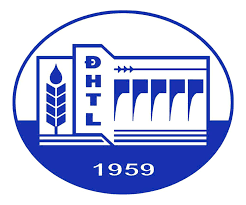
**TRƯỜNG ĐẠI HỌC THỦY LỢI**

****

**Môn Học**

**Hệ Điều Hành**

**Đề Tài**

**Phần Mềm Hẹn Giờ Tắt Máy Tính**

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

**Sinh viên thực hiện : Trần Chí Tuấn - 61TH - 1951064111**

**Mạc Long Vũ - 61TH - 1951064114**

*Hà Nội, ngày 7/5/2022*

# 

# **MỤC LỤC**

**MỞ ĐẦU**

1. **Giới thiệu 3**
2. **Công cụ sử dụng 3**

**CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH VÀ HÀM API TRÊN HỆ ĐIỀU HÀNH WINDOWS 3**

1. **Tổng quan về hệ điều hành windows**
2. **Tổng quan về tiến trình**
3. **Hàm API trên hệ điều hành windows 4**

**CHƯƠNG 2: CHƯƠNG TRÌNH ỨNG DỤNG HỖ TRỢ HỆ ĐIỀU HÀNH VÀ THỰC NGHIỆM 4**

1. **Giới thiệu chương trình ứng dụng**
2. **Giải thuật bài toán**
3. **Mã nguồn chương trình và thực nghiệm**

**MỞ ĐẦU**

1. **Giới thiệu**

Hệ điều hành (*Operating System - OS*) là [phần mềm hệ thống](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_h%E1%BB%87_th%E1%BB%91ng) quản lý [phần cứng máy tính](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_c%E1%BB%A9ng), [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) và cung cấp các [dịch vụ](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_n%E1%BB%81n_(daemon)) chung cho [các chương trình máy tính](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh).

Đối với các chức năng phần Cứng như [đầu vào, đầu ra](https://vi.wikipedia.org/wiki/%C4%90%E1%BA%A7u_v%C3%A0o/%C4%91%E1%BA%A7u_ra) và cấp phát [bộ nhớ](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B), hệ điều hành đóng vai trò quan trọng giữa các chương trình và phần cứng máy tính, mặc dù mã ứng dụng thường được thực thi trực tiếp bởi phần cứng và thường thực hiện [các lệnh hệ thống](https://vi.wikipedia.org/wiki/L%E1%BB%9Di_g%E1%BB%8Di_h%E1%BB%87_th%E1%BB%91ng) đến Chức năng hệ điều hành hoặc bị hệ điều hành [làm gián đoạn](https://vi.wikipedia.org/wiki/Ng%E1%BA%AFt). Hệ điều hành cung cấp một môi trường cho phép người sử dụng phát triển và thực hiện các [ứng dụng](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_%E1%BB%A9ng_d%E1%BB%A5ng) của họ một cách dễ dàng.

Với mục đích hỗ trợ người dùng trong việc sử dụng hệ điều hành, chúng em đã phát triển ứng dụng hẹn giờ tắt máy, sử dụng hàm API trên hệ điều hành windows.

1. **Công cụ sử dụng**

Ngôn ngữ lập trình: C#

Phần mềm code: Visual studio

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

1. **Tổng quan về hệ điều hành windows**

Microsoft Windows (hoặc đơn giản là Windows) là tên của một hệ điều hành dựa trên giao diện người dùng đồ hoạ được phát triển và được phân phối bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft). Nó bao gồm một vài các dòng hệ điều hành, mỗi trong số đó phục vụ một phần nhất định của ngành công nghiệp [máy tính](https://www.dienmayxanh.com/laptop).

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

1. **Tổng quan về tiến trình**

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.

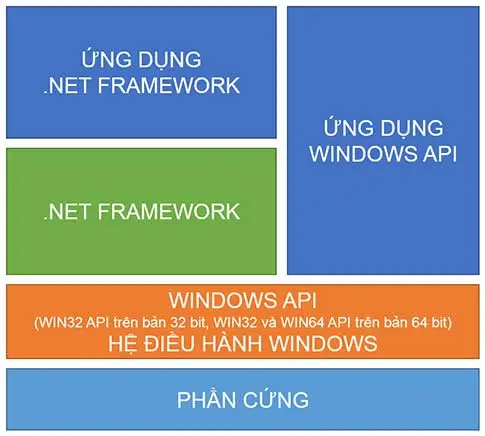
1. **Hàm API trên hệ điều hành windows**

**3.1 Hàm API là gì?**

API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau.

Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

3.2 Hàm API trên hệ điều hành windows

Windows API (WinAPI, hay nhiều nơi vẫn dùng tên cũ là Win32 API) là các hàm thư viện và các định nghĩa khác (struct, enum,…) được Windows cung cấp cho người lập trình, để viết các ứng dụng trên nền Windows.

Mô hình windows API

### **THÀNH PHẦN CỦA WINDOWS API**

Các hàm API có thể được chia thành một số loại chính:

* Base Services: thao tác với các tài nguyên trên Windows như hệ thống tập tin (file systems), các thiết bị, các tiến trình (processes), các luồng (threads), quản lý bộ nhớ,… Các hàm này nằm trong thư viện kernel32.dll.
* Advanced Services: truy xuất Windows registry, tắt, mở máy tính, Windows service, tài khoản người dùng,… Các hàm này nằm trong thư viện dvapi32.dll.
* Graphics Device Interface: xuất dữ liệu đồ họa ra màn hình, máy in,… Các hàm này nằm trong thư viện gdi32.dll.
* User Interface: cung cấp các thành phần giao diện người dùng như form, nút, textbox, nhận tín hiệu chuột, bàn phím,… Các hàm này nằm trong thư viện shell32.dll, user32.dll, comctl32.dll, comdlg32.dll:
  + Common Dialog Box Library: các hộp thoại cơ bản trong Windows mà hầu như ứng dụng nào cũng sử dụng, như hộp thoại mở file, lưu file,…
  + Common Control Library: nút, status bars, progress bars, toolbars, tabs,…
  + Windows Shell: thu tín hiệu chuột, bàn phím,…
* Network Services: các thao tác liên quan đến mạng
* Multimedia: thao tác với các thiết bị đa phương tiện

**Một số hàm API:**

1. Declare Function AnyPopup Lib "user32" Alias "AnyPopup" () As Long
2. Công dụng: Đưa ra chỉ số cửa sổ popup hiện đang tồn tại trên màn hình.
3. Trị trả về: Integer ~ True (Khác zero) nếu có cửa sổ popup.
4. Declare Function AdjustWindowRectEx Lib "user32" Alias "AdjustWindowRectEx"
5. (lpRect As RECT, ByVal dsStyle As Long, ByVal bMenu As Long, ByVal dwEsStyle As
6. Long) As Long
7. Công dụng: Điều chỉnh cửa sổ khi có vùng làm việc client (Không tính kích thước của
8. thanh tiêu đề, đường viền và các phần thêm) được khai báo, khi biết kiểu cửa sổ.
9. Tham số kèm: LpRect Hình chữ nhật chứa vùng làm việc client. DwStyle Kiểu cửa sổ.
10. BMenu Đưa giá trị True (Khác zero) nếu cửa sổ có trình đơn DwEsStyle kiểu cửa sổ mở rộng.

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

1. **Giới thiệu chương trình ứng dụng**

Chương trình Hẹn giờ Shutdown giúp người dùng kiểm soát thời gian sử dụng máy tính hoặc đơn giản là quá bận nhưng chưa thể tắt máy ngay.

Chức năng của chương trình là tắt máy khi chạy hết thời gian đã hẹn.

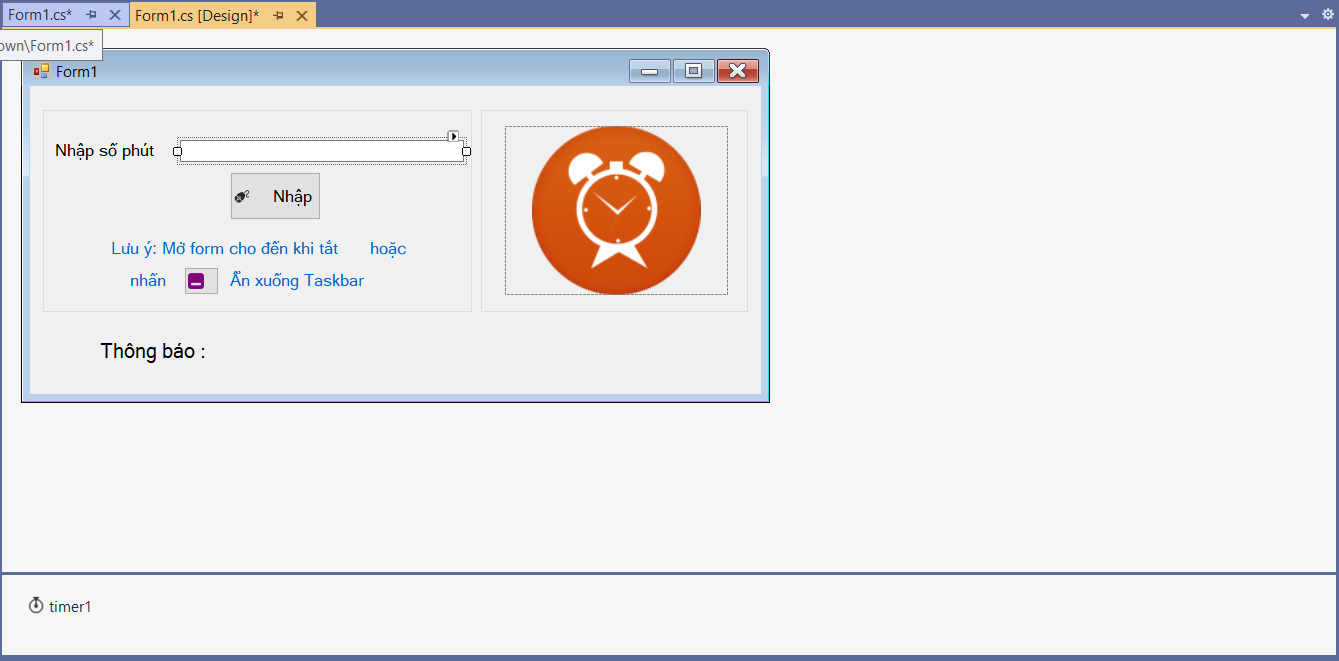
1. **Giải thuật bài toán**

**Đầu vào:** Người dùng nhập thời gian để tắt máy (tính bằng số phút)

Sự kiện: khi nhấn vào button, thời gian sẽ bắt đầu đếm ngược

**Đầu ra:** Tắt máy khi hết thời gian

Bước 1: Thiết kế form



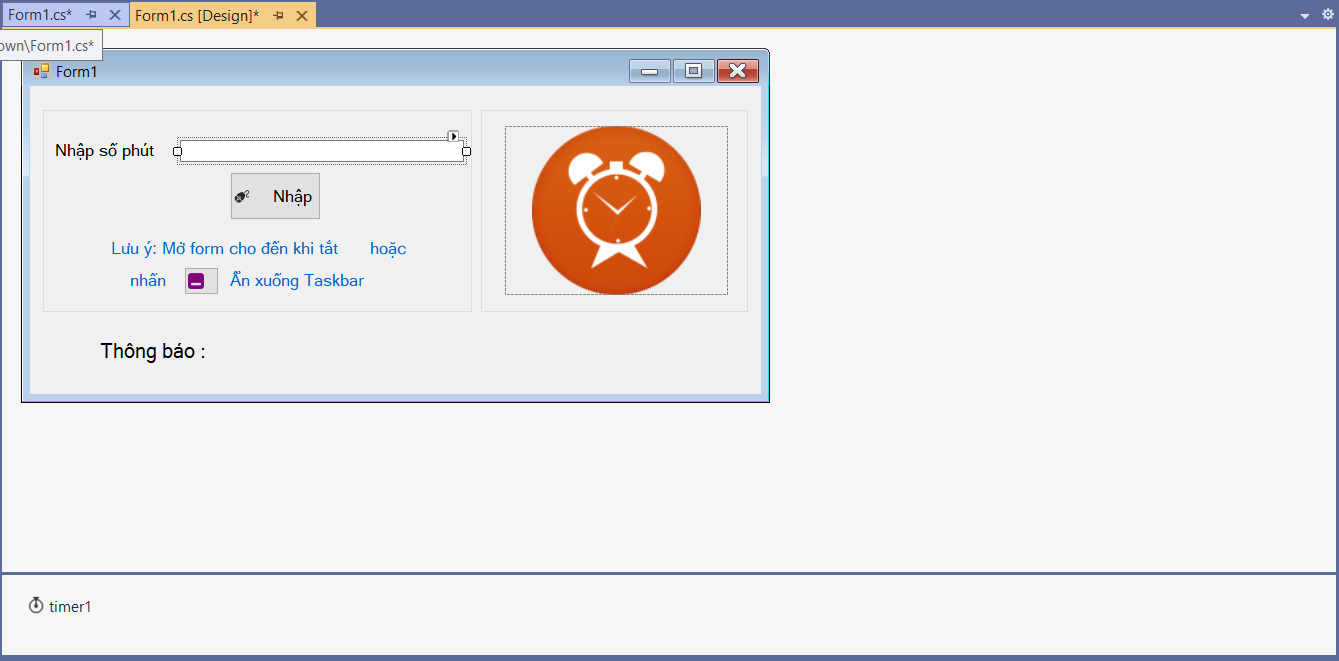
Bước 2: Xử lý sự kiện click nhập số phút

Bước 3: Xử lý hàm đếm ngược và tắt máy (gọi API)

Bước 4: Khai báo thư viện và gọi các hàm tới hàm main

1. **Mã nguồn chương trình và thực nghiệm**

- Thiết kế form:



* **Form1.cs: Hàm xử lý đầu vào**

public Form1()

{

InitializeComponent();

}

int soGiayTatMay;

private void button1\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox1.Text.Trim(), out soGiayTatMay))

{

button1.Enabled = false;

timer1.Interval = 1000;

timer1.Enabled = true;

soGiayTatMay = Math.Abs(soGiayTatMay) \* 60;

}

else

{

MessageBox.Show("Hãy nhập vào số phút cần hẹn.");

}

}

* **Hàm xử lý thời gian và tắt máy:**

private void timer1\_Tick(object sender, EventArgs e)

{

if (soGiayTatMay > 0)

{

label1.Text = "Sẽ tắt máy sau " + soGiayTatMay + " giây.";

soGiayTatMay = soGiayTatMay - 1;

}

else

{

//gọi api

System.Diagnostics.Process.Start("shutdown", "/s /f /t 0");

Application.Exit();

}

}

* **Hàm main**

[STAThread]

static void Main()

{

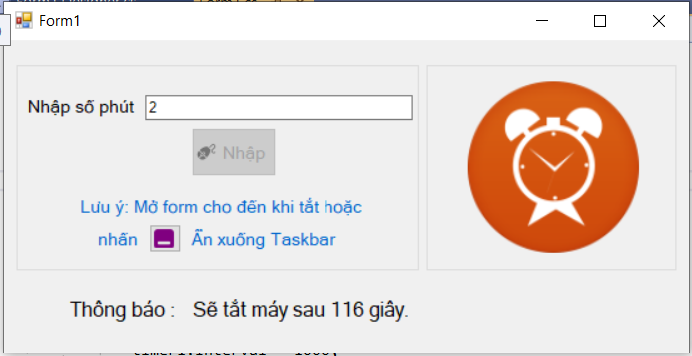
Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

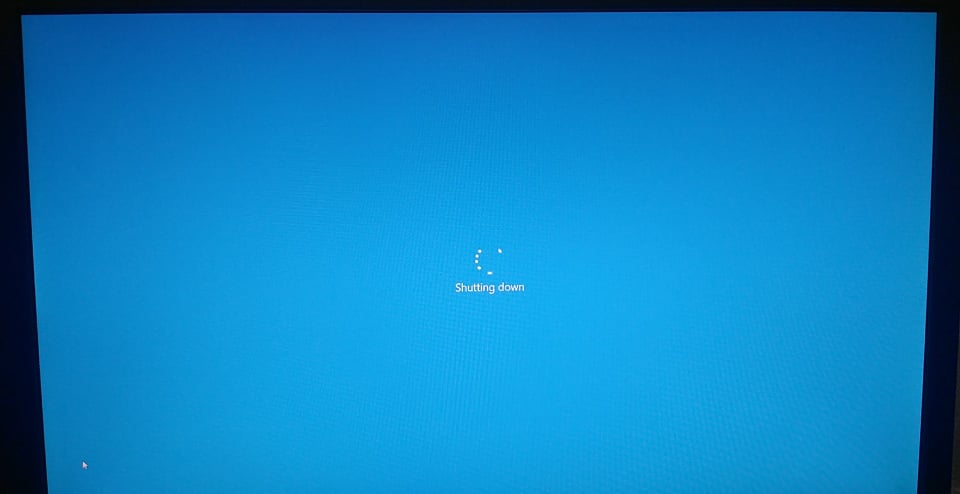
Application.Run(new Form1());

}

1. **Hình ảnh thực nghiệm chạy chương trình**



**Hình ảnh chương trình đếm ngược**



**Hình ảnh chương trình đã tắt được máy tính**