



TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Project 1: MyShell



Môn: Nguyên lý hệ điều hành.

Người làm:

- Nguyễn Đại An 20215296
- Nguyễn Tuấn Anh 20210015
- Trần Hồ Khánh Ly 20210561

ONE LOVE. ONE FUTURE.

1. Mục tiêu
2. Thành phần
 1. Chương trình myShell.
 2. Các chương trình con (plugins).
3. Thu hoạch

Mục tiêu

Sử dụng tối đa thư viện windows.h (Win32) và hạn chế phụ thuộc vào các câu lệnh của C++ (trừ các kiểu dữ liệu có sẵn và iostream).

```
ib > C shellWindow.h > {} ms > [E] paren
#include <windows.h>
#include <windowsx.h>
#include <string>
#include <vector>
#include "shellCommand.h"
namespace ms {
```

```
> C shellSetting.h > {} ms > [E] ModeExec
#include <aclapi.h>
#include <iostream>
#include <windows.h>
#include <sstream>
#include <string>
#include <vector>

#include "shellUtility.h"
```

Thành phần: myShell.exe

WinMain: Khởi tạo chương trình có GUI cho hệ điều hành Window

```
Shell.cpp > WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
#include "../src/lib/shellWindow.h"
#include <iostream>

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
                  LPSTR lpCmdLine, int nCmdShow) {
    ms::initPathConfig();
    return ms::createWindow(hInstance, hPrevInstance, lpCmdLine, nCmdShow);
}
```

WinMain: Khởi tạo chương trình có GUI cho hệ điều hành Window

```
bool initPathConfig() {  
    WCHAR a[settingBuff];  
    readConfig(L"PATH_CONFIG", a);  
    std::wstring b(a);  
    if (b.size() == 0) {  
        writeConfig(L"PATH_CONFIG", L"");  
        return 0;  
    }  
    PathConfig = ms::suSplitCommand(b, configDelimiter);  
    return 1;  
}
```

WinMain: Khởi tạo chương trình có GUI cho hệ điều hành Window

```
int createWindow(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {
    PathFileDirectory = getCurrentPath().append(L"\\");
    PathCurrentDirectory = PathFileDirectory;
    currentDirectoryText=shellName;
    currentDirectoryText.append(PathCurrentDirectory).append(L">");
    FreeConsole();
    WNDCLASSEXW wc = {};
    ZeroMemory(&wc, sizeof(WNDCLASSEX));
    wc.cbSize = sizeof(WNDCLASSEX);
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc = hwndMainProc;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    wc.hInstance = hInstance;
    wc.hIcon = (HICON)LoadImage(NULL, TEXT(".\\src\\img\\logo.ico"), IMAGE_ICON, 0, 0, LR_LOADFROMFILE);
    if (wc.hIcon == NULL) {
        printf("Load Image Error: %x\n", GetLastError());
    }
    wc.hIconSm = wc.hIcon;
```

Thành phần: myShell.exe

FreeConsole(): Tách rời chương trình sắp tạo ra khỏi console được gán bởi hệ điều hành, giúp các thao tác vẽ lại UI diễn ra thuận lợi.

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
                  LPSTR lpCmdLine, int nCmdShow) {
    // TODO:
    PathFileDirectory = getCurrentPath().append(L"\\");
    PathCurrentDirectory = PathFileDirectory;
    FreeConsole();
    std::cout << GetLastError();
    WNDCLASSEXW wc = {};
    // ZeroMemory(&wc, sizeof(WNDCLASSEX));
    wc.cbSize = sizeof(WNDCLASSEX);
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc = WndProc;
```

WNDCLASSEXW: Khai báo các thông tin về kiểu của window sẽ tạo.

```
WNDCLASSEXW wc = {};  
ZeroMemory(&wc, sizeof(WNDCLASSEX));  
wc.cbSize = sizeof(WNDCLASSEX);  
wc.style = CS_HREDRAW | CS_VREDRAW;  
wc.lpfnWndProc = WndProc;  
wc.cbClsExtra = 0;  
wc.cbWndExtra = 0;  
wc.hInstance = hInstance;  
wc.hIcon = (HICON)LoadImage(NULL, TEXT(".\\src\\img\\logo.ico"), IMAGE_ICON, 0, 0, LR_LOADFROMFILE);  
if (wc.hIcon == NULL) {  
    printf("Load Image Error: %x\\n", GetLastError());  
}  
wc.hIconSm = wc.hIcon;  
  
wc.hCursor = LoadCursor(NULL, IDC_ARROW);  
wc.hbrBackground = CreateSolidBrush(RGB(0, 255, 255));  
// TODO:  
wc.lpszMenuName = NULL;  
wc.lpszClassName = &parentClassName[0];  
  
if (!RegisterClassExW(&wc)) {  
    MessageBox(NULL, TEXT("Window Registration Failed!"), TEXT("myShell"),  
        MB_ICONEXCLAMATION | MB_OK);  
    return 0;  
}
```

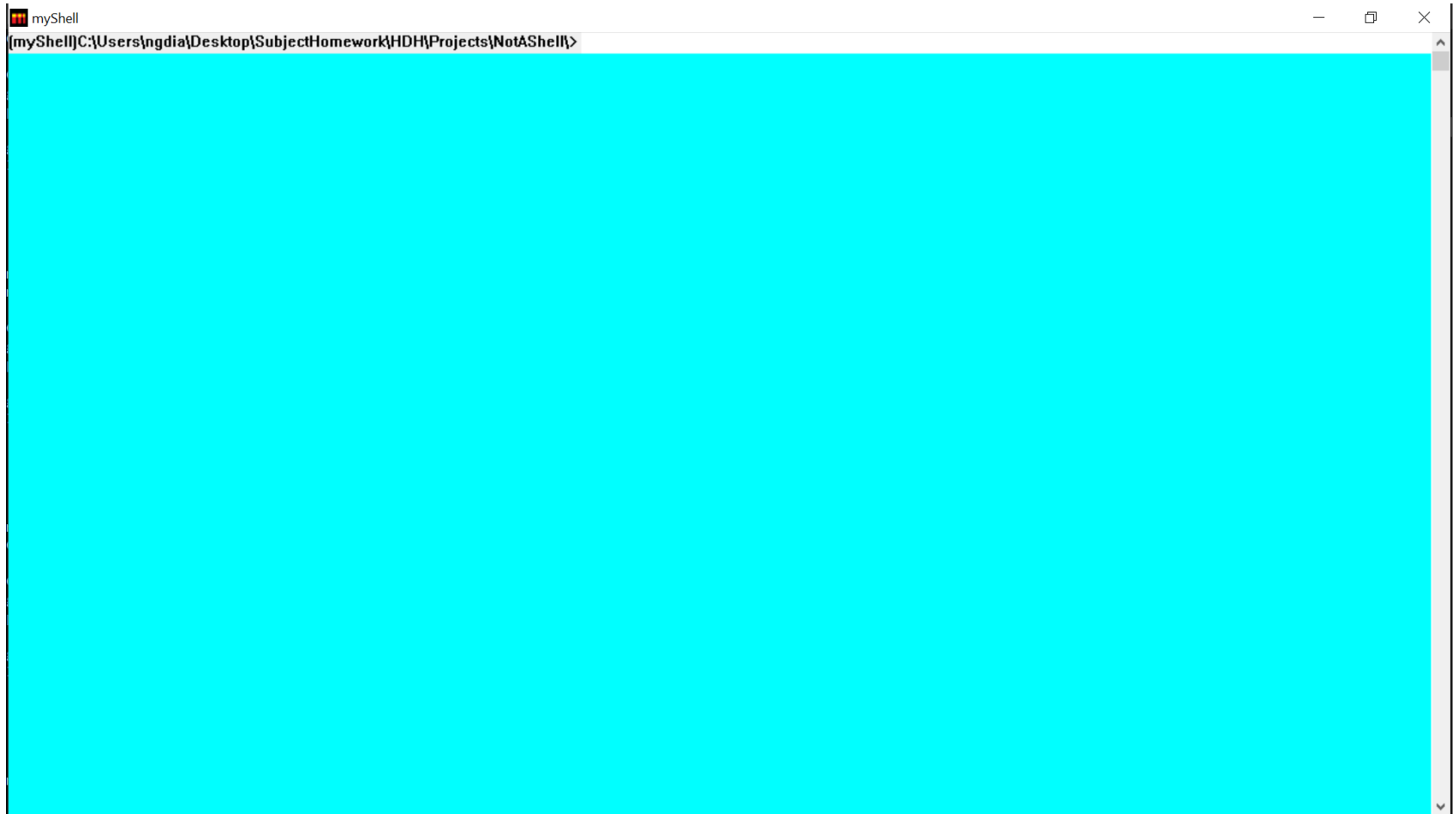

Thành phần: myShell.exe

Tiến trình bắt đầu bằng cách khởi tạo cửa sổ mới với giá trị HWND lưu trong hwndMain.


Cửa sổ có tiêu đề là myShell.

```
hwndMain = CreateWindowExW(  
    WS_EX_CLIENTEDGE,  
    parentClassName,  
    L"myShell",  
    WS_OVERLAPPEDWINDOW | WS_VSCROLL | WS_CLIPCHILDREN | WS_CLIPSIBLINGS,  
    CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,  
    NULL, NULL, hInstance, NULL);  
ShowWindow(hwndMain, nCmdShow);
```

Thành phần: myShell.exe



Thành phần: myShell.exe

 myShell

(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>

Thành phần: myShell.exe

Trong cửa sổ chính sẽ có 2 cửa sổ con với giá trị HWND lưu trong hwndStatic và hwndEdit.

1. Cửa sổ “static” thuộc kiểu Edit Control, được khai báo giá trị ES_READONLY nhằm mục đích lưu lại lịch sử tương tác của người dùng.
2. Cửa sổ “edit” thuộc kiểu Edit Control phục vụ việc nhập lệnh. Cửa sổ cho phép nhập trên 1 dòng duy nhất và cho phép scroll theo chiều ngang.



Thành phần: myShell.exe

Trong cửa sổ chính sẽ có 2 cửa sổ con với hwnd lưu trong hwndStatic và hwndEdit.

```
staticContent = currentDirectoryText;
staticRect = GetStringBorderW(staticContent, NULL);
LINE_SIZE = staticRect.bottom - staticRect.top;
hwndStatic = CreateWindowW(L"Edit",
    (LPWSTR)staticContent.data(),
    WS_CHILD | WS_VISIBLE | ES_READONLY | ES_MULTILINE | WS_CLIPSIBLINGS,
    staticRect.left, staticRect.top, staticRect.right, staticRect.bottom,
    hwndMain,
    0,
    hInstance, NULL);
hwndEdit = CreateWindowW(L"Edit",
    (LPWSTR)editContent.data(),
    WS_CHILD | WS_VISIBLE | ES_AUTOHSCROLL | WS_HSCROLL | WS_CLIPSIBLINGS,
    staticRect.right, staticRect.top, windowRect.right, staticRect.bottom,
    hwndMain,
    0,
    hInstance, NULL);
ShowScrollBar(hwndEdit, SB_BOTH, FALSE);
```

Xử lý 1 số tín hiệu như:

1. Tổ hợp Ctrl+E khi đang nhập lệnh gửi message kết thúc chương trình.
2. Phím Enter thực thi câu lệnh hiện được gõ trong cửa sổ edit.
3. Tổ hợp Ctrl+C kill process gần nhất.

Thành phần: myShell.exe

Vẽ lại window khi maximize, minimize hoặc có thao tác thực hiện lệnh (Nhấn Enter) từ người dùng.

```
break;
case WM_SIZE:
    result = DefWindowProcW(hwnd, msg, wParam, lParam);
    if (wParam == SIZE_MINIMIZED || hwndEdit == NULL) {
        break;
    }
    GetWindowRect(hwnd, &windowRect);
    windowRect.right = windowRect.right - windowRect.left;
    windowRect.bottom = windowRect.bottom - windowRect.top;
    windowRect.left = 0;
    windowRect.top = 0;
    staticRect = GetStringBorderW(staticContent, hwndStatic);
    RECT directRect;
    directRect = GetStringBorderW(currentDirectoryText, hwndStatic);
    SetWindowPos(hwndStatic, HWND_BOTTOM, staticRect.left, staticRect.top,
        staticRect.right - staticRect.left, staticRect.bottom - staticRect.top, SWP_NOMOVE);
    SetWindowPos(hwndEdit, HWND_TOP, directRect.right,
        staticRect.bottom - LINE_SIZE, windowRect.right - directRect.right, LINE_SIZE, SWP_SHOWWINDOW);
    break;
```

Hàm callback của editControl.

```
LRESULT CALLBACK hwndEditProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) {  
    if (msg == WM_CHAR) {  
        switch (wParam) {  
            case 1: // Ctrl+A  
                SendMessage(hwnd, EM_SETSEL, 0, -1);  
                return 1;  
            case 3: // Ctrl+C  
                if(ModeExecution.compare(L"foreground") == 0){  
                    buildin_child_kill(sc_lastCreatedChild.hProcess);  
                }  
                return 1;  
            case 5: // Ctrl+E  
                SendMessage(hwndMain, WM_CLOSE, 0, 0);  
                return 1;  
        }  
    }  
    return 0;  
}
```


Hàm callback của editControl.

```
case 13: // Enter
    //Get content of Edit window
    editContent.resize(GetWindowTextLength(hwnd));
    editContent.resize(GetWindowTextW(hwnd, (LPWSTR)editContent.data(), editContent.size() + 1));
    //Append content to Static window
    staticContent.append(editContent).append(L"\r\n"); // NewLine OK
    scExecuteCommand(hwndMain, staticContent, editContent, PathFileDirectory, PathCurrentDirectory);
    staticContent.append(currentDirectoryText); // No NewLine OK

    editContent = L"";

    SetWindowTextW(hwndStatic, staticContent.c_str());

    SetWindowTextW(hwndEdit, editContent.c_str());

    SendMessage(hwndMain, WM_SIZE, SIZE_RESTORED, 0);
    return 1;
```

Chương trình sẽ gọi function

`executeCommand(std::wstring command)`

Để thực hiện câu lệnh.

1. Kiểm tra câu lệnh có tồn tại trong số các built-ins/plugins được định nghĩa trước hay trong địa chỉ PATH do người dùng thêm vào không.
2. Nếu câu lệnh tồn tại, khởi tạo tiến trình ứng với câu lệnh thông qua `createProcess`. Ở mode foreground thì chuyển hướng I/O của chương trình con sang pipe để thao tác với chương trình mẹ.
3. Nếu câu lệnh không tồn tại, in ra dòng thông báo lỗi.

- “Plugins”: Các chương trình ứng với các câu lệnh được cài đặt sẵn của myShell được định nghĩa trong thư mục plugins.
- “Built-in”: Các function tích hợp sẵn trong phần mềm myshell.


Thành phần: Các câu lệnh

```
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>help
List of built-in command: cd, exit, mode, cls, help, child, path.
List of plugins available:
- date.exe
- echo.exe
- list.exe
- path.exe
- time.exe
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell> |
```

Thành phần: Các câu lệnh

```
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>help
List of built-in command: cd, exit, mode, cls, help, child, path.
List of plugins available:
- date.exe
- echo.exe
- list.exe
- path.exe
- time.exe
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell> |
```

Thành phần: Các câu lệnh

 myShell

```
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>echo hello  
hello
```

```
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>mode  
Current shell operating mode: foreground
```

```
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>date  
Current date: 29/6/2023
```

```
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>time  
Current time: 14:27:912 46
```

```
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>list
```

```
<unknown> (PID: 4)
```

```
<unknown> (PID: 56)
```

```
<unknown> (PID: 108)
```

```
<unknown> (PID: 424)
```

```
<unknown> (PID: 656)
```

```
<unknown> (PID: 868)
```

```
<unknown> (PID: 876)
```

```
<unknown> (PID: 940)
```

```
<unknown> (PID: 960)
```

```
<unknown> (PID: 968)
```

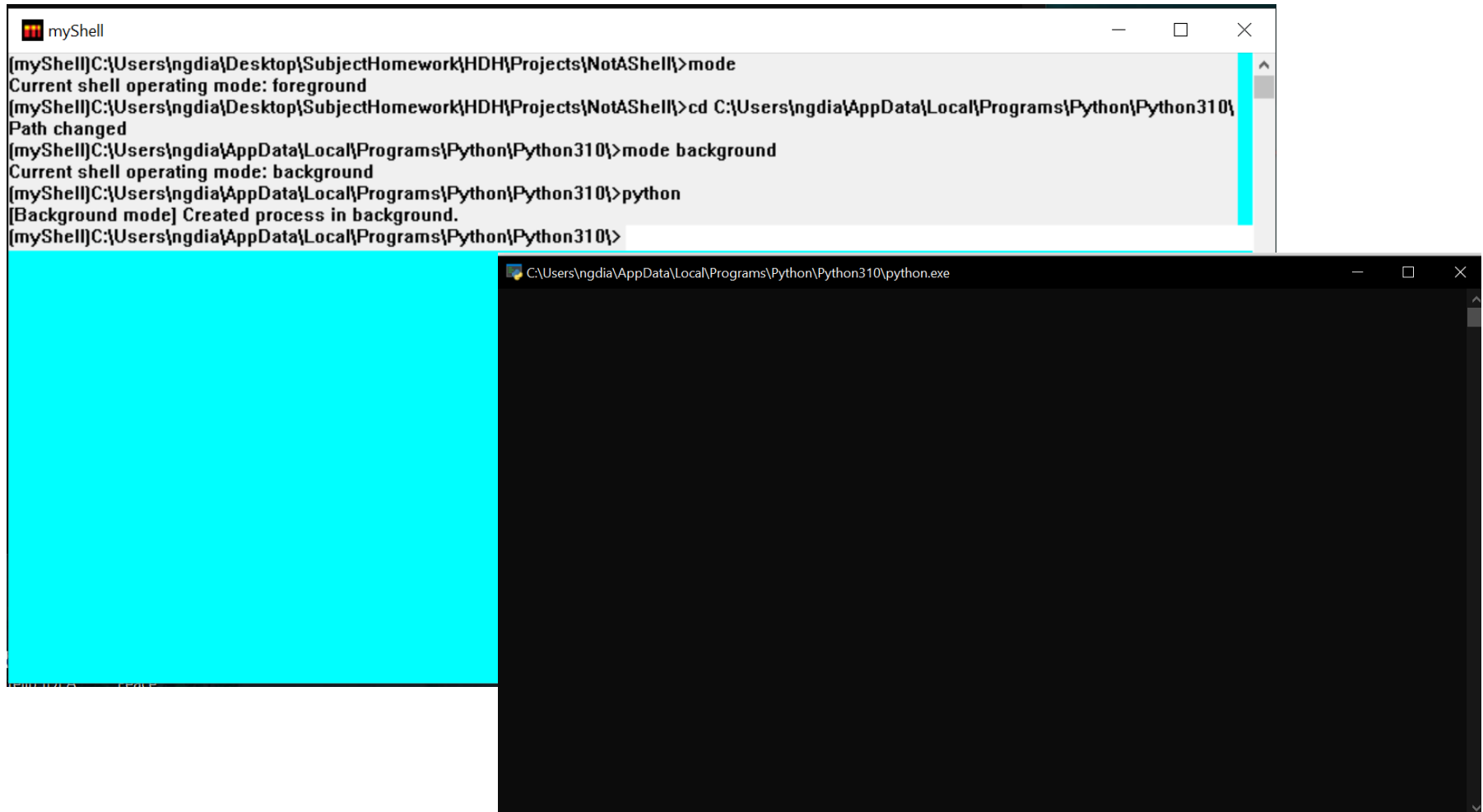
```
<unknown> (PID: 632)
```

```
<unknown> (PID: 704)
```

```
<unknown> (PID: 724)
```

```
<unknown> (PID: 808)
```

Thành phần: myShell.exe




The image shows two overlapping windows. The top window, titled 'myShell', is a command prompt with a red icon. It displays the following text:

```
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>mode
Current shell operating mode: foreground
(myShell)C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>cd C:\Users\ngdia\AppData\Local\Programs\Python\Python310\
Path changed
(myShell)C:\Users\ngdia\AppData\Local\Programs\Python\Python310>mode background
Current shell operating mode: background
(myShell)C:\Users\ngdia\AppData\Local\Programs\Python\Python310>python
[Background mode] Created process in background.
(myShell)C:\Users\ngdia\AppData\Local\Programs\Python\Python310>
```

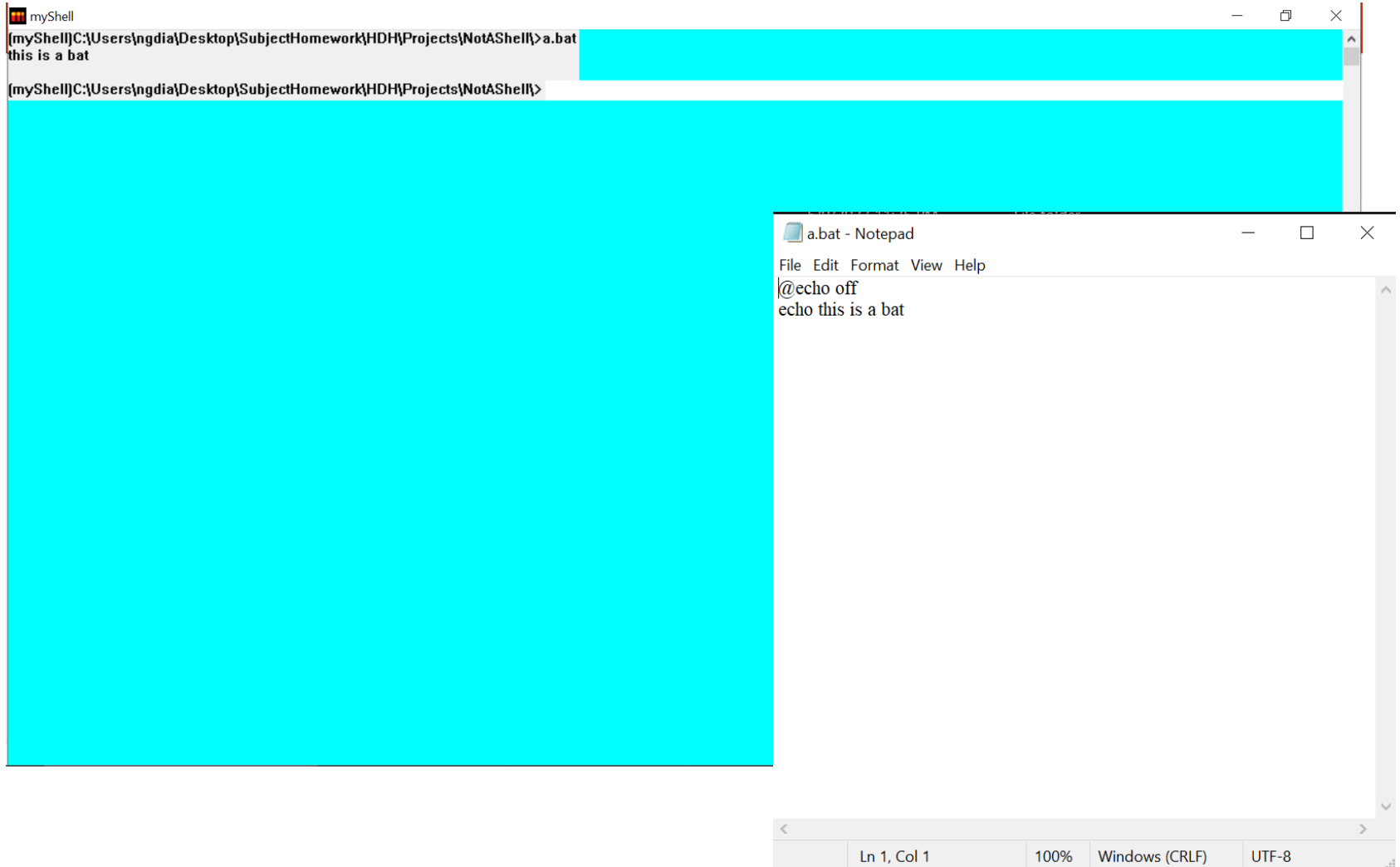
The bottom window, titled 'C:\Users\ngdia\AppData\Local\Programs\Python\Python310\python.exe', is a black terminal window with a blue icon, representing the Python interpreter running in the background.

Thành phần: myShell.exe

 myShell — □ ×

```
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>mode
Current shell operating mode: foreground
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>cd C:\Users\ngdia\AppData\Local\Programs\Python\Python310\
Path changed
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>mode background
Current shell operating mode: background
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>python
[Background mode] Created process in background.
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>child
List of active child processes managed by MyShell:
1-Child Process 6136
++Thread ID: 6348
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>child suspend 6136
Suspend process successfully!
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>child resume 6136
Resume process successfully!
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>child kill 6136
Kill process successfully!
[myShell]C:\Users\ngdia\AppData\Local\Programs\Python\Python310>
```


Thành phần: myShell.exe



```
myShell
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>a.bat
this is a bat
[myShell]C:\Users\ngdia\Desktop\SubjectHomework\HDH\Projects\NotAShell>

a.bat - Notepad
File Edit Format View Help
@echo off
echo this is a bat
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Về win32:

- Cũ, mã nguồn đóng, khó sử dụng.
- Được thay thế bởi .NET.

Về shell:

- Shell là 1 phần mềm khá thú vị.
- Việc sử dụng pipe đã giúp nhóm hiểu thêm về console.

HUST

