再就是捕获 `rol^c`，附上代码：

```

#include "AgendaUI.hpp"
#include <iostream>

AgendaUI aui;
void Stop(int signo)
{
    aui.test.quitAgenda();
    std::cout << std::endl;
    exit(0);
}
int main () {
    signal(SIGINT, Stop);
    aui.OperationLoop();
    return 0;
}

```

这一部分可以学习 signal 函数+博客 (https://blog.csdn.net/nuaa_ybl/article/details/7091745);

最后就是一个 debug (肝到了四点 , , ,), 就是 service 删除 user 的时候 , 先删除 user 再退出所有的课程 (这时候把 dirty 标记为 1), 但是在退出所有课程的时候如果这个用户没有参加任何课程 , 就会把 dirty 重新置为 0 , 所以删除这种类型的用户的时候就会出现信息无法正常保存的情况。(解决方法是如果这个用户没有参加任何课程 , 那么删除课程的时候就不要改变 dirty)

心得总结

makefile 编写入门 ;

学会了 linux 常用的一些操作 ;

巩固了 gdb , valgrind , c++ (lambda 泛型算法) 等等知识 ;

了解到三层构架 (数据层 , 逻辑层 , 交互层)

捕获信号操作入门

正则表达式入门

一些意见和建议

实训遇到的坑太多太多 , 感觉十分的浪费时间而且没有太大的收获 , 感觉可能是得不偿失的 , 至于原因 : 无论是 date 的构造函数需要检查的坑还是 storage 的测试样例包含空格的情况 , ta 都不给出一定的提示 ; 同时评测也不给出任何测试样例 , 导致自己根本不知道错在哪 ; 同时测评的报错信息还不一定就是直接指出程序出错的地方 ;

以上的种种 , 无限增大了实训的难度 , 基本上过不过全靠运气 , 无论怎样改 , 怎样想破脑袋你也不知道到底测试数据是什么 , 到底他要我们干什么 , 不禁让我一度怀疑实训方式的合理性。

然后我们陷入了 “猜测-更改-测评报错” 的死循环中 , , , , , 我有一件 mmp 不知道当不当讲。

这样子个人觉得是浪费了许多不必要浪费的时间 (当然 , 我也不是那种随手能写出图形界面 , 花式扩展的大佬 , 所以只能老老实实地完成好阶段一阶段二)

这个不只是我一个人这么觉得，还有很多人都这么觉得，，，，

所以希望可以在以后的实训中有所改进（当然我知道我基本上是想多了）：应该要提供一些比较典型的，有代表性的测试样例啊啊啊啊；希望可以有比较正常，精准的报错；同时希望可以多提供一些测评机会（后面两个建议比起第一个到不显得那么迫切）。

以上所有，只是个人的一些意见，如有失偏颇，还请见谅。