**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------\*\*\*----------**



**BÀI TẬP LỚN**

**Môn: Nguyên lý hệ điều hành**

|  |  |
| --- | --- |
| ***Đề tài:*** | **SỬ DỤNG HÀM API TRONG C# ĐỂ TÌM KIẾM TIẾN TRÌNH NOTEPAD** |

|  |  |
| --- | --- |
| *Họ và tên:* | 1. Đặng Anh Tuấn 2. Phan Lan Nhi 3. Nguyễn Thị Kiều Trang 4. Vũ Văn Quý 5. Đàm Khắc Thái |
| *Lớp:* | 63HTTT1 |
| *Giảng viên hướng dẫn:* | ThS Vũ Thành Vinh |

***Hà Nội., 2023***

**Mục lục**

[**I, Tổng quan về bài toán** 1](#_Toc148300586)

[**II, Một số hàm API của Windows dùng với C#** 2](#_Toc148300587)

[**III, Cách thức khai báo hàm API trong C#** 3](#_Toc148300588)

[1. Import thư viện 3](#_Toc148300589)

[2. Khai báo và gọi hàm API 3](#_Toc148300590)

[**IV, Các hàm API đã gọi** 4](#_Toc148300591)

[1. Hàm `FindWindow` 4](#_Toc148300592)

[2. Hàm `GetWindowThreadProcessId` 4](#_Toc148300593)

[3. Hàm `SendMessage` 5](#_Toc148300594)

[**V, Mã nguồn của chương trình** 5](#_Toc148300595)

[**VI, Kết quả chương trình** 6](#_Toc148300596)

# **I, Tổng quan về bài toán**

Đoạn mã sử dụng ngôn ngữ C# để tìm kiếm tiến trình Notepad và một số luồng liên quan. Nó sẽ tìm kiếm cửa sổ Notepad bằng tên lớp và tiêu đề đã xác định trước đó. Nếu không tìm thấy, nó sẽ khởi động Notepad và chờ một giây để Notepad khởi động hoàn toàn trước khi tìm kiếm lại cửa sổ Notepad.

Sau đó, nó lấy ID của tiến trình Notepad và in ra thông tin về tiến trình, bao gồm ID, tên, trạng thái, mức độ ưu tiên và 3 bộ nhớ chính của tiến trình. Ở đây chương trình đang sử dụng phương thức Responding của đối tượng Process để kiểm tra xem tiến trình Notepad có đang phản hồi hay không. Phương thức Responding trả về giá trị bool: true nếu tiến trình đang phản hồi (responsive) và false nếu không phản hồi (not responsive). Điều này giúp chương trình kiểm tra xem liệu tiến trình Notepad đã đóng hoàn toàn hay chưa sau khi gửi tin nhắn đóng. Nếu tiến trình vẫn còn tồn tại, chương trình sẽ tiếp tục chờ cho đến khi tiến trình Notepad đóng hoàn toàn trước khi in thông báo tới người dùng. Tiếp theo, nó in ra thông tin về luồng liên quan trong tiến trình Notepad.

Cuối cùng, nó yêu cầu người dùng nhấn Enter để đóng tiến trình Notepad. Nếu người dùng nhấn Enter, nó sẽ gửi một tin nhắn tới cửa sổ Notepad để đóng nó và đợi cho đến khi tiến trình Notepad đã đóng hoàn toàn. Cuối cùng, nó in ra thông báo để cho biết rằng tiến trình Notepad đã đóng.

# **II, Một số hàm API của Windows dùng với C#**

**1. FindWindow: Tìm kiếm cửa số chương trình trong windows**  
[DllImport("user32.dll")]  
public static extern IntPtr FindWindow(string lpClassName, string lpWindowName);  
  
**2. FindWindowEx: Tìm kiếm cửa số con khi đã biết được Handle của cửa số cha.**  
[DllImport("user32.dll")]  
public static extern IntPtr FindWindowEx(IntPtr hWnd1, IntPtr hWnd2, string lpsz1, string lpsz2);  
  
**3. SetForegroundWindow: Active một cửa sổ chương trình.**  
[DllImport("user32.dll")]  
public static extern bool SetForegroundWindow(IntPtr hWnd);  
  
**4. SendMessageW: gửi thông điệp đến cửa số ứng dụng (thông điệp unicode)**  
[DllImport("user32.dll")]  
public static extern IntPtr SendMessageW(IntPtr hWnd, uint Msg, IntPtr wParam, IntPtr lParam);  
  
**5. GetWindowText: lấy caption của cửa số ứng dụng**  
[DllImport("user32.dll")]  
public static extern int GetWindowText(IntPtr hWnd, StringBuilder lpString, int nMaxCount);  
  
**6. Các hàm về menu**  
**6.1 GetMenu: lấy menu của cửa sổ ứng dụng.**  
[DllImport("user32.dll")]  
public static extern IntPtr GetMenu(IntPtr hwnd);  
  
**6.2 GetSubMenu: lấy submenu khi có giá trị của Menu**  
[DllImport("user32.dll")]  
public static extern IntPtr GetSubMenu(IntPtr hMenu, int nPos);  
  
**6.3 GetMenuItemID: lấy Id của menu trong Submenu**  
[DllImport("user32.dll")]  
public static extern IntPtr GetMenuItemID(IntPtr hMenu, int nPos);

# **III, Cách thức khai báo hàm API trong C#**

## 1. Import thư viện

using System.Runtime.InteropServices;

## 2. Khai báo và gọi hàm API

Cú pháp:[DllImport(DLL\_FileName)][public|private] static extern Return\_Type Function\_Name([type parameter,... ]);

Trong đó:

* "user32.dll" là tên của thư viện DLL chứa hàm API. Đây là thư viện Windows API chứa các hàm liên quan đến giao diện người dùng.
* SetLastError = true: Là một tham số của thuộc tính DllImport trong việc khai báo một API từ một DLL bên ngoài. Nếu có lỗi, cần kiểm tra mã lỗi trả về sau khi gọi API để xác định nếu có lỗi xảy ra và xử lý nó một cách thích hợp.
* `ReturnType` là kiểu dữ liệu trả về của hàm API.
* `FunctionName` là tên của hàm API.
* `ParameterType1`, `ParameterType2`, ... là kiểu dữ liệu của các tham số đầu vào của hàm API.
* `parameter1`, `parameter2`, ... là tên các tham số đầu vào của hàm API.

Ví dụ gọi hàm API tìm kiếm cửa sổ dựa trên tên lớp và trên sửa sổ:

[DllImport("user32.dll", SetLastError = true)]

static extern IntPtr FindWindow(string lpClassName, string lpWindowName);

# **IV, Các hàm API đã gọi**

Tất cả các hàm API trong đoạn mã đều được gọi từ thư viện user32.dll, một thư viện quan trọng trong Windows API có chức năng liên quan đến giao diện người dùng. Dưới đây là giải thích chi tiết về từng hàm API đã gọi trong chương trình:

## 1. Hàm `FindWindow`

* Mô tả: Tìm kiếm cửa sổ dựa trên tên lớp và tên cửa sổ.
* Tham số:

+ `lpClassName`: Tên lớp của cửa sổ cần tìm kiếm.

+ `lpWindowName`: Tên của cửa sổ cần tìm kiếm.

* Trả về:

+ `IntPtr`: Con trỏ đến cửa sổ đã tìm thấy. Trả về `IntPtr.Zero` nếu không tìm thấy cửa sổ.

## 2. Hàm `GetWindowThreadProcessId`

* Mô tả: Lấy ID của tiến trình liên quan đến một cửa sổ.
* Tham số:

+ `hWnd`: Con trỏ đến cửa sổ cần lấy ID tiến trình.

+ `processId`: Tham chiếu tới biến để lưu ID tiến trình.

* Trả về:

+ `uint`: ID của tiến trình.

## 3. Hàm `SendMessage`

* Mô tả: Gửi một thông điệp tùy chỉnh đến một cửa sổ.
* Tham số:

+ `hWnd`: Con trỏ đến cửa sổ mà thông điệp sẽ được gửi đến.

+ `Msg`: Mã thông điệp.

+ `wParam`: Tham số wParam của thông điệp.

+ `lParam`: Tham số lParam của thông điệp.

* Trả về:

+ `bool`: Trả về `true` nếu thành công, `false` nếu không thành công.

# **V, Mã nguồn của chương trình**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

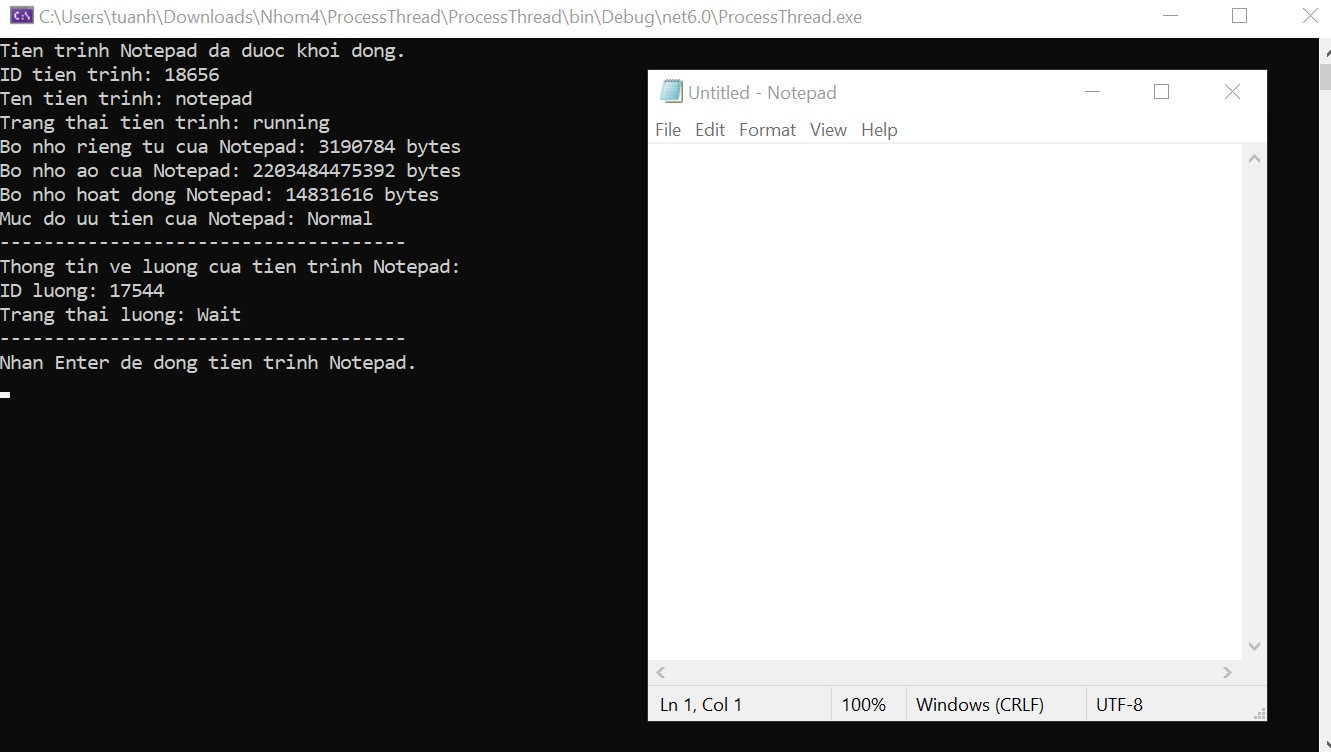
Description automatically generated

A screenshot of a computer program

Description automatically generated

Source code: <https://github.com/DTuan03/API.git>

# **VI, Kết quả chương trình**



A screenshot of a computer

Description automatically generated