

MÔN: HỆ ĐIỀU HÀNH

Bài thực hành số 1.2: Viết phần mềm kích hoạt/dừng process

I. Mục tiêu

- Giúp SV làm quen với việc lập trình để kích hoạt/dừng process bất kỳ.


II. Nội dung

- Xây dựng chương trình nhỏ cho phép người dùng chọn file khả thi cần chạy, kích hoạt chạy nó và dừng/xóa nó khi cần thiết.

III. Chuẩn đầu ra

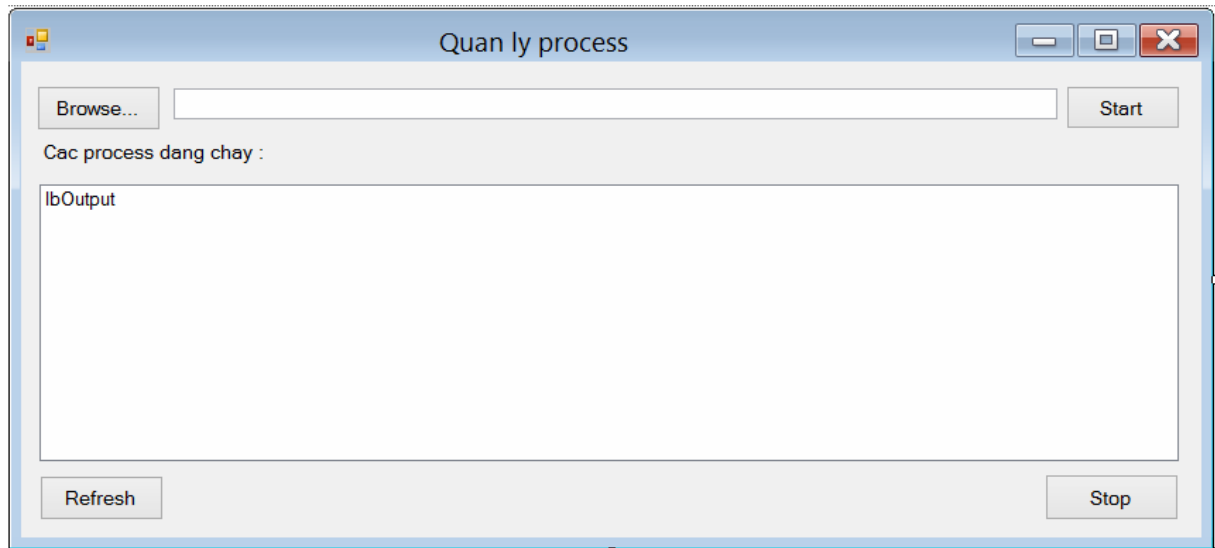
- Sinh viên nắm vững và sử dụng thành thạo class Process của .NET để quản lý process.
- Sinh viên hiểu rõ 1 trong 3 cách kích hoạt/dừng process được trình bày trong slide bài giảng : process đang chạy có thể kích hoạt tự động process khác chạy và khi cần có thể giết chết process khác.


IV. Qui trình

1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Windows, chọn icon "Windows Form Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. ProcessManager), click button OK để tạo Project theo các thông số đã khai báo.
3. Form đầu tiên của ứng dụng đã hiển thị trong cửa sổ thiết kế. Hãy chọn Form để hiển thị cửa sổ thuộc tính của nó (thường ở góc dưới phải màn hình), duyệt tìm thuộc tính Text trong danh sách rồi hiệu chỉnh nội dung của thuộc tính này thành "Quản lý process". Chuỗi văn bản này sẽ được hiển thị ngay trên thanh title bar của Form ứng dụng.
4. Việc thiết kế chi tiết trong form là quá trình lập 4 thao tác : tạo mới/xóa/hiệu chỉnh thuộc tính/tạo hàm xử lý sự kiện cho từng đối tượng giao diện cần dùng trong form.
5. Nếu cửa sổ ToolBox chưa hiển thị chi tiết, chọn menu View.Toolbox để hiển thị nó (thường nằm ở bên trái màn hình). Click chuột vào button  (Auto Hide) nằm ở góc trên phải của cửa sổ ToolBox để chuyển nó về chế độ hiển thị thường trực.
6. Duyệt tìm phần tử Button (trong nhóm Common Controls), chọn nó, dời chuột về góc trên trái của form và vẽ nó với kích thước mong muốn. Xem cửa sổ thuộc tính của Button vừa vẽ, duyệt tìm và hiệu chỉnh thuộc tính Text = "Browse...", duyệt tìm và thay đổi thuộc tính (Name) = btnBrowse.
7. Duyệt tìm phần tử TextBox (trong nhóm Common Controls), chọn nó, dời chuột về phía bên phải Button vừa vẽ và vẽ TextBox với kích thước mong muốn. Xem cửa sổ thuộc tính của TextBox vừa vẽ, duyệt tìm và hiệu chỉnh thuộc tính (Name) = txtPath.
8. Lập lại bước 6 để vẽ thêm button ngay bên phải TextBox với thuộc tính Text = "Start", (Name) = btnStart.
9. Duyệt tìm phần tử Label (trong nhóm Common Controls), chọn nó, dời chuột về ngay dưới button Browse và vẽ nó với kích thước mong muốn. Xem cửa sổ thuộc tính của Label vừa vẽ, duyệt tìm và hiệu chỉnh thuộc tính Text = "Các process đã được kích hoạt và đang chạy :".
10. Duyệt tìm phần tử ListBox (trong nhóm Common Controls), chọn nó, dời chuột về ngay dưới Label và vẽ ở bước 9 rồi vẽ nó với kích thước mong muốn. Xem cửa sổ thuộc tính của ListBox vừa vẽ, duyệt tìm và hiệu chỉnh thuộc tính (Name) = lbOutput.
11. Lập lại bước 6 để vẽ thêm button ngay bên dưới trái ListBox với thuộc tính Text = "Refresh", (Name) = btnRefresh.

12. Lặp lại bước 6 để vẽ thêm button ngay bên dưới phải ListBox với thuộc tính Text = "Stop", (Name) = btnStop.

Sau khi thiết kế xong, Form có dạng sau:



13. Dời chuột về button btnBrowse, ấn kép chuột vào nó để tạo hàm xử lý sự kiện Click chuột cho button, cửa sổ mã nguồn sẽ hiển thị để ta bắt đầu viết code cho hàm. Cách tổng quát để tạo hàm xử lý sự kiện là chọn đối tượng btnBrowse, cửa sổ thuộc tính của nó sẽ hiển thị, click icon  để hiển thị danh sách các sự kiện của đối tượng, duyệt tìm sự kiện quan tâm (Click), ấn kép chuột vào comboBox bên phải sự kiện Click để máy tạo tự động hàm xử lý cho sự kiện này. Cửa sổ mã nguồn sẽ hiển thị khung sườn của hàm vừa được tạo với thân rỗng, viết thân cho hàm này như sau:

```
private void btnBrowse_Click(object sender, EventArgs e)
{
    //tạo form duyệt chọn file khả thi cần chạy
    OpenFileDialog dlg = new OpenFileDialog();
    //hiển thị form duyệt chọn file khả thi cần chạy
    DialogResult ret = dlg.ShowDialog();
    //kiểm tra quyết định của người dùng, nếu người dùng chọn OK thì ghi nhận tên file
    if (ret == DialogResult.OK)
        txtPath.Text = dlg.FileName;
}
```

14. Dời chuột về button btnStart, ấn kép chuột vào nó để tạo hàm xử lý sự kiện Click chuột cho button, cửa sổ mã nguồn sẽ hiển thị khung sườn của hàm vừa được tạo với thân rỗng, viết thân cho hàm này như sau:

```
private void btnStart_Click(object sender, EventArgs e) {
    //tạo mới đối tượng quản lý Process.
    Process myProcess = new Process();
    try {
        //thiết lập đường dẫn file cần chạy
        myProcess.StartInfo.FileName = txtPath.Text;
        //kích hoạt process
        myProcess.Start();
        //hiển thị danh sách các process đang chạy
        DisplayProcess();
    }
    catch (Exception ex) {
        MessageBox.Show(ex.Message);
    }
}
```

```
}
```

15. Dời chuột về button btnStop, ấn kép chuột vào nó để tạo hàm xử lý sự kiện Click chuột cho button, cửa sổ mã nguồn sẽ hiển thị khung sườn của hàm vừa được tạo với thân rỗng, viết thân cho hàm này như sau:

```
private void btnStop_Click(object sender, EventArgs e) {
    //xác định tên, ID của process đang được chọn để xóa
    String[] param = lbOutput.SelectedItem.ToString().Split(',');
    //tìm đối tượng quản lý process có ID xác định
    Process proc = Process.GetProcessById(Int32.Parse(param[1]));
    //xóa process
    proc.Kill();
}
```

16. Dời chuột về button btnRefresh, ấn kép chuột vào nó để tạo hàm xử lý sự kiện Click chuột cho button, cửa sổ mã nguồn sẽ hiển thị khung sườn của hàm vừa được tạo với thân rỗng, viết thân cho hàm này như sau:

```
private void btnRefresh_Click(object sender, EventArgs e)
{
    //hiển thị lại danh sách các process đang chạy hiện hành
    DisplayProcess();
}
```

17. Dời chuột về đầu class Form1 rồi thêm các lệnh định nghĩa thuộc tính và hàm dịch vụ cần dùng như sau:

```
void DisplayProcess() {
    //xóa nội dung cũ của ListBox
    lbOutput.Items.Clear();
    //tìm danh sách các process đang chạy
    Process[] plist = Process.GetProcesses();
    //hiển thị tên, ID của từng process
    foreach (Process process in plist) {
        lbOutput.Items.Add(process.ProcessName + ", " + process.Id );
    }
}
```

18. Hiệu chỉnh lại hàm khởi tạo Form theo nội dung như sau :

```
public Form1() {
    InitializeComponent();
    //hiển thị danh sách các process đang chạy
    DisplayProcess();
}
```

19. Dời chuột về đầu file mã nguồn Form1 rồi thêm lệnh using như sau :

```
using System.Diagnostics;
```

20. Chọn menu Debug.Start Debugging để dịch và chạy thử ứng dụng, form chương trình hiển thị sẽ hiển thị y như bản thiết kế.

21. Hãy click chuột vào button Browse, cửa sổ duyệt chọn file sẽ hiển thị, hãy duyệt và chọn 1 file khả thi (*.exe) cần chạy. Khi cần kích thoát chạy thật sự, hãy click chuột vào button Start.

22. Lặp lại bước 21 nhiều lần để chạy nhiều ứng dụng khác nhau.

23. Khi cần xóa chương trình đang chạy nào, chọn nó trong Listbox rồi click chuột vào button Stop.

24. Khi cần hiển thị lại danh sách các process đang chạy hiện hành, click chuột vào button Refresh.
25. Lặp lại các bước 21, 23, 24 theo ý thích để kiểm tra việc kích hoạt và xóa từng process.