



全学期解析班

科目: COMP3331/9331

进度: Assignment讲解





目录

CONTENTS

01

程序设计

02

数据结构

03

函数逻辑

04

错误捕获



1. 设计好Server和Client的程序结构
2. 设计Server和Client沟通的方式
3. 设计相关函数的逻辑
4. 编写/调试代码
5. 异常捕获
6. 文档注释/测试编写



1. Server

主线程: 接待连进来的客户端, 建立子线程
子线程*n : print客户端的状态/命令

2. Client

主线程: 发送消息
子线程: 接收(打印)消息



登录

逻辑:

```
client ---username-----> server
client <---username_is_exist--- server
client ---password-----> server
client <---login status----- server
```

Client Terminal	Server Terminal
>java Client 127.0.0.1 5000	>java Server 5000 destroyforum
Enter username: <u>Yoda</u>	Waiting for clients
Enter password: <u>sdrfdfs12</u>	Client connected
Invalid password	Incorrect password
Enter username: <u>Yoda</u>	
Enter password: <u>jedi*knight</u>	Yoda successful login



登录

如何读取credentials.txt?

新建用户名

>java Client 127.0.0.1 5000	
Enter username: <u>Obi-wan</u>	Client connected
Enter new password for Obi-wan: r2d2	New user
Enter one of the following commands: CRT,	Obi-wan successfully logged in

登录

已登录用户不允许再登陆..

Client 1 Terminal	Client 2 Terminal	Server Terminal
<pre>>java Client 127.0.0.1 6000 Enter username: <u>Yoda</u> Enter password: <u>jedi*knight</u> Welcome to the forum Enter one of the following commands: CRT, MSG, DLT, EDT, LST, RDT, UPD, DWN, RMV, XIT, SHT: (extra space added before user's response)</pre>	<pre>>java Client 127.0.0.1 6000 Enter username: <u>Yoda</u> Yoda has already logged in Enter username: <u>R2D2</u> Enter password: <u>c3p0sucks</u></pre>	<pre>>java Server 6000 destroyforum Waiting for clients Client connected Yoda successful login Client connected Yoda has already logged in R2D2 successful login</pre>



1. Threads 操作

CRT 创建thread

MSG 为thread添加消息

LST 显示所有threads

RDT 显示thread信息

EDT 修改

DLT&RMV 删除



2. IO 操作

UPD 上传文件

DWN 下载文件

3. 其他

XIT 退出

SHT 关闭论坛



函数逻辑

需要存储的东西:

1. Thread: name, owner
2. Message: content, owner, is_file

存储方法:

字典, 列表...

数据结构

Naïve dict:

建立简单

可能需要建立多个dict

JSON:

管理/读取简单

改动较麻烦



数据结构

1. Naïve dict:

```
{"Thread_name": ["message", "message2", ...]}
```

2. JSON-Like:

```
{"ThreadName": "Thread_name",  
  "Owner": "owner_a",  
  "Data": [  
    {  
      "Message": "message_1",  
      "Owner " : "owner_b",  
      "Number": 1  
    },  
    {  
      "Message": "message_2",  
      "Owner " : "owner_c",  
      "Number": 2  
    }  
  ]  
}
```



JSON <--> FILE 如何操作?

1. file to json

`json.load(fp)`

2. json to file

`json.dump(fp)`



异常捕获

- 方法:
使用 `try:...` `except:...`
- 注意:
所有函数都应该考虑异常处理.
- 例子 - 如何捕获:
发送文件/接收文件

- 注释格式 & 测试编写:

```
def example(attr1:int, attr2:str)->dict:
    """
    ~~~~~
    描述你的函数逻辑
    1. 第一步干嘛
    2. 第二步干嘛
    3. 第三步干嘛

    :param attr1: 描述这个参数的类型/意义
    :param attr2: 描述这个参数的类型/意义
    :return: 返回的东西是怎么样的

    >>> example(1, 'hello')
    {1: 'hello'}

    """
    return dict({attr1:attr2})

import doctest
doctest.testmod()
```

1. 多线程操作全局变量/文件可能会出错, 解决方案:
 - 使用线程锁 `threading.Lock()`
2. 切记提交前用Vlab运行一次



每周课程资料内容都会更新在Github : [LINK](https://github.com/lrlrlrlr/COMP9331_COMP3331_20T3)

https://github.com/lrlrlrlr/COMP9331_COMP3331_20T3

谢谢观看

DueApe – 让你的海外学习更简单