

BÀI TẬP TUẦN 1 – IT4062

Yêu cầu nộp bài:

- Tạo Makefile để biên dịch
- Đóng gói tất cả file mã nguồn và Makefile vào một file nén có tên theo định dạng HotenSV_MSSV_HW1.zip. Ví dụ: BuiTrongTung_20201234_HW1.zip
- Nộp bài theo quy định

Quản lý tài khoản người dùng

Giả sử tài khoản người dùng được lưu trong file có tên là account.txt, trong đó mỗi dòng có cấu trúc như sau:

username status

Trong đó giá trị status là 0 nếu tài khoản bị khóa, là 1 nếu tài khoản hoạt động.

Viết chương trình hiển thị menu như sau:

1. Log in
2. Post message
3. Logout
4. Exit

Các chức năng được thực hiện như sau:

- Chức năng 1: Đăng nhập. Người dùng nhập thông tin username từ bàn phím. Nếu tài khoản tồn tại và đang hoạt động thì thông báo đăng nhập thành công và ghi nhận trạng thái đã đăng nhập. Ngược lại báo đăng nhập thất bại. Nếu người dùng đã đăng nhập với một tài khoản, báo thất bại.
- Chức năng 2: Đăng bài. Hiển thị giao diện để người dùng nhập một câu từ bàn phím. Sau khi nhận câu người dùng nhập từ bàn phím, kiểm tra trạng thái đăng nhập. Nếu người dùng đã đăng nhập, báo đăng bài thành công. Ngược lại, báo đăng bài thất bại.
- Chức năng 3: Đăng xuất. Nếu trạng thái là chưa đăng nhập thì báo lỗi. Ngược lại hiển thị thông báo đăng xuất thành công và xóa trạng thái đã đăng nhập.
- Chức năng 4: Kết thúc chương trình.

Chương trình cần ghi lại nhật ký hoạt động vào file có tên log_MSSV.txt., ví dụ log_20201234.txt

Mỗi dòng có cấu trúc như sau:

[dd/mm/yyyy hh:mm:ss] \$ Chức năng lựa chọn \$ Giá trị người dùng cung cấp \$ Mã kết quả

Trong đó:

dd/mm/yyyy: Định dạng ngày nhận yêu cầu

hh:mm:ss: Định dạng thời điểm nhận yêu cầu

Mã kết quả: Biểu thị kết quả xử lý yêu cầu của người dùng

+OK: Thành công

-ERR: Có lỗi

Ví dụ:

```
[31/03/2023 14:42:24] $ 1 $ binhnv $ -ERR
[31/03/2023 14:46:24] $ 1 $ tungbt $ +OK
[31/03/2023 14:48:12] $ 1 $ admin $ -ERR
[31/03/2023 14:48:12] $ 2 $ Hello World! $ +OK
[31/03/2023 14:50:01] $ 3 $ $ +OK
[31/03/2023 14:50:47] $ 2 $ Bye bye$ -ERR
[31/03/2023 14:51:57] $ 3 $ $ -ERR
[31/03/2023 14:52:10] $ 4 $ $ +OK
```

*Lưu ý: File **account.txt** nằm cùng thư mục với file thực thi chương trình. Cần kiểm soát và bắt các lỗi vào ra file.*

Ví dụ file chứa thông tin tài khoản có nội dung như sau:

```
tungbt 1
levn 0
hungbt 1
```

Sau đây là một kịch bản kiểm thử minh họa các chức năng

Chức năng	INPUT	OUTPUT
1	Username: tung	Account is not exist
1	Username: levn	Account is banned
2	Post message: How are you?	You have not logged in.
3		You have not logged in.
1	Username: tungbt	Hello tungbt
1	Username: hungbt	You have already logged in
2	Post message: How are you	Successful post
3		Successful log out

Thang điểm:

- Chức năng 1: 4 điểm
- Chức năng 2: 1 điểm
- Chức năng 3: 1 điểm
- Ghi file log đúng định dạng và đủ thông tin: 1 điểm
- Tổ chức và trình bày mã nguồn: 3 điểm

Một số gợi ý về tổ chức, trình bày mã nguồn:

Các vấn đề về tổ chức, trình bày mã nguồn

- Clean code(tóm tắt): <https://viblo.asia/p/tom-tat-cuon-clean-code-cua-uncle-bob-6J3Zg07MImB>
- Clean code(đầy đủ): <https://toihocdesignpattern.com/dich-sach/clean-code>
- Sử dụng quy ước định danh. Ví dụ:

- Camel case: https://en.wikipedia.org/wiki/Camel_case
- Snake case: https://en.wikipedia.org/wiki/Snake_case
- Quy ước khác: <https://viblo.asia/p/naming-rules-cac-quy-tac-vang-trong-lang-dat-ten-ByEZkMXE5Q0>

- Comment cho các hàm mô-đun theo quy tắc thống nhất. Ví dụ:

```
/**
 * @function splitString: Split a string into one alphabetic string and one numeric string.
 *
 * @param source: A pointer to a input string.
 * @param letters: A pointer to a string contains all of letters in source string.
 * @param numbers: A pointer to a string contains all of numbers in source string.
 *
 * @return: 0 if success.
 *         1 if an error occurs because source string has any special character.
 */
int splitString(char *source, char *letters, char *numbers);
```