

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN



## *BÁO CÁO ĐỒ ÁN* **CƠ SỞ NGÀNH MẠNG**

- Đề tài 1** : Tìm hiểu kỹ thuật lập trình  
Hook, xây dựng ứng dụng bắt  
phím KeyLogger
- Đề tài 2** : Xây dựng ứng dụng hỗ trợ  
quản lý công việc theo lịch cho  
công ty bằng giao thức TCP/IP

**GVHD** : ThS. Nguyễn Thế Xuân Ly  
**SVTH** : Mai Đình Chiến  
**Nhóm HP** : xx.91  
**Lớp SH** : 15T2  
**Mã SV** : 102150086

*Đà Nẵng, tháng 06 năm 2020*

## LỜI NÓI ĐẦU

Ngày nay, hầu như mọi công việc hàng ngày liên quan đến cuộc sống của chúng ta đều diễn ra trên máy tính. Từ việc soạn thảo văn bản, gửi nhận thông tin đến việc tra cứu, truy cập thông tin từ hệ thống mạng máy tính tồn cầu Internet đối với người sử dụng là công việc thường ngày và rất phổ biến.

Về bài toán tìm hiểu về kỹ thuật Hook, đây là một kỹ thuật rất hay và bổ ích. Tuy nhiên, nó là một con dao hai lưỡi, sau khi tìm hiểu em đã hiểu về kỹ thuật hook, ứng dụng của kỹ thuật hook và đồng thời cũng biết cách phòng ngừa khỏi các ứng dụng của hook như KeyLogger để bảo vệ thông tin cá nhân.

Em đã được tham gia vào một ý tưởng thực tế về xây dựng một ứng dụng Hỗ trợ quản lý công việc theo lịch cho công ty trong mạng LAN. Ý tưởng này áp dụng các kiến thức về Lập trình mạng em đã được học trong thời gian này.

Em xin chân thành cảm ơn sự hướng dẫn tận tình của thầy NGUYỄN THẾ XUÂN LY, cảm ơn sự trao đổi, góp ý của các bạn trong lớp đã giúp em hoàn thành đề tài này.

Với sự hạn chế về mặt kiến thức nên báo cáo của em vẫn còn nhiều thiếu sót và hạn chế, vậy em rất mong nhận được sự góp ý thêm của quý thầy cô và các bạn.

*Đà Nẵng, ngày 13 tháng 06 năm 2020*

**Sinh viên thực hiện**

Mai Đình Chiến

## MỤC LỤC

DANH SÁCH HÌNH VẼ .....	3
PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH .....	4
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	4
1.1. Tìm hiểu về KeyLogger .....	4
1.1.1. Khái niệm.....	4
1.1.2. Phân loại về KeyLogger .....	4
1.1.3. Cách hoạt động của KeyLogger .....	5
1.1.4. Cài đặt Hook.....	5
1.1.5. Một số hàm liên quan đến hook.....	6
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	8
2.1. Yêu cầu bài toán. ....	8
2.2. Giải pháp .....	8
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ .....	9
3.1. Môi trường triển khai và chọn ngôn ngữ .....	9
3.2. Kết quả thực thi chương trình.....	9
3.3. Ưu nhược điểm.....	10
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	11
4.1. Kết luận .....	11
4.2. Hướng phát triển .....	11
PHẦN II: LẬP TRÌNH MẠNG .....	12
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	12
1.1. Giao thức TCP/IP. ....	12
1.1.1. Tổng quan.....	12
1.1.2. Phương thức hoạt động của bộ giao thức TCP/IP.....	13
1.2. Lập trình Socket. ....	14
1.2.1. Khái niệm Socket và cổng port.....	14
1.2.2. Nguyên lý hoạt động .....	15
1.2.3. Socket trong C#.....	16
1.3. Mô hình Client/Server.....	16

<b>CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....</b>	<b>19</b>
<b>2.1. Yêu cầu bài toán .....</b>	<b>19</b>
<b>2.2. Xây dựng chương trình.....</b>	<b>19</b>
<b>2.2.1. Client.....</b>	<b>19</b>
<b>2.2.2 Server .....</b>	<b>20</b>
<b>CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ .....</b>	<b>21</b>
<b>3.1. Môi trường triển khai và chọn ngôn ngữ .....</b>	<b>21</b>
<b>3.2. Kết quả thực thi chương trình.....</b>	<b>21</b>
<b>3.3. Ưu nhược điểm.....</b>	<b>25</b>
<b>CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>25</b>
<b>4.1. Kết luận .....</b>	<b>25</b>
<b>4.2. Hướng phát triển .....</b>	<b>26</b>
<b>PHẦN III. KẾT LUẬN CHUNG .....</b>	<b>27</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>28</b>

**DANH SÁCH HÌNH VẼ**

Hình 1. Sơ đồ thuật toán keylogger .....	8
Hình 2. Giao diện chương trình keylogger .....	9
Hình 3. Kết quả thu được của chương trình gồm file log + hình ảnh chụp màn hình ....	9
Hình 4. Nội dung của file log gồm các ký tự được nhập từ bàn phím KeyLogger thu thập được .....	10
Hình 5. Kết quả thu thập được gửi qua email .....	10
Hình 6. Quá trình đóng mở gói dữ liệu trong TCP/IP .....	13
Hình 7. Cấu trúc dữ liệu trong TCP/IP .....	14
Hình 8. Mô hình mạng OSI .....	15
Hình 9. Mô hình client/server giai đoạn 1 .....	16
Hình 10. Mô hình client/server giai đoạn 2 .....	17
Hình 11. Mô hình client/server giai đoạn 3 .....	17
Hình 12. Mô hình client/server giai đoạn 4 .....	18
Hình 13. Mô hình Client-Server .....	19
Hình 14. Giao diện Server .....	21
Hình 15. Giao diện login .....	21
Hình 16. Giao diện Client .....	22
Hình 17. Xem công việc trong ngày .....	22
Hình 18. Chi tiết công việc .....	23
Hình 19. Phân công công việc cho người dùng .....	24
Hình 20. Server thông báo cho người dùng về công việc được nhận .....	24
Hình 21. Màn hình lịch của client là leader .....	25

## PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH

**TIÊU ĐỀ:** *Tìm hiểu kỹ thuật lập trình Hook, xây dựng ứng dụng bắt phím KeyLogger*

### CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

#### 1.1. Tìm hiểu về KeyLogger

##### 1.1.1. Khái niệm

KeyLogger hay “Trình theo dõi thao tác bàn phím” là một chương trình máy tính ban đầu được viết nhằm mục đích theo dõi và ghi lại mọi thao tác thực hiện trên bàn phím vào một tập trình nhật ký (log) để cho người cài đặt nó sử dụng. Vì chức năng mang tính vi phạm vào riêng tư của người khác nên các trình keylogger được xếp vào nhóm các phần mềm gián điệp.

Về sau khi keylogger phát triển cao hơn, nó không những ghi lại thao tác bàn phím mà còn ghi lại cả các hình ảnh hiển thị trên màn hình (screen) bằng cách chụp (screen-shot) hoặc quay phim thậm chí ghi nhận cách con trỏ chuột trên máy tính di chuyển.

##### 1.1.2. Phân loại về KeyLogger

KeyLogger bao gồm hai loại, một loại là KeyLogger phần cứng và một loại là phần mềm.

Thực tế hiện nay loại KeyLogger phần mềm được sử dụng nhiều hơn so với KeyLogger phần cứng. Theo những người lập trình, keyLogger viết ra với chỉ có một mục đích là giúp các bạn giám sát con cái, người thân xem họ làm gì với PC, với internet, khi chat với người lạ. Nhưng cách sử dụng và chức năng của KeyLogger hiện tại trên thế giới khiến người ta thường hay phân loại KeyLogger theo mức độ nguy hiểm bằng các câu hỏi:

- Nhiễm vào máy không qua cài đặt hoặc cài đặt vào máy nhanh?
- Có thuộc tính ẩn hoặc giấu trên trình quản lý tiến trình và trình cài đặt và gỡ bỏ chương trình?
- Theo dõi không thông báo hoặc máy tính bị nhiễm khó tự phát hiện?
- Có thêm chức năng Captures Screen hoặc ghi lại thao tác chuột?
- Khó tháo gỡ.
- Có khả năng lây nhiễm, chống tắt?

Cứ mỗi câu trả lời “có”, sẽ cho một điểm. Điểm càng cao, KeyLogger càng vượt khỏi mục đích giám sát đến với mục đích do thám và tính nguy hiểm nó càng cao.

KeyLogger có thể được phân loại theo số điểm:

- Loại số 1 (0 điểm): KeyLogger loại bình thường; chạy công khai, có thông báo cho người bị theo dõi, đúng với mục đích giám sát.
- Loại số 2 (1-2 điểm): KeyLogger nguy hiểm, KeyLogger loại này sẽ chạy ngầm, hướng đến mục đích do thám nhiều hơn là giám sát (nguy hại đến các thông tin cá nhân).
- Loại số 3 (3-5 điểm): KeyLogger loại này rất nguy hiểm, ẩn giấu hoàn toàn theo dõi trên một phạm vi rộng, mục đích do thám rõ ràng.
- Loại số 4 (6 điểm): KeyLogger nguy hiểm nghiêm trọng, thường mang theo bởi các trojan-virus cực kỳ khó tháo gỡ, là loại KeyLogger nguy hiểm nhất.

### 1.1.3. Cách hoạt động của KeyLogger

Thông thường, một chương trình KeyLogger sẽ gồm có ba phần chính:

- Chương trình điều khiển: dùng để điều phối hoạt động, tinh chỉnh các thiết lập, xem các tập tin nhật ký cho KeyLogger. Phần này là phần được giấu kỹ nhất của KeyLogger, thường chỉ có thể gọi ra bằng một tổ hợp phím tắt đặc biệt.
- Tập tin hook, hoặc là chương trình monitor dùng để ghi nhận lại các thao tác bàn phím, capture screen (đây là phần quan trọng nhất).
- Tập tin nhật ký(log), nơi chứa đựng hoặc ghi lại toàn bộ những gì hook ghi nhận được.

Trong một hệ thống (Window, Linux, Mac,...), khi bấm 1 phím trên bàn phím, bàn phím sẽ chuyển nó thành tín hiệu chuyển vào CPU, CPU sẽ chuyển nó tới hệ điều hành để hệ điều hành dịch thành chữ hoặc số cho chính nó hoặc các chương trình khác sử dụng.

Nhưng khi trong hệ thống đó có KeyLogger, không những chỉ có hệ điều hành theo dõi mà cả hook file của KeyLogger theo dõi, nó sẽ ghi nhận và dịch lại các tín hiệu ghi vào tập tin nhật ký. Đồng thời nó có thể theo dõi màn hình và thao tác chuột.

### 1.1.4. Cài đặt Hook

Đặt một thủ tục hook bằng cách gọi hàm **SetWindowsHookEx** đặc tả kiểu hook gọi thủ tục, thủ tục có phải được tổ chức với tất cả các thread hay chỉ được tổ chức với một thread cụ thể và một pointer chỉ tới một điểm vào thủ tục. Phải đặt một thủ tục hook tồn cục vào một DLL riêng biệt từ ứng dụng cài đặt thủ tục hook. Ứng dụng cài đặt phải có handle chỉ tới module DLL trước khi nó có thể đặt thủ tục hook. Hàm **GetModuleHandle** khi được đưa tên của module sẽ trả handle của module đó. Cuối cùng sử dụng hàm **SetWindowsHookEx** để đặt địa chỉ hook vào trong chuỗi hook dành riêng. **SetWindowsHookEx** chuyển handle module, một pointer chỉ tới điểm vào thủ tục hook và cho danh hiệu thread, chỉ ra rằng thủ tục hook phải được tổ chức

với tất cả các thread trong hệ thống.

### 1.1.5. Một số hàm liên quan đến hook

#### 1.1.5.1. SetWindowsHookEx

- Công dụng: cài đặt 1 hook procedure vào một hook chain.
- Cú pháp:

```
HHOOK SetWindowsHookEx(idHook, lpfn, hMod, dwThreadId)
    int idHook;
    HOOKPROC lpfn;
    HINSTANCE hMod;
    DWORD dwThreadId;
```

- Hàm SetWindowsHookEx cài đặt một hàm hook application-defined vào trong một chuỗi hook. Hàm này là version mở rộng của hàm SetWindowsHook.

- Thông số:

idHook: loại hook, bao gồm các giá trị sau:

lpfn: một con trỏ đến hook procedure. Nếu là global hook thì tham số này phải trỏ đến một hook procedure trong một DLL. Ngược lại thì có thể trỏ đến một đoạn mã đóng vai trò hook procedure trong process hiện tại.

hMod: handle của DLL chứa hook procedure. Trong trường hợp local hook thì tham số này được đặt là NULL.

dwThreadId: định danh của thread mà hook procedure sẽ gắn vào. Nếu giá trị này là 0 thì mọi thread sẽ bị ảnh hưởng (global), ngược lại thì chỉ có thread được xác định là bị ảnh hưởng (local).

#### 1.1.5.2. Hàm gỡ bỏ một hàm hook UnhookWindowsHookEx

- Công dụng: gỡ cài đặt 1 hook procedure khỏi một hook chain được cài đặt bởi SetWindowsHookEx.

- Cú pháp:

```
BOOL UnhookWindowsHookEx(hhook)
    HHOOK hhook;
```

- Hàm UnhookWindowsHookEx bỏ đi một hàm hook application ra khỏi một chuỗi hàm hook. Một hàm hook xử lý các sự kiện trước khi chúng được gửi tới vòng lặp thông điệp ứng dụng trong hàm WinMain.

- Thông số:

hhook Chỉ ra hàm hook được dỡ bỏ. Đây là giá trị được trả về bởi hàm SetWindowsHookExIdentifies khi hàm hook được cài đặt.

- Chú ý: Hàm UnhookWindowsHookEx phải được sử dụng trong sự kết hợp với hàm SetWindowsHookEx.

#### 1.1.5.3. Hàm gọi hook kế tiếp trong hook-chain CallNextHookEx:

- Công dụng: chuyển quyền điều khiển cùng các thông tin hook cho hook procedure kế tiếp trong hook chain.



- Cú pháp:

LRESULT CallNextHookEx(hHook, nCode, wParam, lParam)

HHOOK hHook;

int nCode;

WPARAM wParam;

LPARAM lParam;

- Hàm CallNextHookEx function chuyển các thông tin hook đến hàm xử lý hook kế tiếp trong hook chain.

- Thông số:

hHook	Chỉ định hook handle.
nCode	Chỉ định hook code để gọi đến hook kế tiếp. Hàm xử lý hook dùng giá trị này để chỉ định xử lý thông điệp được gọi từ hook thế nào.
wParam	Chỉ định 16 bits thông tin mở rộng của thông điệp.
lParam	Chỉ định 32 bits thông tin mở rộng của thông điệp.

*Bảng 1. Chức năng của các biến trong hàm CallNextHookEx*

- Giá trị trả về: Giá trị trả về là kết quả của quá trình xử lý và tùy thuộc vào thông số nCode.

## CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

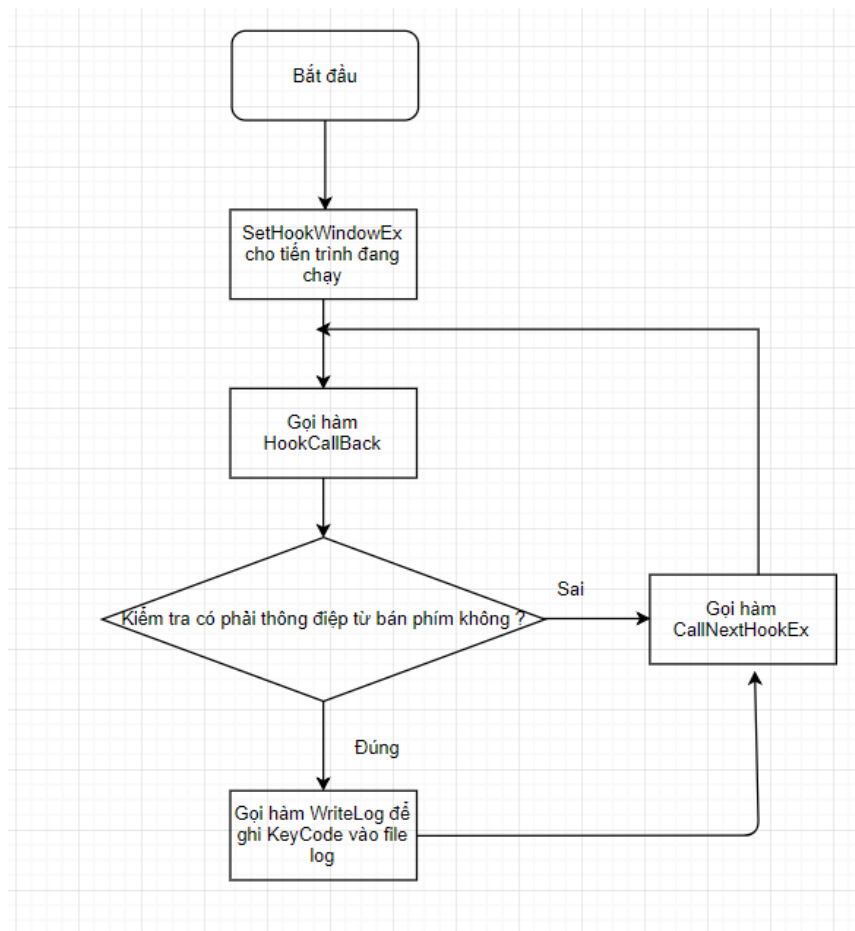
### 2.1. Yêu cầu bài toán.

Mục đích yêu cầu của đề tài là xây dựng một chương trình theo dõi thao tác bàn phím. Có nghĩa là tại bất kỳ thời điểm nào chương trình đang được chạy thì sẽ ghi lại tất cả các phím được nhấn trong các tiến trình đang chạy trong hệ thống. Khi nhấn phím thì bàn phím sẽ chuyển nó thành tín hiệu chuyển vào CPU, CPU sẽ chuyển nó tới hệ điều hành để hệ điều hành dịch thành chữ hoặc số. Ý tưởng nảy ra là sẽ tạo bản sao của tín hiệu đó và ghi lại vào nhật ký (log).

### 2.2. Giải pháp

Để thực hiện được các ý tưởng trên, cần phải thực hiện các việc sau:

- Thả hook vào tất cả các tiến trình đang chạy. Ghi nhận tất cả các thông tin khi xảy ra sự kiện nhấn phím(key up/ key down).
- Có một đoạn chương trình ghi nhận tín hiệu đó và ghi dữ liệu nhận được vào file log.



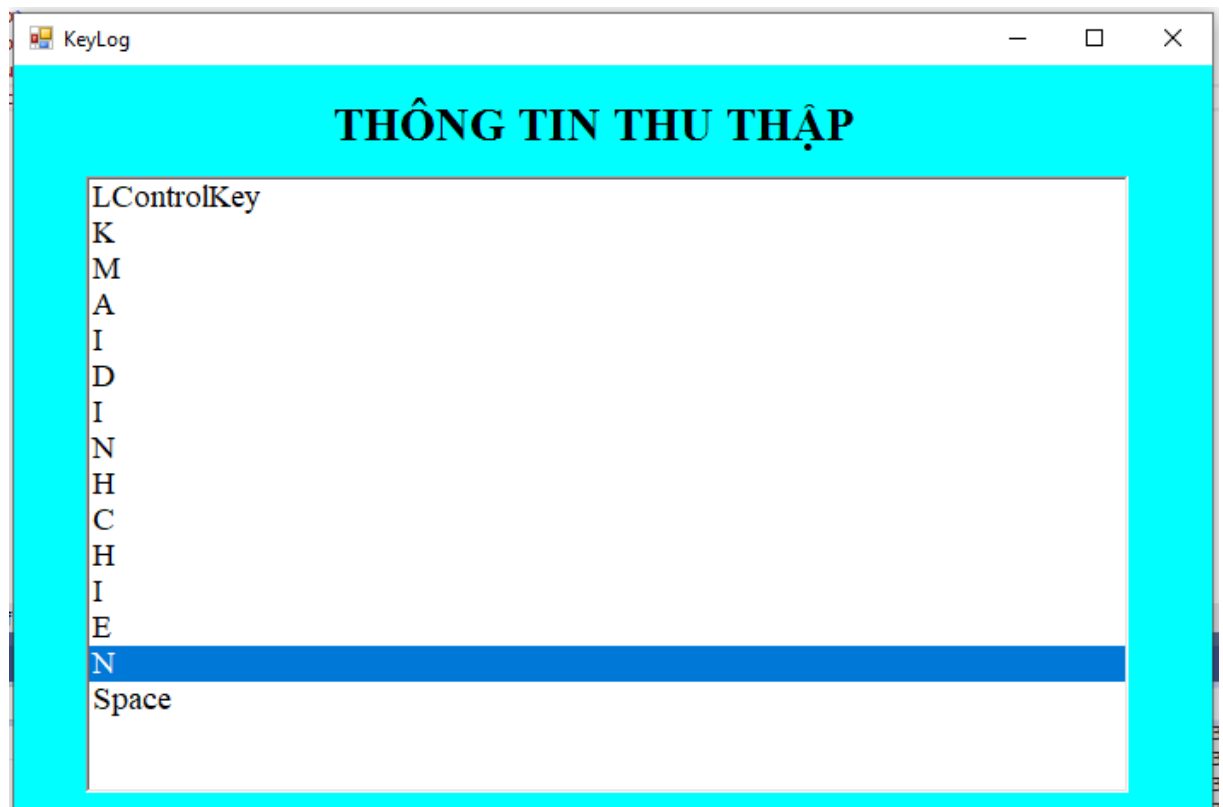
Hình 1. Sơ đồ thuật toán keylogger

## CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1. Môi trường triển khai và chọn ngôn ngữ

- Trình biên dịch: Visual Studio 2019
- Ngôn ngữ: C#

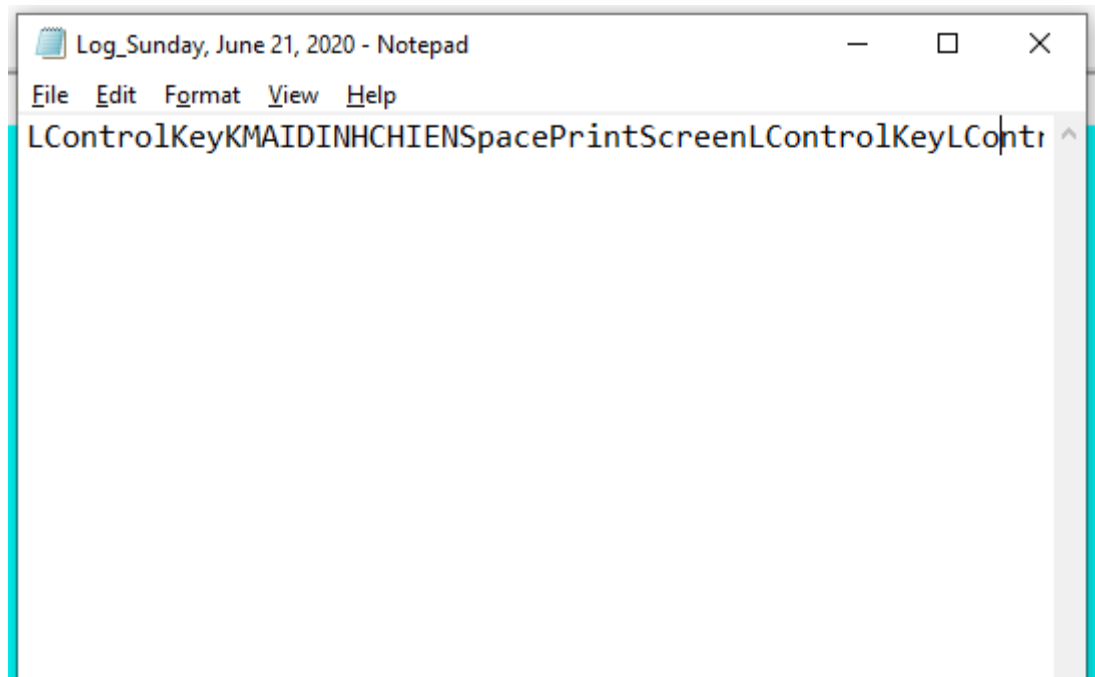
### 3.2. Kết quả thực thi chương trình



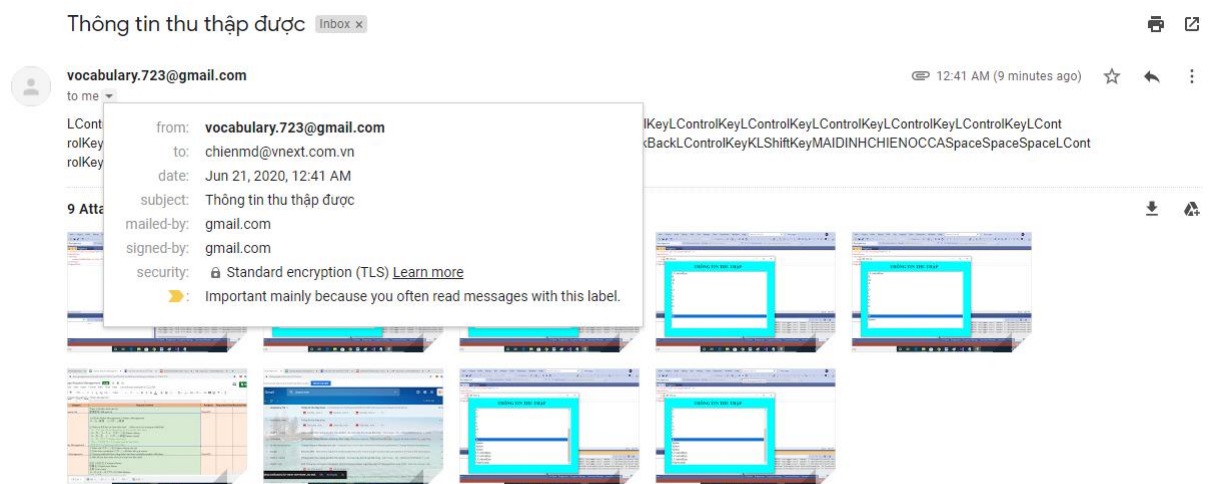
Hình 2. Giao diện chương trình keylogger

app.publish	6/21/2020 12:31 AM	File folder	
Image_Sunday, June 21, 2020	6/21/2020 12:39 AM	File folder	
KeyLogger	6/21/2020 12:31 AM	Application Manif...	2 KB
KeyLogger	6/20/2020 6:36 PM	Application	12 KB
KeyLogger.exe.config	6/21/2020 12:31 AM	XML Configuratio...	1 KB
KeyLogger.exe.manifest	6/21/2020 12:31 AM	MANIFEST File	4 KB
KeyLogger.pdb	6/20/2020 6:36 PM	Program Debug D...	26 KB
Log_Sunday, June 21, 2020	6/21/2020 12:42 AM	Text Document	1 KB

Hình 3. Kết quả thu được của chương trình gồm file log + hình ảnh chụp màn hình



Hình 4. Nội dung của file log gồm các ký tự được nhập từ bàn phím KeyLogger thu thập được



Hình 5. Kết quả thu thập được gửi qua email

### 3.3. Ưu nhược điểm

- Ưu điểm:
  - Thu thập được thông tin bàn phím.
  - Gửi được qua email người cài keylogger.
  - Chụp ảnh màn hình.
  - Khởi động cùng hệ điều hành.
  - Tạo hotkey hiện ẩn chương trình.
- Nhược điểm:

- Chỉ gửi được mail khi có mạng.
- Chưa ẩn được chương trình khỏi task manager

## **CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **4.1. Kết luận**

Chương trình đã ghi nhận được thao tác bàn phím trong tất cả tiến trình đang hệ thống và sao lưu lại dữ liệu vào một file, một số tính năng phụ của chương trình như :

- Chụp ảnh màn hình.
- Gửi dữ liệu thu thập được tới mail của người tạo.
- Khởi động cùng hệ điều hành.
- Tạo hotkey hiện ẩn chương trình.

### **4.2. Hướng phát triển**

- Xây dựng giao diện thân thiện với người dùng.
- Ẩn chương trình khỏi sự quản lý của Task Manager.
- Chương trình có thể hoạt động trên các hệ điều hành khác ngoài Windows.
- Nâng cao tính năng gửi gmail trong việc gửi số lượng ảnh lớn từ máy có cài keylogger tới gmail khác.

## **PHẦN II: LẬP TRÌNH MẠNG**

**TIÊU ĐỀ:** *Xây dựng ứng dụng hỗ trợ quản lý công việc theo lịch cho công ty*

### **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

#### **1.1. Giao thức TCP/IP.**

##### **1.1.1. Tổng quan.**

TCP/IP là bộ giao thức cho phép kết nối các hệ thống mạng không đồng nhất với nhau.

TCP/IP được xem như giản lược của mô hình tham chiếu OSI với 4 tầng như sau:

- Tầng liên kết (Datalink Layer)
- Tầng mạng (Internet layer)
- Tầng giao vận (Transport Layer)
- Tầng Ứng Dụng (Application Layer)

##### **a. Tầng liên kết:**

Tầng liên kết (còn được gọi là tầng liên kết dữ liệu hay tầng giao tiếp mạng) là tầng thấp nhất trong mô hình TCP/IP, bao gồm các thiết bị giao tiếp mạng và các chương trình cung cấp các thông tin cần thiết để có thể hoạt động, truy nhập đường truyền vật lý qua các thiết bị giao tiếp mạng đó.

##### **b. Tầng mạng:**

Tầng Internet ( hay còn gọi là tầng Mạng) xử lý quá trình truyền gói tin trên mạng, các giao thức của tầng này bao gồm : IP ( Internet Protocol) , ICMP ( Internet Control Message Protocol) , IGMP ( Internet Group Message Protocol )

##### **c. Tầng giao vận:**

Tầng giao vận phụ trách luồng dữ liệu giữa 2 trạm thực hiện các ứng dụng của tầng trên, tầng này có 2 giao thức chính là TCP ( Transmission Control Protocol) và UDP (User Datagram Protocol).

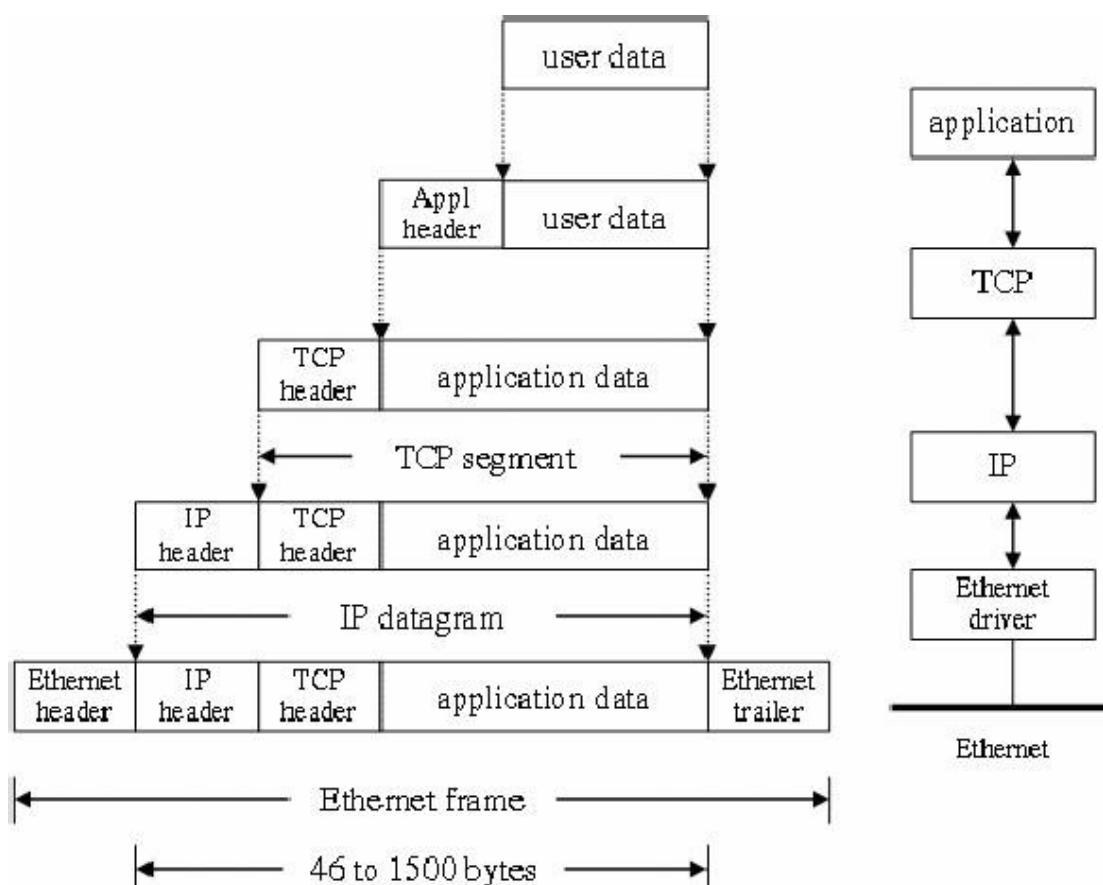
- TCP cung cấp luồng dữ liệu tin cậy giữa 2 trạm, nó sử dụng các cơ chế như chia nhỏ các gói tin ở tầng trên thành các gói tin có kích thước thích hợp cho tầng mạng bên dưới, báo nhận gói tin, đặt hạn chế thời gian timeout để đảm bảo bên nhận biết được các gói tin đã gửi đi. Do tầng này đảm bảo tính tin cậy nên tầng trên sẽ không cần quan tâm đến nữa.

- UDP cung cấp một dịch vụ rất đơn giản hơn cho tầng ứng dụng. Nó chỉ gửi dữ liệu từ trạm này tới trạm kia mà không đảm bảo các gói tin đến được tới đích. Các cơ chế đảm bảo độ tin cậy được thực hiện bởi tầng trên tầng ứng dụng.

#### d. Tầng Ứng Dụng:

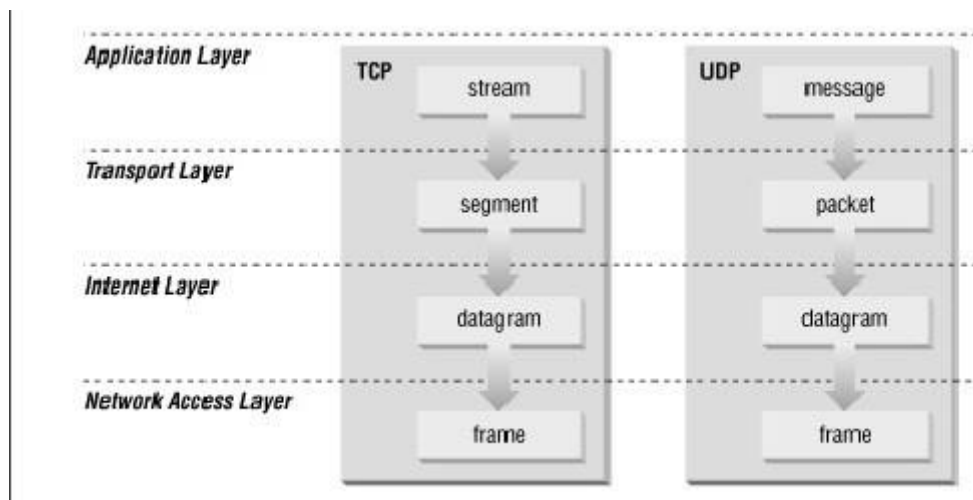
Là tầng trên của mô hình TCP/IP bao gồm các tiến trình và các ứng dụng cung cấp cho người sử dụng để truy cập mạng. Có rất nhiều ứng dụng được cung cấp trong tầng này, mà phổ biến là Telnet: sử dụng trong việc truy cập mạng từ xa, FTP (File Transport Protocol) dịch vụ truyền tệp tin, EMAIL: dịch vụ truyền thư tín điện tử. WWW (Word Wide Web).

#### 1.1.2. Phương thức hoạt động của bộ giao thức TCP/IP.



Hình 6. Quá trình đóng mở gói dữ liệu trong TCP/IP

Cũng tương tự như trong mô hình OSI, khi truyền dữ liệu, quá trình tiến hành từ tầng trên xuống tầng dưới, qua mỗi tầng dữ liệu được thêm vào thông tin điều khiển gọi là Header. Khi nhận dữ liệu thì quá trình xảy ra ngược lại. dữ liệu được truyền từ tầng dưới lên và qua mỗi tầng thì phần header tương ứng sẽ được lấy đi và khi đến tầng trên cùng thì dữ liệu không còn phần header nữa.



Hình 7. Cấu trúc dữ liệu trong TCP/IP

Hình trên cho ta thấy lược đồ dữ liệu qua các tầng.. Trong hình ta thấy tại các tầng khác nhau dữ liệu được mang những thuật ngữ khác nhau

- Trong tầng ứng dụng: dữ liệu là các luồng được gọi là stream.
- Trong tầng giao vận: đơn vị dữ liệu mà TCP gửi xuống gọi là TCP segment.
- Trong tầng mạng, dữ liệu mà IP gửi xuống tầng dưới gọi là IP Datagram
- Trong tầng liên kết, dữ liệu được truyền đi gọi là frame.

## 1.2. Lập trình Socket.

### 1.2.1. Khái niệm Socket và cổng port

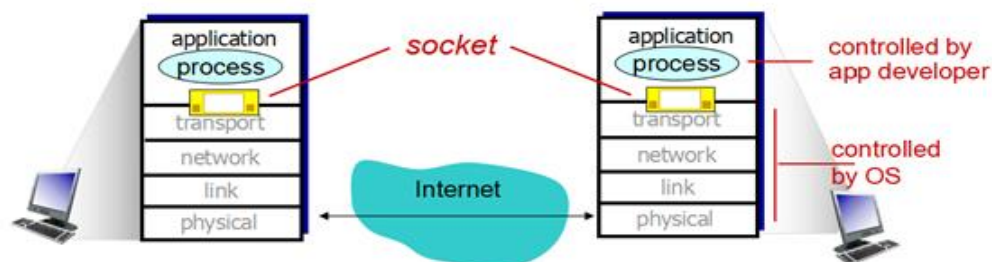
#### 1.2.1.1. Socket

**Socket** là một điểm cuối trong một kết nối giữa hai chương trình đang chạy trên mạng.

Trên quan điểm của người phát triển ứng dụng, **Socket** là một phương pháp để thiết lập kết nối truyền thông giữa một chương trình yêu cầu dịch vụ (được gắn nhãn Client) và một chương trình cung cấp dịch vụ (được gắn nhãn là Server) trên mạng hoặc trên cùng một máy tính.

Đối với người lập trình, họ nhìn nhận **Socket** như một giao diện nằm giữa tầng ứng dụng và tầng khác trong mô hình mạng OSI có nhiệm vụ thực hiện việc giao tiếp giữa chương trình ứng dụng với các tầng bên dưới của mạng.





Hình 8. Mô hình mạng OSI

### 1.2.1.2. Số hiệu cổng của Socket

Để có thể thực hiện các cuộc giao tiếp, một trong hai quá trình phải công bố số hiệu cổng socket mà mình sử dụng. Mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong hệ thống. Khi quá trình được gán một số hiệu cổng, nó có thể nhận dữ liệu gửi đến cổng này từ các quá trình khác. Quá trình còn lại cũng được yêu cầu tạo ra một socket.

Ngoài số hiệu cổng, hai bên giao tiếp còn phải biết địa chỉ IP của nhau. Địa chỉ IP giúp phân biệt máy tính này với máy tính khác trên mạng TCP/IP. Trong khi số hiệu cổng dùng để phân biệt các quá trình khác nhau trên cùng một máy tính.

Số hiệu cổng gán cho socket phải duy nhất trên phạm vi máy tính đó, có giá trị trong khoảng từ 0 đến 65535 (16 bit). Trong thực tế thì các số hiệu cổng từ 0 đến 1023 (gồm có 1024 cổng) đã dành cho các dịch vụ nổi tiếng như: http: 80, telnet: 21, ftp: 23, ... Nếu chúng ta không phải là người quản trị thì nên dùng từ cổng 1024 trở lên.

### 1.2.2. Nguyên lý hoạt động

Dưới góc độ lập trình các socket thường làm việc theo cặp, một socket đóng vai trò làm server còn các socket khác đóng vai trò như clients. Socket phía server xác định một cổng cho giao tiếp mạng, sau đó chờ nghe yêu cầu mà client gửi tới nó bằng client socket. Do đó các cổng cho server socket phải được biết bởi các chương trình client.

Như vậy số hiệu cổng của socket phía server được xác định bởi chương trình, ngược lại cổng cho client socket được xác định bởi hệ điều hành. Khi một socket phía client gửi một gói tin tới socket phía server thì trong gói tin đã có chứa thông tin về địa chỉ của hệ thống client và cổng của socket phía client nên server hoàn toàn có thể gửi thông tin phản hồi cho client.

Khái quát quá trình trao đổi dữ liệu thông qua các socket như sau:

+ Chương trình phía server tạo ra một socket, socket này được chương trình gắn với một cổng trên server. Sau khi được tạo ra, socket này sẽ chờ lắng nghe yêu cầu từ phía clients.

+ Khi chương trình phía clients cần kết nối với một server, nó cũng tạo ra một socket, socket này cũng được hệ điều hành gắn với một cổng. Chương trình client sẽ cung cấp cho socket của nó địa chỉ mạng và cổng của socket phía server và yêu cầu thực hiện kết nối.

+ Chương trình phía server và chương trình phía clients trao đổi dữ liệu với nhau bằng cách đọc từ socket hoặc ghi vào socket của mình.

### 1.2.3. Socket trong C#

#### 1.2.3.1. Lớp System.net.Socket

Lớp socket hỗ trợ các phương thức cần thiết để xây dựng các chương trình client sử dụng socket ở chế độ có kết nối. Dưới đây là một số phương thức thường dùng để xây dựng client.

***public Socket(AddressFamily, SocketType, ProtocolType) throws IOException***

Phương thức này dùng để tạo một thể hiện của lớp Socket có địa chỉ chỉ định AddressFamily, loại Socket SocketType và giao thức. .

***public IPEndPoint(IPAddress, int32)***

Tạo một đối tượng mới của lớp IPEndPoint, tham số truyền vào là địa chỉ IPAddress và số hiệu cổng dùng để giao tiếp.

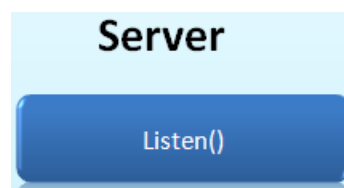
***public Close()***

Phương thức này sẽ đóng socket lại, giải phóng kênh ảo, xóa nối kết giữa clients và server.

### 1.3. Mô hình Client/Server

Mô hình client/server sử dụng socket ở chế độ hướng kết nối TCP

Giai đoạn 1: Server tạo socket, gán số hiệu cổng và lắng nghe yêu cầu kết nối.



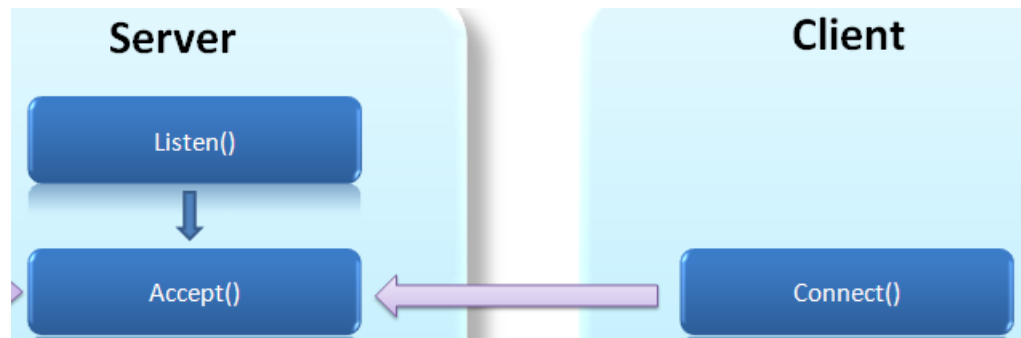
Hình 9. Mô hình client/server giai đoạn 1

*socket()*: Server yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.

*bind()*: Server yêu cầu gán số hiệu cổng (port) cho socket.

*listen()*: Server lắng nghe các yêu cầu nối kết từ các client trên cổng đã được gán.

Giai đoạn 2: Client tạo Socket, yêu cầu thiết lập một nối kết với Server.



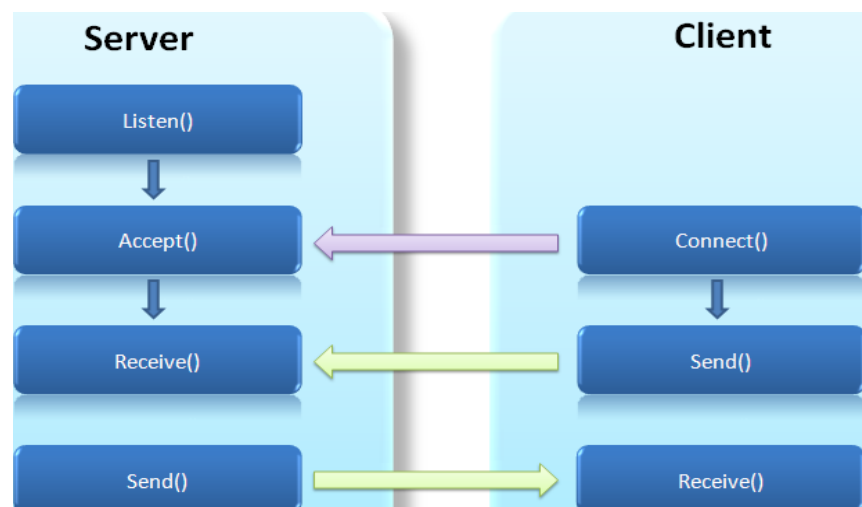
Hình 10. Mô hình client/server giai đoạn 2

*socket()*: Client yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn rảnh cho socket của Client.

*Connect()*: Client gửi yêu cầu kết nối đến Server có địa chỉ IP và Port xác định.

*Accept()*: Server chấp nhận kết nối của Client, khi đó một kênh giao tiếp ảo được hình thành, Client và Server có thể trao đổi thông tin với nhau thông qua kênh ảo này.

Giai đoạn 3: Trao đổi thông tin giữa Client và Server



Hình 11. Mô hình client/server giai đoạn 3

Sau khi chấp nhận yêu cầu kết nối, thông thường Server thực hiện lệnh *receive()* và nghe cho đến khi có thông điệp yêu cầu (Request Message) từ Client gửi đến.

Server phân tích và thực thi yêu cầu. Kết quả sẽ được gửi về client bằng lệnh *send()*.

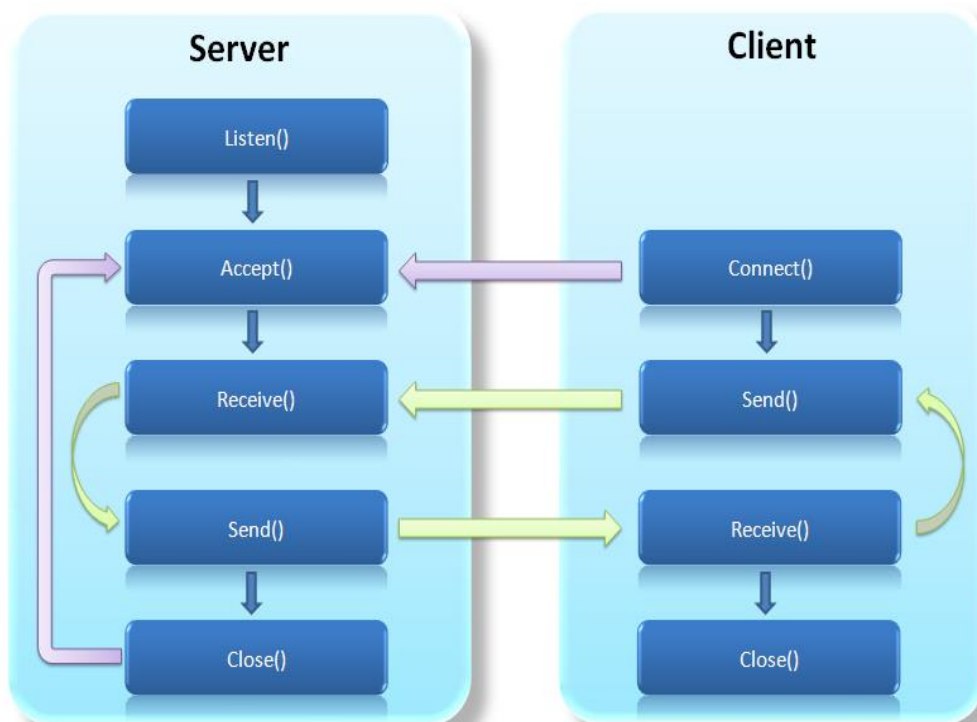
Sau khi gửi yêu cầu bằng lệnh *send()*, client chờ nhận thông điệp kết quả (ReplyMessage) từ Server bằng lệnh *receive()*.

Giai đoạn 4: Kết thúc phiên làm việc.

Các câu lệnh *receive()*, *send()* có thể được thực hiện nhiều lần (ký hiệu bằng hình ellipse).

Kênh ảo sẽ bị xóa khi Server hoặc Client đóng socket bằng lệnh *close()*.

Như vậy toàn bộ quá trình sẽ diễn ra như hình bên dưới:



Hình 12. Mô hình client/server giai đoạn 4

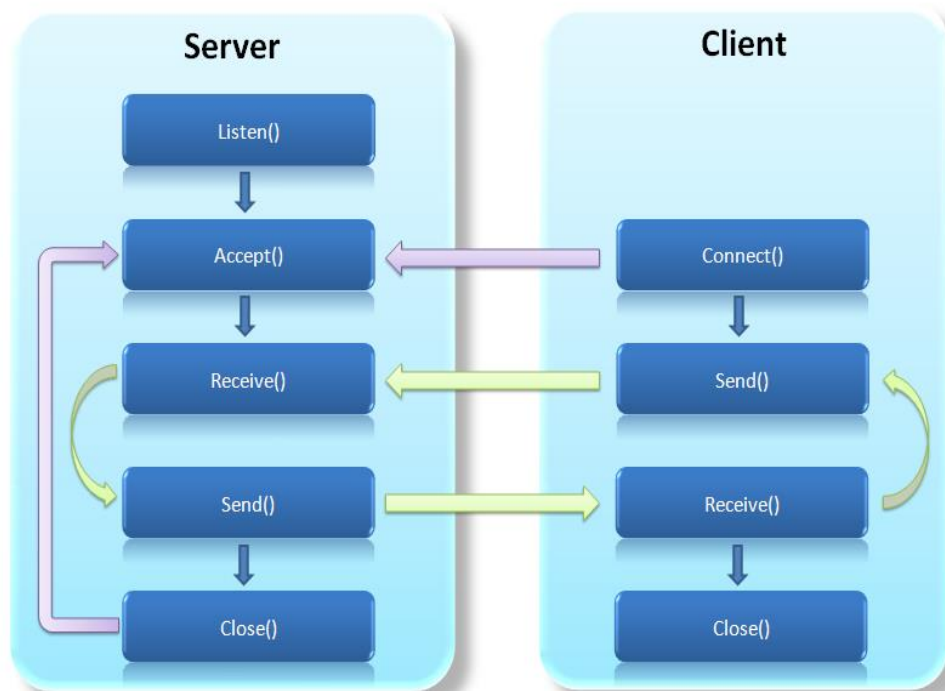
## CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Yêu cầu bài toán

Bài toán: **Xây dựng ứng dụng hỗ trợ quản lý công việc theo lịch cho công ty.**

Bài toán yêu cầu xây dựng chương trình hỗ trợ quản lý công việc giữa các máy dựa vào giao thức TCP trên môi trường LAN. Xây dựng chương trình theo mô hình Client/Server với giao diện dễ dàng sử dụng.

### 2.2. Xây dựng chương trình.



Hình 13. Mô hình Client-Server

#### 2.2.1. Client

-Tạo clientSocket:

```

Socket clientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.IP);
IPEndPoint IP = new
IPEndPoint(IPAddress.Parse("192.168.43.206"), 6740);

```

client.Connect(IP);

-Các hoạt động của client đều sẽ gửi lên Server:

```

Client.Send(Serialize(string.Format("User,{0},{1},{2}", username,
Job.Job, Job.Status)));

```

Trong đó, mỗi lần thay đổi trạng thái công việc, người dùng đều gửi 1 thông điệp lên server là một chuỗi từ khoá gồm mã người dùng, công việc, trạng thái công việc.

-Nhận thông báo về công việc liên quan tới chính mình từ Server:

```
byte[] data = new byte[1024];  
client.Receive(data);  
  
string message = Deserialize(data) as string;
```

Lọc các gói tin từ server phù hợp với phân quyền người dùng và hiển thị thông báo ra giao diện để người dùng nhận biết dễ dàng.

```
Notify.ShowBalloonTip(Cons.notifyTimeOut, "Lịch công việc", message,  
ToolTipIcon.Info);
```

### 2.2.2 Server

-Khởi tạo Server

```
EndPoint IP = new EndPoint(IPAddress.Any, 6740);  
Socket server = new Socket(AddressFamily.InterNetwork,  
SocketType.Stream, ProtocolType.IP);  
  
server.Bind(IP);
```

-Sử dụng Thread để tiến hành nhận các block từ Client

```
Thread receive = new Thread(Receive);  
receive.IsBackground = true;  
receive.Start(client);
```

-Gửi phản hồi cho Client

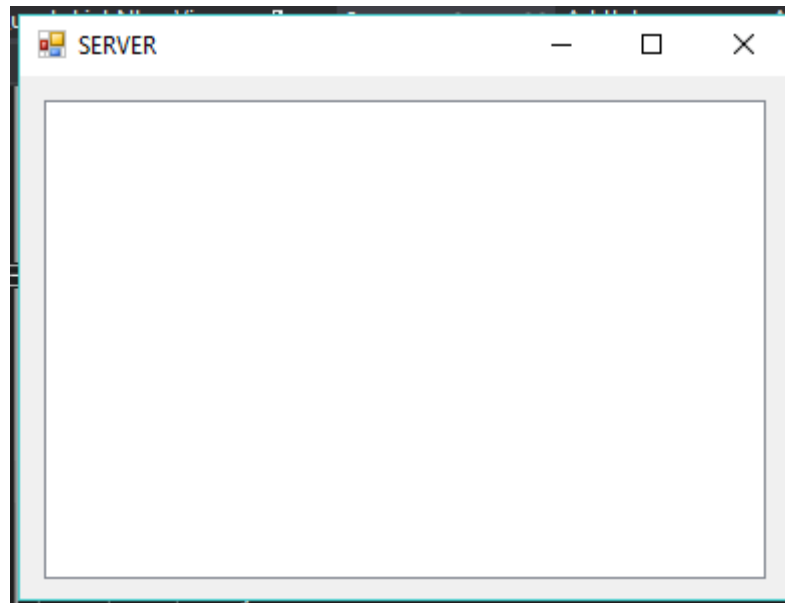
```
client.Send(Serialize(message));
```

## CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

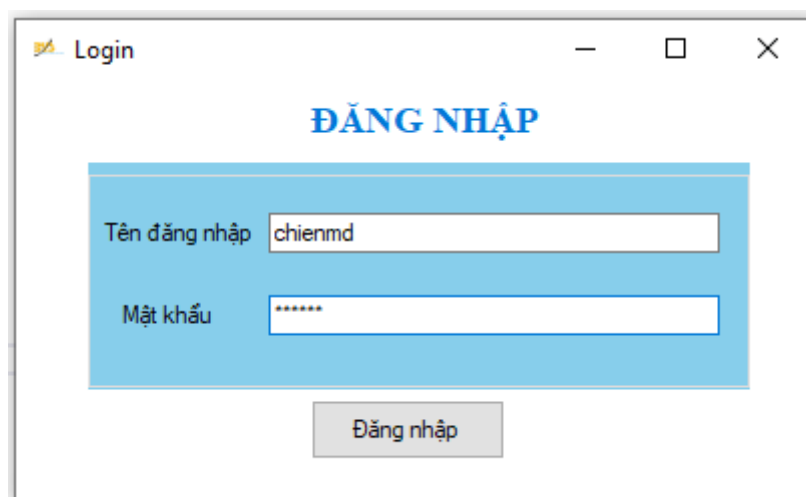
### 3.1. Môi trường triển khai và chọn ngôn ngữ

- Trình biên dịch: Visual Studio 2017
- Ngôn ngữ: C#

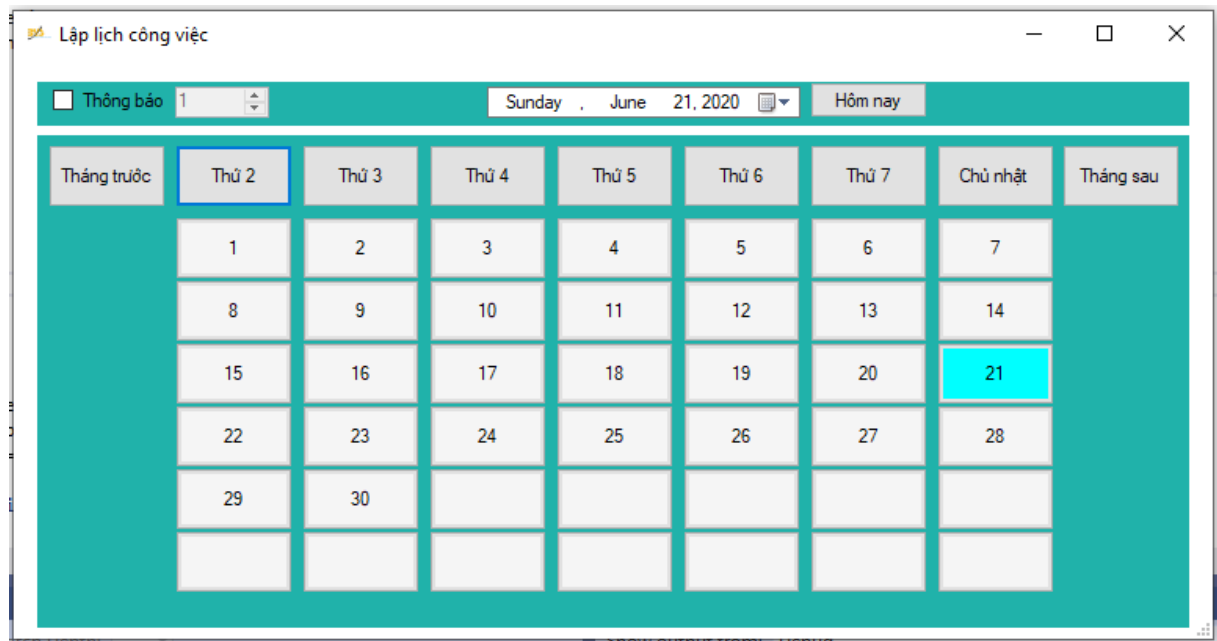
### 3.2. Kết quả thực thi chương trình



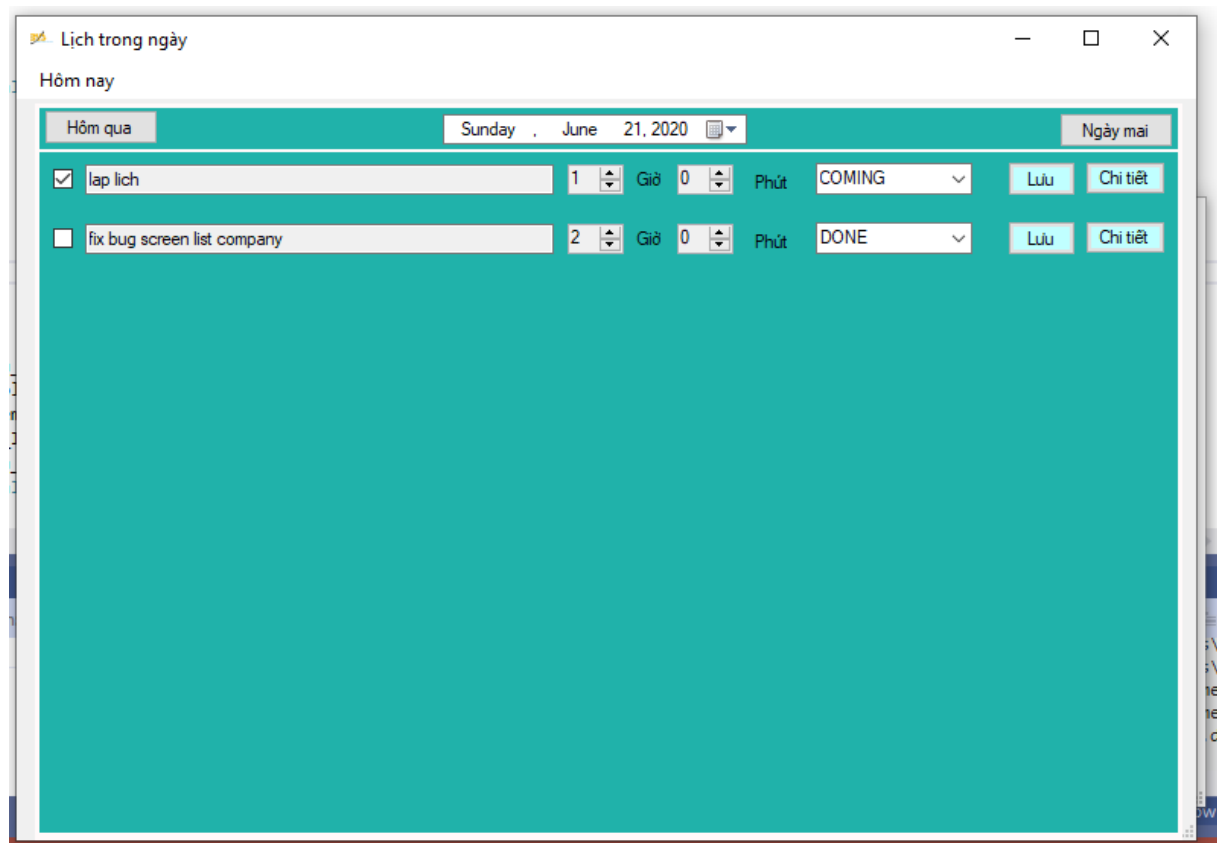
Hình 14. Giao diện Server



Hình 15. Giao diện login

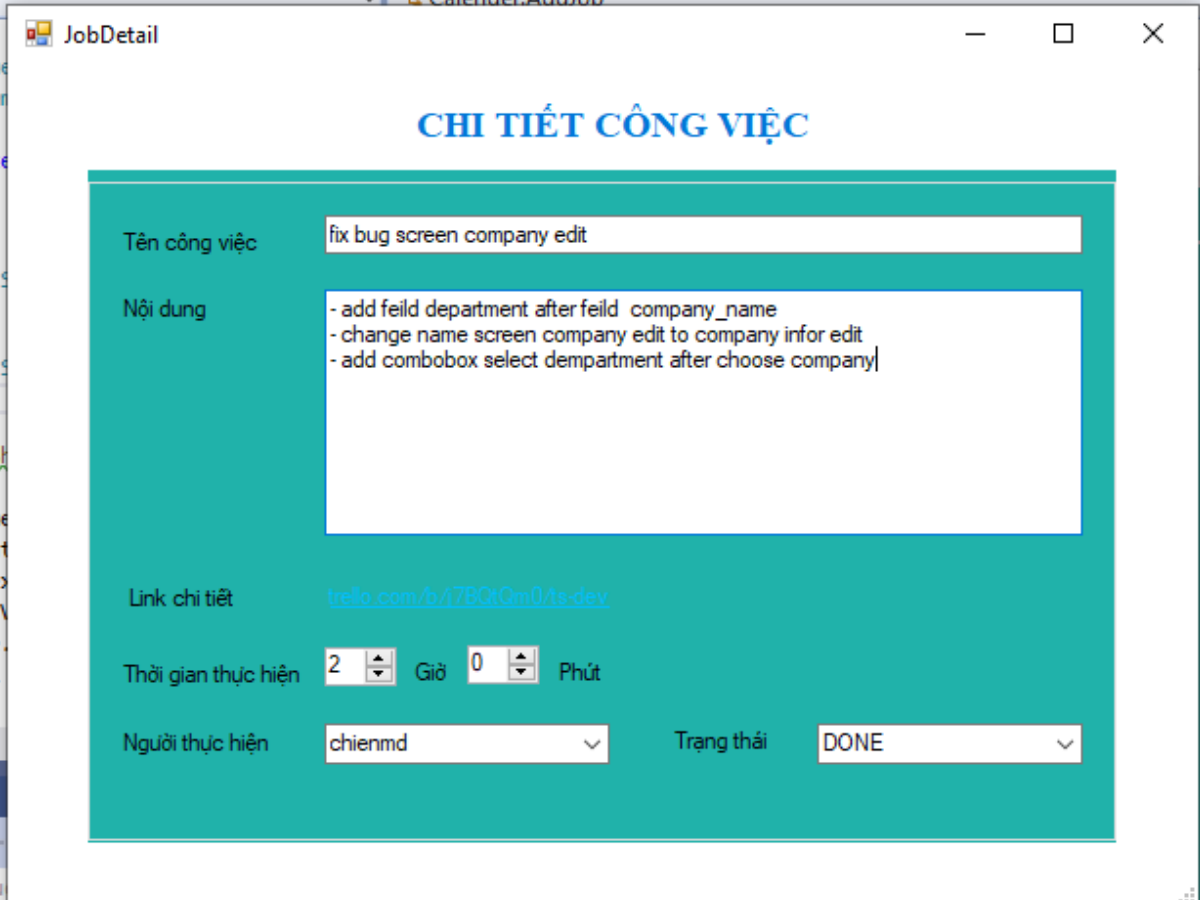


Hình 16. Giao diện Client



Hình 17. Xem công việc trong ngày





The screenshot shows a web application window titled "JobDetail". The main heading is "CHI TIẾT CÔNG VIỆC" (Job Details). The form contains the following fields:

- Tên công việc** (Job Name): A text input field containing "fix bug screen company edit".
- Nội dung** (Content): A text area containing a list of tasks:
  - add feild department after feild company\_name
  - change name screen company edit to company infor edit
  - add combobox select dempartment after choose company
- Link chi tiết** (Detailed Link): A text input field containing the URL "trelo.com/b/4783tQm0As-dev".
- Thời gian thực hiện** (Execution Time): Two spinners for hours and minutes. The hours spinner is set to "2" and the minutes spinner is set to "0".
- Người thực hiện** (Person performing): A dropdown menu with "chienmd" selected.
- Trạng thái** (Status): A dropdown menu with "DONE" selected.

Hình 18. Chi tiết công việc

**AddJob**

### THÊM CÔNG VIỆC

Tên công việc:

Nội dung:

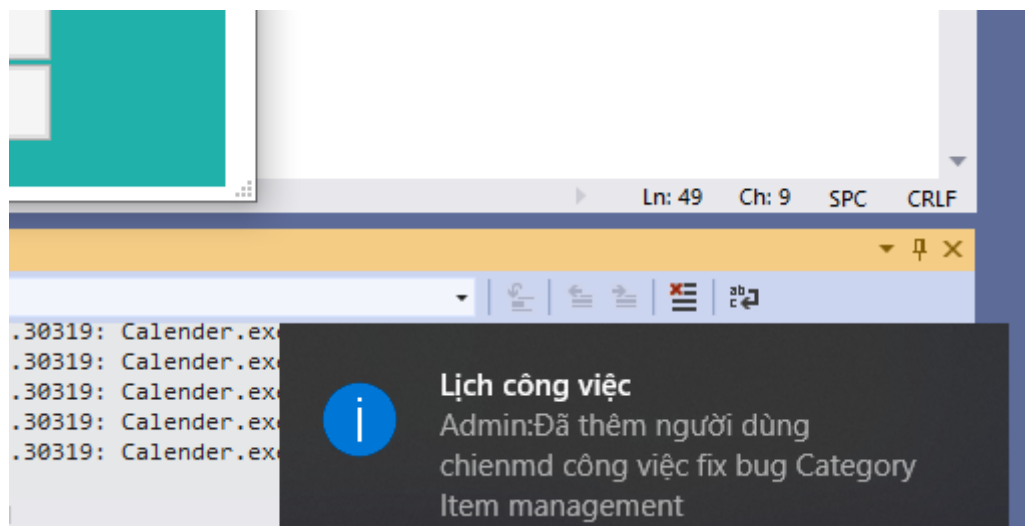
Link chi tiết:

Thời gian thực hiện:  Giờ  Phút

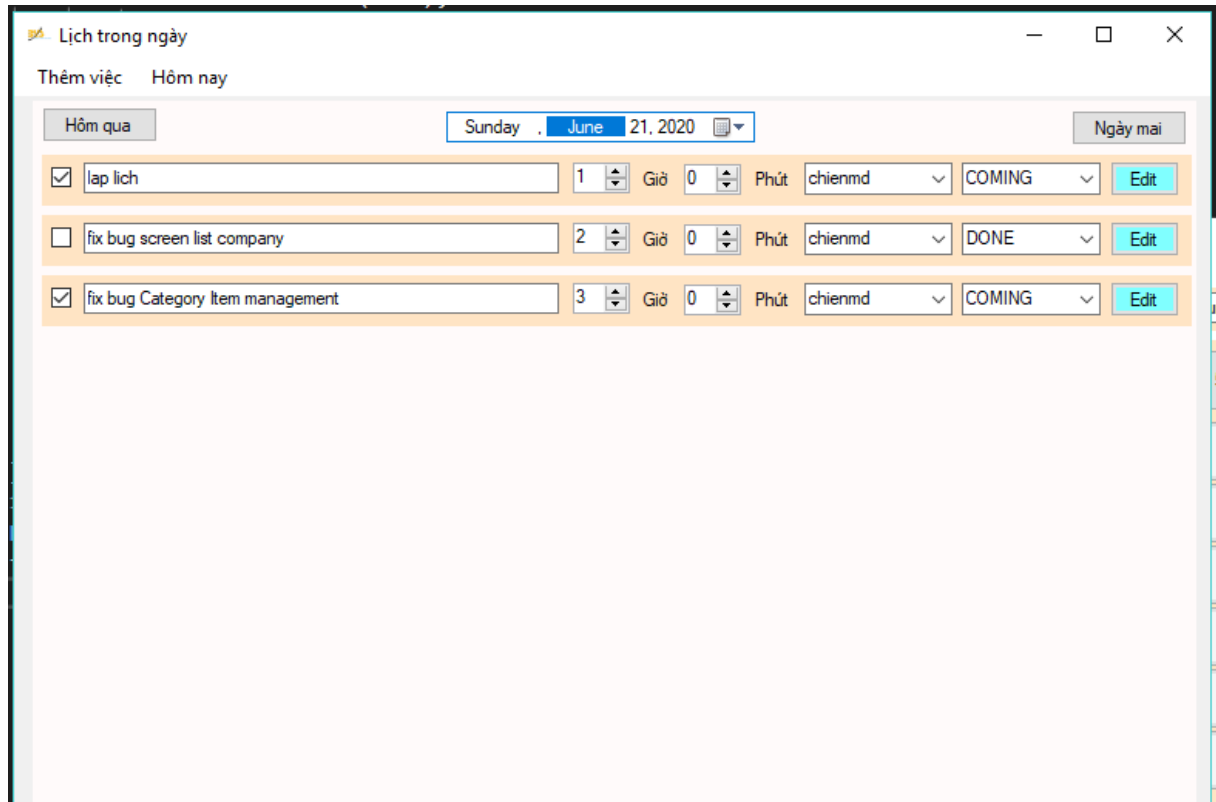
Người thực hiện:

Trạng thái:

Hình 19. Phân công công việc cho người dùng



Hình 20. Server thông báo cho người dùng về công việc được nhận



Hình 21. Màn hình lịch của client là leader

### 3.3. Ưu nhược điểm

- Ưu điểm:
  - Người dùng sẽ được phần mềm hỗ trợ trong việc kiểm soát các công việc được giao như : thông báo đều đặn về công việc được giao trong ngày, thông báo về công việc vừa được giao. Người quản lý sẽ nhận được thông báo công việc của nhân viên.
  - Người quản lý có thể tạo công việc mới => Thêm các nhân viên sẽ thực hiện công việc đó.
- Nhược điểm:
  - Lỗi hiển thị khi thêm nhiều nhân viên vào một công việc.

## CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 4.1. Kết luận

- Lý thuyết: Hiểu được mô hình Client – Server. Cách truyền nhận thông điệp bằng giao thức TCP.

- Thực tiễn: Xây dựng được ứng dụng hỗ trợ công việc theo lịch cho nhiều người dùng trong mạng LAN có phân quyền(Admin và User).
- Người dùng sẽ được phần mềm hỗ trợ trong việc kiểm soát các công việc được giao như : thông báo điều đặn về công việc được giao trong ngày, thông báo về công việc vừa được giao. Người quản lý sẽ nhận được thông báo công việc của nhân viên.
- Người quản lý có thể tạo công việc mới => Thêm các nhân viên sẽ thực hiện công việc đó.

Tuy nhiên, vẫn còn một số hạn chế như lỗi hiển thị khi thêm nhiều nhân viên vào một công việc, giao diện phân việc của quản lý chưa được rõ ràng.

#### **4.2. Hướng phát triển**

- Phát triển phần mềm thêm các tính năng giúp người dùng dễ sử dụng hơn như xem danh sách nhân viên đã được phân công trong một công việc cụ thể, xoá công việc.
- Xây dựng giao diện ứng dụng thân thiện hơn.

### PHẦN III. KẾT LUẬN CHUNG

Sau quá trình làm đồ án cơ sở ngành mạng với hai phần hệ điều hành và lập trình mạng, giúp em biết được nhiều điều:

- Biết áp dụng lý thuyết vào xây dựng bài toán thực tế.
- Về hệ điều hành:
  - Hiểu được rõ hơn về các tiến trình cũng như cách đồng bộ tiến trình để các tiến trình không bị xung đột trong khi chạy.
  - Biết cách bắt các sự kiện trong windows để thực hiện một số hành động.
  - Hiểu được quy trình hoạt động của hook bàn phím.
  - Hiểu được về ngôn ngữ C#, cách sử dụng smtp, registry.
- Về lập trình mạng:
  - Giúp em nắm rõ hơn về mô hình Client-Server từ đó xây dựng chương trình hỗ trợ quản lý công việc trên môi trường LAN có ứng dụng rất thiết thực vào thực tế.
  - Hiểu rõ cách truyền thông điệp giữa client-server.
  - Hiểu thêm về lập trình trên ngôn ngữ C#: Winform, kết nối Sql Server
  - Biết cách trình bày một giao diện phù hợp với chương trình.
- Biết cách trình bày báo cáo, slide, phục vụ cho việc học tập cũng như làm việc sau này.

## TÀI LIỆU THAM KHẢO

### Hệ điều hành

- [1] Nguyễn Hải Châu, *Nguyên lý hệ điều hành*, đại học Công Nghệ
- [2] Nguyễn Phú Trường, *Giáo trình hệ điều hành*, đại học Cần Thơ
- [3] Trần Hạnh Nhi, *Giáo trình hệ điều hành nâng cao*, đại học Khoa học tự nhiên Hồ Chí Minh
- [4] <https://stackoverflow.com>

### Lập trình mạng

- [1] [https://msdn.microsoft.com/en-us/library/system.net.sockets\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.sockets(v=vs.110).aspx)
- [2] Richard Blum, *C# Network Programming*
- [3] <https://stackoverflow.com>