

实例 3：用户登录

登录系统通常分为普通用户与管理员权限，在用户登录系统时，可以根据自身权限进行选择登录。本实例要求实现一个用户登录的程序，该程序分为管理员用户与普通用户，其中管理员账号密码在程序中设定，普通用户的账号与密码通过注册功能添加。

实例目标

- 熟练地读取和写入数据
- 掌握目录与文件路径操作

实例分析

用户登录模块分为管理员登录和普通用户登录，在用户使用软件时，系统会先判断用户是否为首次使用：若是首次使用，则进行初始化，否则进入用户类型选择。用户类型分为管理员和普通用户两种，若选择管理员，则直接进行登录；若选择普通用户，先询问用户是否需要注册，若需要注册，先注册用户再进行登录。用户登录模块的具体流程如图 1 所示。

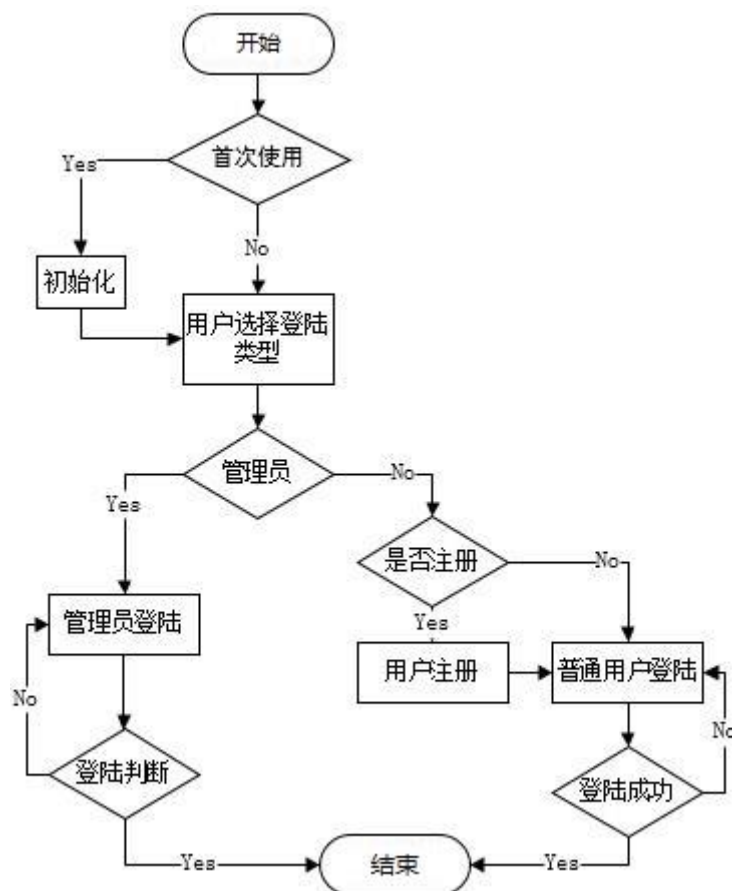


图1 用户登录流程图

用户登录模块包含的文件有：

- (1) 标识位文件 `flag.txt`。
- (2) 管理员账户文件 `u_root.txt`。
- (3) 普通用户账户文件。

标识位文件用于检测是否为初次使用系统，其中的初始数据为 0，在首次启动系统后将其数据修改为 1；管理员账户文件用于保存管理员的账户信息，该账户在程序中设置，管理员账户唯一；普通用户文件用于保存普通用户注册的账户，每个用户对应一个账户文件，普通用户账户被统一存储于普通用户文件夹 `users` 中。

实现用户登录模块可以编写如下函数实现：

- `is_first_start()`：判断是否首次使用系统；
- `init()`：数据初始化；
- `print_login_menu()`：打印登陆菜单；
- `user_select()`：用户选择；
- `root_login()`：管理员登陆；
- `user_register()`：用户注册；
- `user_login()`：普通用户登陆。

代码实现

创建一个 `user_login.py` 文件，在该文件中按实例分析依次定义各个函数：

1、`is_first_start()`

`is_first_start()` 函数首先用于判断是否为首次使用系统，为保证每次读取到的都为同一个标志位对象，我们需要一个标志位文件 `flag.txt`，将标志位对象的初始值 0 存储到文件中。每次启动系统后调用 `is_first_start()` 函数打开 `flag.txt` 文件，从其中读取数据，并进行判断。

此外 `is_first_start()` 函数还将根据标识文件的判断结果执行不同的分支：若是首次启动，则更改标志位文件内容、初始化资源、打印登陆菜单，之后进行用户选择；若不是首次启动，则直接打印登陆菜单，并接收用户选择。

`is_first_start()` 函数的实现如下：

```
# 判断是否首次使用系统

def is_first_start():

    if os.path.exists('flag.txt') == False:

        print('首次启动')

        flag = open('flag.txt', 'w+')

        flag.write('1')

        flag.close() # 关闭文件
```

```
init() # 初始化资源

print_login_menu() # 打印登录菜单

user_select() # 选择用户

else:

    flag = open('flag.txt', 'r')

    word = flag.read()

    if len(word) == 1:

        init() # 初始化资源

        print_login_menu() # 打印登录菜单

        user_select() # 选择用户
```

2、init()

初次启动系统时，需要创建管理员账户和普通用户文件夹，这两个功能都在 `init()` 函数中完成。`init()` 函数的实现如下：

```
# 初始化管理员

def init():

    if os.path.exists('users') == False:

        file = open('u_root.txt', 'w') # 创建并打开管理员账户文件

        root = {'rnum': 'root', 'rpwd': "123456"}

        file.write(str(root)) # 写入管理员信息

        file.close() # 关闭管理员账户文件

        os.mkdir('users') # 创建普通用户文件夹
```

3、print_login_menu()

`print_login_menu()` 函数用于打印登陆菜单，该菜单中有两个选项，分别为管理员登陆和普通用户登陆。`print_login_menu()` 函数的实现如下：

```
# 打印登录菜单

def print_login_menu():

    print('----用户登录----')

    print('1-管理员登陆')

    print('2-普通用户登陆')

    print('-----')
```

4、user_select()

在打印出登陆菜单后，系统应根据用户输入，选择执行不同的流程。此功能在 `user_select()` 函数中实现，该函数首先接受用户的输入，若用户输入“1”，则调用 `root_login()` 函数进行管理员登陆；若用户输入“2”，则先询问用户是否需要注册。`user_select()` 函数的实现如下：

```
# 用户选择

def user_select():

    while True:

        user_type_select = input('请选择用户类型')

        if user_type_select == '1':      # 管理员登陆验证

            root_login()

            break

        elif user_type_select == '2':    # 普通用户

            while True:

                select = input('是否需要注册? (y/n): ')

                if select == 'y' or select == 'Y':

                    print('----用户注册----')

                    user_register()      # 用户注册

                    break

                elif select == 'n' or select == 'N':

                    print('----用户登录----')

                    break

            else:

                print('输入有误，请重新选择')

            user_login()                  # 用户登录

            break

        else:

            print('输入有误，请重新选择')
```

6、root_login()

`root_login()` 函数用于实现管理员登陆，该函数可接收用户输入的账户和密码，将接收到的数据与存储在文件 `u_root` 中的管理员账户信息进行匹配，若匹配成功则提示登陆成功，并打印管理员功能菜单；若匹配失败则给出提示信息，并重新验证。

`root_login()` 函数的实现如下：

```
# 管理员登陆

def root_login():

    while True:

        print('****管理员登陆****')

        root_number = input('请输入账户名: ')

        root_password = input('请输入密码: ')

        file_root = open('u_root.txt', 'r') # 只读打开文件

        root = eval(file_root.read())      # 读取账户信息

        # 信息匹配

        if root_number == root['rnum'] and root_password == root['rpwd']:

            print('登陆成功! ')

            break

        else:

            print('验证失败')
```

7、user_register()

user_register()函数用于注册普通用户。当用户在 user_select()函数中选择需要注册用户之后，该函数被调用。user_register()函数可接收用户输入的账户名、密码和昵称，并将这些信息保存 to users 文件夹中与账户名同名的文件中。

user_register()函数的实现如下：

```
# 用户注册

def user_register():

    user_id = input('请输入账户名: ')

    user_pwd = input('请输入密码: ')

    user_name = input('请输入昵称: ')

    user = {'u_id': user_id, 'u_pwd': user_pwd, 'u_name': user_name}

    user_path = "./users/" + user_id

    file_user = open(user_path, 'w') # 创建用户文件

    file_user.write(str(user))      # 写入

    file_user.close()              # 保存关闭
```

8、user_login()

user_login()函数用于实现普通用户登陆，该函数可接收用户输入的账户名和密码，并将

账户名与 `users` 目录中文件列表的文件名匹配，若匹配成功，说明用户存在，进一步匹配用户密码，账户名和密码都匹配成功则提示“登陆成功”，并打印用户功能菜单。若账户名不能与 `users` 目录中文件列表的文件名匹配，则说明用户不存在。

`user_login()`函数的实现如下：

```
# 普通用户登录

def user_login():

    while True:

        print('****普通用户登录****')

        user_id = input('请输入账户名: ')

        user_pwd = input('请输入密码: ')

        # 获取 user 目录中所有的文件名

        user_list = os.listdir('./users') # 遍历元组，判断 user_id 是否在元组中

        flag = 0

        for user in user_list:

            if user == user_id:

                flag = 1

                print('登录中.....')

                # 打开文件

                file_name = './users/' + user_id

                file_user = open(file_name)

                # 获取文件内容

                user_info = eval(file_user.read())

                if user_pwd == user_info['u_pwd']:

                    print('登录成功! ')

                    break

        if flag == 1:

            break

        elif flag == 0:

            print('查无此人! 请先注册用户')

            break
```

至此，用户登陆模块所需的功能已全部实现。

需要注意的是，由于 `init()` 和 `user_login()` 函数中使用了 `os` 模块的 `listdir()` 函数，因此需在程序开头添加 `os` 模块，代码如下所示：

```
import os
```

此后在文件末尾添加如下代码，便可开始执行程序：

```
is_first_start()
```

以上代码用于调用 `is_first_start()` 函数，`is_first_start()` 函数是用户登陆模块的核心功能，也是其它各函数的入口。

代码测试

在程序所在文件夹中创建文件 `flag.txt`，并在其中写入数据“0”，保存退出。

首次运行程序，控制台打印如下信息：

```
首次启动！

----用户登录----

1-管理员登陆

2-普通用户登陆

-----

请选择用户类型：
```

此时查看程序所在目录，发现其中新建了文件夹 `users`、文件 `u_root`。在控制台输入“1”，进入管理员登陆界面，分别输入账户名 `root` 和密码 `123456`，程序的运行结果如下所示：

```
请选择用户类型：1

****管理员登陆****

请输入账户名：root

请输入密码：123456

登陆成功！
```

由以上运行结果可知，管理员的用户名和密码匹配成功。

再次运行程序，控制台打印如下信息：

```
----用户登录----

1-管理员登陆

2-普通用户登陆

-----

请选择用户类型：
```

由以上结果可知，`c_flag()` 函数调用成功。

本次选择使用普通用户登陆，并注册新用户，运行结果如下所示：

```
请选择用户类型：2

是否需要注册？(y/n)：y

----用户注册----

请输入账户名：itcast

请输入密码：123123

请输入昵称：黑马

****普通用户登陆****

请输入账户名：itcast

请输入密码：123123

登陆中....

登陆成功！
```

此时打开 `users` 目录，可以看到其中新建了名为“itcast”的文件。结合以上执行结果，可知用户注册、普通用户登录都已成功实现。