

Socket.winsock2

Chụp màn hình

Folder quản lý: SCREEN_SHOT

- Chạy Server:

```
cd ..SCREEN_SHOT\Server
make; ./server
```

- Chạy Client:

```
cd ..SCREEN_SHOT\Client
make; ./client
```

Quy trình:

1. Client gửi yêu cầu (*chưa làm*)
2. Server nhận yêu cầu (*chưa làm*)
3. Server xử lý:

```
// chụp màn hình và gửi về Client
ScreenShotAndSendToClient(ac_socket, filename) {

    // chụp và lưu ảnh về file
    SaveScreenShotToFile(filename);

    // mở file và truyền về cho Client
    ...
}
```

Trong đó:

ac_socket: socket đang kết nối với Client
filename: tên (link) file lưu ảnh ở Server (.bmp)

4. Client nhận ảnh:

```
// nhận dữ liệu từ Server và lưu về file ảnh
SaveScreenShotToFile(m_socket, filename);
```

Trong đó:

m_socket: socket đang kết nối với Server

filename: tên (link) file lưu ảnh ở Client (.bmp)

Bắt phím nhấn

Folder quản lý: KEYDOWN

- Chỉ làm tính năng bắt phím nhấn chứ chưa truyền dữ liệu qua socket
- Chạy file:

```
cd ../KEYDOWN\  
g++ keydown.cpp -o keydown.exe; ./keydown
```

Giải thích:

Sau khi chạy file, các phím nhấn sẽ được bắt và nội dung hiển thị tùy vào từng hàm

- Bắt các phím là chữ cái, phân biệt chữ hoa/chữ thường

```
checkAlphabet();
```

- Bắt các phím là số, không bắt được bàn phím số bên phải (nếu có)

```
checkNumber();
```

- Kết hợp 2 hàm trên

```
catchKeydown();
```

- Bắt phím nhấn và chuyển về kí tự (bằng hàm toUnicode)

```
catchAlltoUnicode() {  
    ...  
  
    // in ra thông báo về kí tự tương ứng  
    toUni(int code)  
  
    ...  
}
```

- Bắt phím nhấn và xuất description **(Nên dùng)**

```
catchCodeAlltoString() {  
    ...  
  
    // trả về dưới dạng string (Alt, F5, A, 1, ...)  
    string desc = VirtualKeyCodeToString(UCHAR virtualKey);  
  
    ...  
}
```

Xử lý cây thư mục

Folder quản lý: FOLDERTASK

- Các hàm xử lý được đưa vào 1 class tên FolderTask

```
#include "FolderTask.h"
```

- Chạy file:

```
cd ../FOLDERTASK\  
g++ *.cpp -o main.exe -lws2_32; ./main
```

Giải thích:

```
// kiểm tra file trong path đã tồn tại chưa  
bool isExistFile(const char* path);  
  
// xóa file với path chứa tên file  
bool deleteFile(const char* path);  
  
// xóa folder với path đã cho  
bool deleteFolder(const char *path);  
  
// liệt kê con trực tiếp của folder (path)  
vector<string> list_files(const char *path);  
  
// liệt kê tất cả con của folder (path)  
vector<string> recursive_list_files(const char *path);  
  
// tạo file (tên file nằm trong path)  
bool createFile(const char* path);
```

```
// tạo folder (tên folder nằm trong path)
bool createFolder(const char* path);

// đổi tên
// oldName và newName có đường dẫn giống nhau
// chỉ khác tên
bool renameFileFolder(const char* oldName, const char* newName);

// copy file
bool copyFile(const char* sourcePath, const char* destPath);

// copy folder
bool copyFolder(const char* sourcePath, const char* destPath);
```

Thử gửi qua socket: FOLDER

- Server gửi Client: vector<string> nhận được thông báo gửi đủ, không mất dữ liệu
- Client nhận từ Server: nhận không toàn vẹn