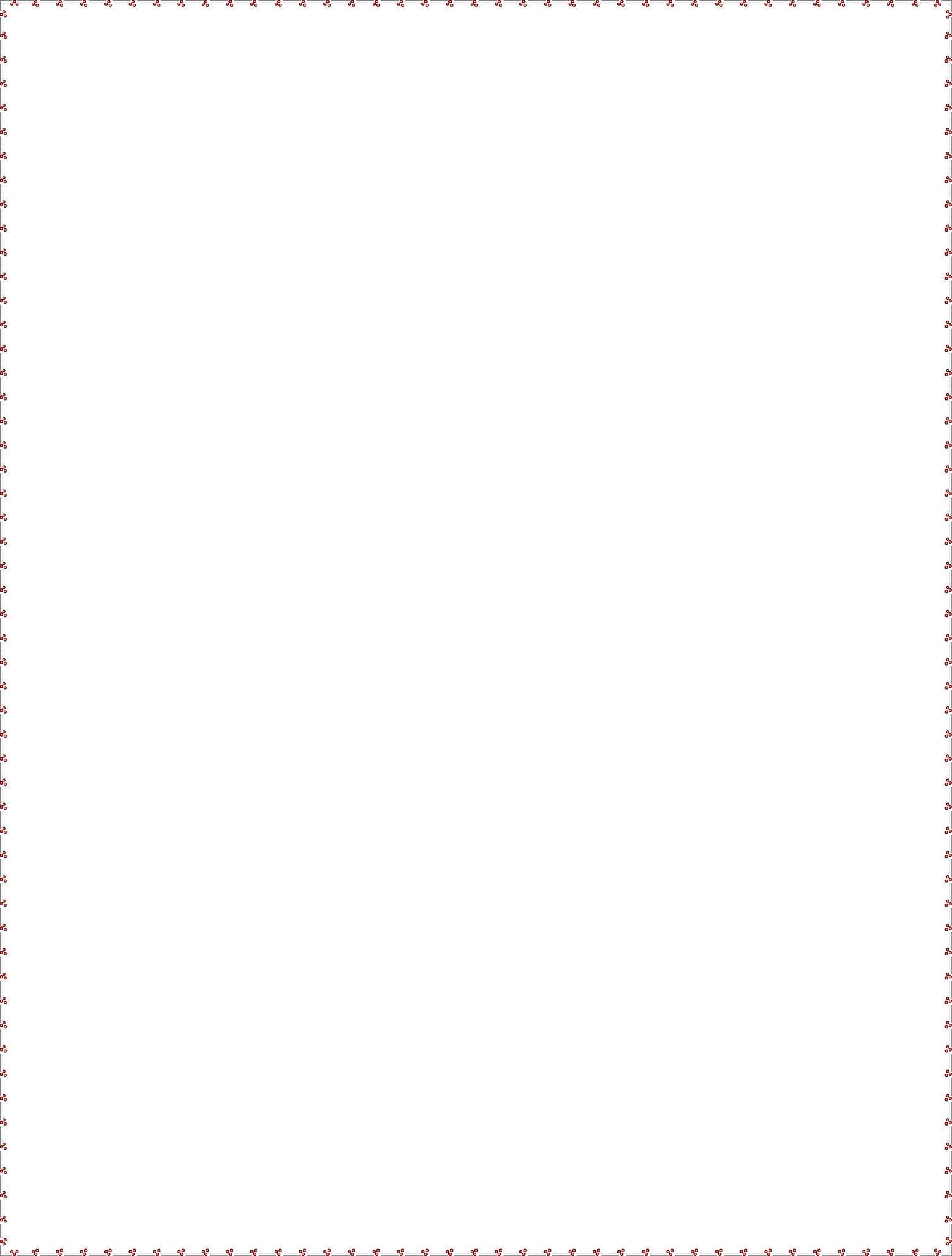
TRƯỜNG ĐẠI HỌC THỦY LỢI



**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỆ ĐIỀU HÀNH**

**ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH ỨNG DỤNG NGÔN NGỮ HỖ TRỢ HỆ ĐIỀU HÀNH WINDOWS**

*Giảng viên hướng dẫn :* TS. Phạm Thanh Bình

*Nhóm sinh viên thực hiện :*

1. Vũ Xuân Huy- 61TH
2. Nguyễn Ngọc Lâm- 61TH
3. Nguyễn Văn Duy - 61TH
4. Lê Khắc Minh Đức - 61TH
5. Bùi Đức Liêm - 61TH

**Hà Nội, 5/2022**

**MỤC LỤC**

[MỞ ĐẦU 3](#_bookmark0)

1. [Giới thiệu: 3](#_bookmark1)
2. [Công cụ sử dụng: 3](#_bookmark2)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH VÀ HÀM API TRÊN HỆ ĐIỀU](#_bookmark3) [HÀNH WINDOWS 4](#_bookmark3)

* 1. [Tổng quan về hệ điều hành windows 4](#_bookmark4)
  2. [Tổng quan về tiến trình. 4](#_bookmark5)
     1. [Tiến trình là gì? 4](#_bookmark6)
     2. [Chương trình ví dụ về tiến trình 4](#_bookmark7)

[1.2 Hàm API trên hệ điều hành windows. 6](#_bookmark8)

* + 1. [Hàm API là gì? 6](#_bookmark9)

[CHƯƠNG 2: CHƯƠNG TRÌNH ỨNG DỤNG HỖ TRỢ HỆ ĐIỀU HÀNH VÀ](#_bookmark11)

[THỰC NGHIỆM 8](#_bookmark11)

* 1. [Giới thiệu bài toán chương trình ứng dụng. 8](#_bookmark12)
  2. [Giải thuật của bài toán. 8](#_bookmark13)
  3. [Mã nguồn chương trình ứng dụng và thực nghiệm 9](#_bookmark14)
     1. [Mã nguỗn chương trình 9](#_bookmark15)
     2. [Hình ảnh màn hình thực nghiệm chương trình 16](#_bookmark16)

[CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 18](#_bookmark17)

* 1. [Kết luận. 18](#_bookmark18)
  2. [Hướng phát triển. 18](#_bookmark19)

[TÀI LIỆU THAM KHẢO 19](#_bookmark20)

1. **Giới thiệu:**

# MỞ ĐẦU

Hệ điều hành là một hệ chương trình hoạt động giữa người dùng và phần cứng của máy tính. Từ đó, cung cấp một môi trường để người sử dụng có thể thi hành các chương trình làm cho máy tính dễ sử dụng hơn, thuận tiện và hiệu quả. Mỗi phiên bản của hệ điều hành đều hỗ trợ mặc định bộ gõ dạng english. Nên để có thể giúp người dùng sử dụng được keyboard của hệ điều hành với unikey tiếng Việt, nhóm đã quyết định lựa chọn đề tài **“Xây dựng ứng dụng unikey hỗ trợ hệ điều hành windown”.**

# Công cụ sử dụng:

Ngôn ngữ lập trình: NodeJS.

Framework: Express;

Phần mềm code: Visual studio code.

# Nội dung bố cục của báo cáo được trình bày trong 3 chương:

**Chương 1:** Tổng quan về hệ điều hành và hàm API trên hệ điều hành windown.

Chương này trình bày các khái niệm tổng quan logic về hệ điều hành windows. Từ đó, giới thiệu về tiến trình trong hệ điều hành widnows, một số hàm API và công dụng của hàm API trên windows.

**Chương 2:** Chương trình ứng dụng ngôn ngữ hỗ trợ hệ điều hành và thực nghiệm.

Chương này, trình bày bài toán, giải thuật ứng dụng unikey. Từ đó đưa ra đầy đủ mã nguồn và chú thích cho từng module chức năng và hình ảnh thực nghiệm của ứng dụng.

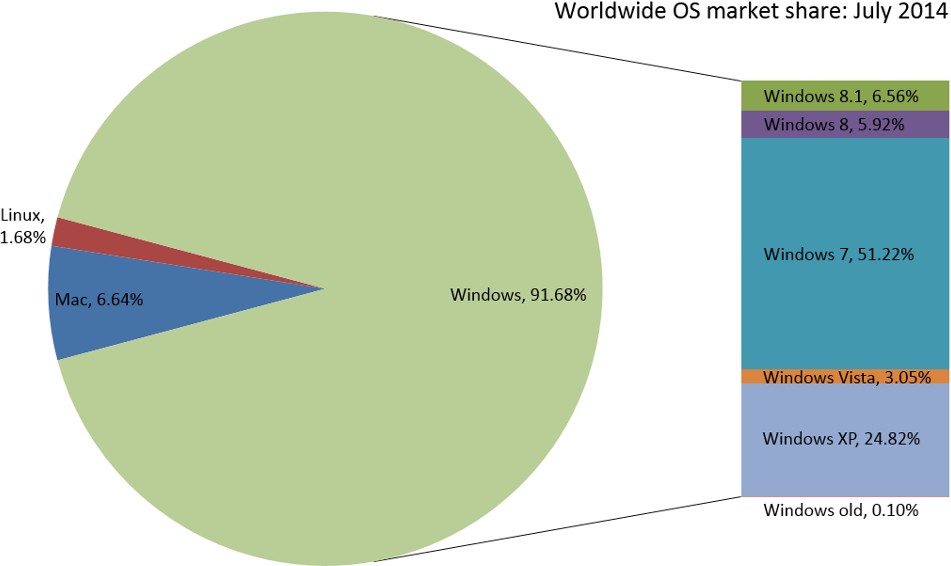
**Chương 3:** Kết luận và hướng phát triển.

Chương này đưa ra kết luận về những gì đã làm được và rút ra bài học định hướng phát triển ứng dụng cho ứng dụng hỗ trợ hệ điều hành trong tương lai.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH VÀ HÀM API TRÊN HỆ ĐIỀU HÀNH WINDOWS

# Tổng quan về hệ điều hành windows.

Định nghĩa: Hệ điều hành là một chương trình hay một hệ chương tình hoạt động giao tiếp giữa người dùng và phần cứng máy tính. Cung cấp một môi trường để người sử dụng có thể thi hành chương trình, giúp ích cho người dùng trong công việc và cuộc sống. Hệ điều hành bao gồm các chương trình dịch, hệ thống cơ sở sữ liệu,... sử dụng tài nguyên của máy tính để giải quyết các yêu cầu người dùng.



# Hình 1: Hình ảnh minh họa sự so sánh giữa các loại hệ điều hành hiện nay

Hệ điều hành windows: Một hệ điều hành đa nhiệm có thể xử lý nhiều chương trình cùng một lúc. Hệ điều hành được phân phối bởi Microsoft.

# Tổng quan về tiến trình.

# Tiến trình là gì?

Tiến trình là một thực thể đang thực hiện điều khiển một đoạn mã lệnh riêng không gian, địa chỉ, ngăn xếp và sở hữu một trạng thái giúp thông báo nó đang làm gì (đang chạy, đang chờ, đã đóng, ...).

Tiến trình có 4 thành phần quan trọng: CPU, bộ nhớ, File, Thiết bị nhập xuất.

# Chương trình ví dụ về tiến trình

**-** Mã nguồn code:

+ Khai báo các hàm, yêu cầu các thư viện được sử dụng để chạy server

const express = require('express');

const server = express();

const basicLocalOrigin = 'http://localhost:';

const port = 8080;

server.get('/', (req, res) => res.send('This is home page'));

server.get('/product', (req, res) => res.send('This is product page'));

server.listen(port, () => console.log("Server running: " + basicLocalOrigin + port))

+ Khai báo hàm hiện thị danh sách các tiến trình install

const Service = require('node-windows').Service

const svc = new Service({

name: 'serverOne',

description: 'This is test',

script: "C:\\Users\\TzonG\\Desktop\\NODE\\index.js"

})

svc.on('install', function() {

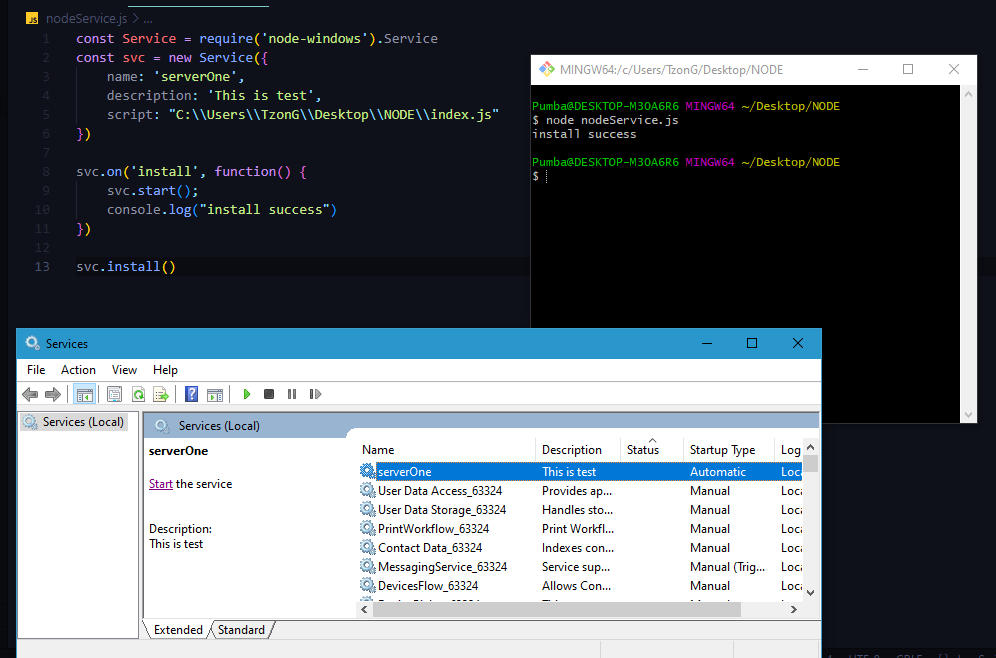
svc.start();

console.log("install success")

})

svc.install()

- Hình ảnh chạy chương trình install:



+ Khai báo hàm hiện thị danh sách các tiến trình uninstall

const Service = require('node-windows').Service

const svc = new Service({

name: 'serverOne',

description: 'This is test',

script: "C:\\Users\\TzonG\\Desktop\\NODE\\index.js"

})

svc.on('uninstall', function() {

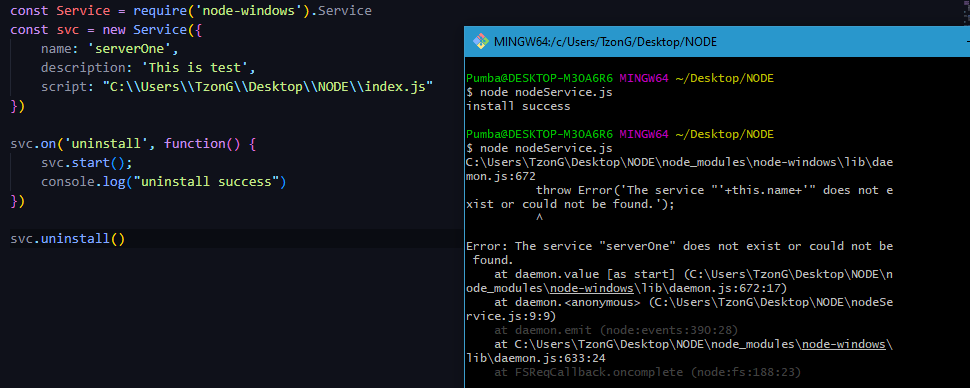
svc.start();

console.log("uninstall success")

})

svc.uninstall()

- Hình ảnh chạy chương trình uninstall:



# Hàm API trên hệ điều hành windows.

# Hàm API là gì?

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dung) phương thức kết nối với các thư viện và ứng dụng khác. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau giữa chương trình và hệ điều hành.

# Một số hàm API trên hệ điều hành widnows

**- Các API của windows :**

1. **KERNEL32**:

Là DLL chính, đảm nhiệm quản lý bộ nhớ, thưc hiện chức năng đa nhiệm và những hàm ảnh hưởng trực tiếp đến hoạt động của Windows.

# USER32:

Thư viện quản lý Windows. Thư viện này chứa các hàm xử lý menu, định giờ, truyền tin, tập tin và nhiều phần không được hiển thị khác của Windows.

# GDI32:

Giao diện thiết bị đồ hoạ (Graphics Device Interface). Thư viện này cung cấp các hàm vẽ trên màn hình, cũng như kiểm tra phần biểu mẫu nào cần vẽ lại.

# WINNM:

Cung cấp các hàm multimedia để xử lý âm thanh, nhạc, video thời gian thực, lấy mẫu, v.v… Nó là DLL 32 bit. (Thư viện 16 bit tên là*MMSYSTEM*)

1. **Node-windows**

Muốn một tập lệnh Node chạy 24/7 trên nền máy tính của bạn mà không cần nhớ khởi động nó? Bạn không muốn có bất kỳ dấu hiệu rõ ràng nào cho thấy nó đang chạy? Chà, bạn có thể thiết lập dự án của mình để chạy dưới dạng Dịch vụ Windows ở chế độ nền, với khả năng khởi động tự động, cũng như các đặc quyền khác để dự án chạy dưới dạng Dịch vụ Windows.

# CHƯƠNG 2: CHƯƠNG TRÌNH ỨNG DỤNG HỖ TRỢ HỆ ĐIỀU HÀNH VÀ THỰC NGHIỆM

# Giới thiệu bài toán chương trình ứng dụng.

* Chương trình gõ Tiếng Việt trên Hệ điều hành windows cho phép người dùng máy tính sử dụng bàn phím đưa vào các sự kiện keyboard thành các ký tự Tiếng việt hiện thị trên các thiết bị và chương trình ghi.
* Nguyên tắc chung của bộ gõ là sử dụng Hock bàn phím chặn các thông điệp về bàn phím như: trạng thái, bàn phím, các thông điệp.
* Dùng các giải thuật riêng của mình để xử lý chuỗi đệm đã thu được thành chuỗi Tiếng Việt và xuất ra các thiết bị ghi.

Từ đó, chương trình sẽ sử dụng hỗ trợ hàm API trên hệ điều hành windows tiến trình của chương trình sẽ chạy ngầm định cho phép người dùng gõ unicode Tiếng việt trên các trình soạn thảo ghi.

# Giải thuật của bài toán.

**Đầu vào:** Sự kiện nhấn người dùng nhấn keyboard. Hệ điều hành sẽ lưu nhật ký của keyboard.

**Đầu ra:** Cho phép người dùng thực hiện chức năng Unicode trên thiết bị ghi. Các bước tiến hành:

Bước 1: Bắt sự kiện người dùng nhấn keyboard. Khai báo khởi tạo trạng thái của chương trình.

Bước 2: Thêm các ký tự từ bàn phím vào để tìm từ cuối cùng trong chuỗi để xử lý.

Bước 3: Khai báo sử dụng chức năng Hock (là cơ một cơ chế mà một ứng dụng có thể chặn các sự kiện như bàn phím, chuột,..) và các giá trị hỗ trợ hock.

Bước 4: Xử lý hàm lấy trạng thái từ sự kiện nhấn bàn phím của user. Sử dụng hỗ trợ của hàm API.

Bước 5: Xử lý hàm trả về thông tin xử lý của hệ điều hành. Bước 6: Xử hàm gọi tới tiến trình console của hệ điều điều hành.

Bước 7: Xử lý hàm xử lý thao tác dấu câu (kiển Telex) và lấy ra chuỗi nguyên â, để xác định vị trí đặt dấu.

Bước 8: Xử lý hàm xóa trong cửa sổ Focus căn cứ vào chuỗi bộ đệm, sau đó gọi tới hàm thủ tục lập bàn phím back để xóa các ký tự ban đầu trong cửa sổ Focus.

Bước 9: Xử lý hiện thị mà hình console và hướng dẫn người dùng sử dụng chương trình ứng dụng.

# Mã nguồn chương trình ứng dụng và thực nghiệm.

# Mã nguỗn chương trình

+ Khai báo thư viện cho phép gọi tới hàm API của windows.

const Service = require('node-windows').Service

+ Khởi tạo đối tượng trong service của window thông qua thư viện node-windows

const svc = new Service({

name: 'serverOne',

description: 'This is test',

script: "C:\\Users\\TzonG\\Desktop\\NODE\\index.js"

})

+ Chạy chương trình

svc.on('install', function() {

svc.start();

console.log("install success")

})

svc.install()

# Hình ảnh màn hình thực nghiệm chương trình

# 

**Hình 3: Hình ảnh On chức năng chương trình ứng dụng**

Hình ảnh gỡ chương trình đã được tải. Ta có thể thấy name: serverOne không tìm thấy bên trong service của window.

# CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# Kết luận.

Khi chúng ta đang làm việc trên Rest API trong node js hoặc bất kỳ loại ứng dụng node js nào, chúng ta cần chạy ứng dụng của mình trong node cmd với npm run start hoặc npm run dev commond và chúng ta cần dừng và khởi động nó theo cách thủ công mỗi lần bằng các lệnh.

Nhưng điều gì sẽ xảy ra nếu chúng ta có thể chạy nó dưới dạng dịch vụ windows để nó tiếp tục chạy trong dịch vụ nền của windows và chúng ta không cần phải duy trì hoạt động của cmd để chạy nút js của chúng ta.

Sau đó, chạy ứng dụng node js như dịch vụ windows trong hình. Chỉ cần làm theo một số phương pháp và bản đồ đường đi, bạn có thể chạy ứng dụng node js của mình làm dịch vụ nền và làm cho cuộc sống của bạn dễ dàng hơn.

Sau thời gian nghiên cứu học hỏi, nhóm em đã xây dựng thành công chương trình ứng dụng hỗ trợ ngôn ngữ cho hệ điều hành windows. Với chức năng on, off chạy ngầm trên tiến trình của hệ điều hành. Tuy nhiên chương trình không tránh đươc nhiều sai sót, nhóm m rất mong nhận được góp ý từ thầy. Nhóm em chân thành cảm ơn thầy nhiều.

# Hướng phát triển.

Chúng em sẽ cố gắng hoàn thiện hơn trong tương lai. Đồng thời, áp dụng kiến thức môn học để phát triển nhiều ứng dụng hơn nữa để hỗ trợ cho hệ điều hành windows.

# TÀI LIỆU THAM KHẢO

* + 1. [(4) Node.js: Install .js project as a Service | Windows Tutorial - YouTube](https://www.youtube.com/watch?v=1smy9yyme1Y)
    2. [(4) How to Run NodeJS Application as a Windows Service or background Service Easy Method Bug Shop - YouTube](https://www.youtube.com/watch?v=obXr1NtSEOU&t=14s)