

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công Nghệ Thông Tin và Truyền Thông



BÀI TẬP LỚN NGUYÊN LÝ HỆ ĐIỀU HÀNH

Đề tài: Xây dựng ứng dụng TINYSHELL

Giảng Viên: Thầy Phạm Đăng Hải

Mã lớp: 124154

Sinh viên thực hiện

Họ tên: Lê Viết Hoàng

MSSV: 20173128

Hà Nội, tháng 06 năm 2021

Mục lục

Mục lục.....	2
Danh mục hình ảnh	2
Lời cảm ơn	3
I. Mở đầu	4
II. Cơ sở lý thuyết và công cụ.....	4
2.1. Công cụ lập trình.....	4
2.2. Ngôn ngữ lập trình	4
2.3. Thư viện lập trình.....	5
III. Xây dựng chương trình.....	5
IV. Cài đặt và kiểm thử.....	8
4.1. Cài đặt môi trường	8
4.2. Kiểm thử.....	8
4.3. Hướng phát triển	12
V. Kết luận	12

Danh mục hình ảnh

Hình 1: Giao diện chương trình	8
Hình 2: Các lệnh time, date, dir	9
Hình 3: Lệnh cls reset lại giao diện chương trình.....	9
Hình 4: Lệnh help	10
Hình 5: Chạy file *.bat.....	10
Hình 6: Chạy ứng dụng clock	11
Hình 7: Chạy ứng dụng calc và clock chế độ nền.....	11
Hình 8: sau khi kill ứng dụng clock.....	12

Lời cảm ơn

Lời đầu tiên, chúng em xin chân thành cảm ơn và bày tỏ lòng biết ơn sâu sắc đến thầy giáo Phạm Đăng Hải. Thầy đã tận tình hướng dẫn, đồng thời cung cấp nhiều tài liệu để giúp em hoàn thiện đề tài này.

Học phần “Nguyên lý hệ điều hành” là một học phần rất quan trọng, được đưa vào giảng dạy tại hầu hết các trường đại học có các ngành về Công nghệ Thông tin nói chung. Dưới sự hướng dẫn của thầy, em đã tiếp thu được nhiều kiến thức bổ ích từ môn học này, góp phần cải thiện kỹ năng lập trình cũng như nâng cao hiểu biết sâu về nguyên lý hoạt động của hệ điều hành. Qua đó, mở rộng tư duy giải quyết các bài toán trong thực tiễn.

Em xin chân thành cảm ơn!

Hà Nội, tháng 06 năm 2021

Sinh viên thực hiện

Hoàng

Lê Viết Hoàng

I. Mở đầu

Tên đề tài: Thiết kế và cài đặt một shell đơn giản

Nội dung:

- Shell nhận lệnh, phân tích và tạo tiến trình con thực hiện
- Shell chứa các câu lệnh quản lý tiến trình
- Shell hiểu một số lệnh đặc biệt
- Shell có thể nhận tín hiệu ngắt từ bàn phím để hủy bỏ foreground process đang thực hiện (CTRL + C)
- Shell có thể thực thi file *.bat

II. Cơ sở lý thuyết và công cụ

2.1. Công cụ lập trình

Microsoft Visual Studio Code là một công cụ soạn thảo mã nguồn (Code Editor – nôm na là chỗ để viết code) do Microsoft phát triển, được giới thiệu lần đầu năm 2015 và chính thức phát hành năm 2016. VSCode có thể cài đặt và sử dụng trên cả Windows, MacOS và Linux. VSCode là mã nguồn mở và nó hoàn toàn miễn phí.

Có thể nói VSCode là sự kết hợp tuyệt vời giữa tính đơn giản của một editor và các công cụ hỗ trợ mạnh mẽ dành cho lập trình viên như Debugger, Git, Terminal, và còn nhiều hơn nữa. Đúng vậy, nhìn chung thì VSCode vẫn chỉ là một Code Editor nhưng độ hữu ích thì không kém cạnh IDE nào.

Dù mới được phát hành trong khoảng 4-5 năm trở lại đây, VSCode đã trở thành một trong những Code Editor mạnh mẽ và phổ biến nhất dành cho lập trình viên. Nhờ hỗ trợ nhiều ngôn ngữ lập trình phổ biến, đầy đủ tính năng và khả năng mở rộng, nên VSCode trở nên cực kỳ thân thuộc và dễ sử dụng với bất kỳ lập trình viên nào.

2.2. Ngôn ngữ lập trình

Ngôn ngữ C++ là một loại ngôn ngữ lập trình bậc trung (middle-level). Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp thấp. Từ thập niên 1990, C++ đã trở thành một trong những ngôn ngữ thương mại ưa thích và phổ biến của lập trình viên.

C++ được thiết kế hướng tới lập trình hệ thống máy tính và phần mềm nhúng trên các mạch vi xử lý, bao gồm cả hệ thống có tài nguyên hạn chế và tài nguyên khổng lồ,

với ưu điểm vượt trội về hiệu suất, hiệu quả và tính linh hoạt cao. C++ có thể tìm thấy ở mọi nơi, với những điểm mạnh là cơ sở hạ tầng phần mềm và các ứng dụng bị hạn chế tài nguyên. bao gồm: phần mềm ứng dụng máy tính cá nhân, trò chơi điện tử, các hệ thống máy chủ (ví dụ: phần mềm thương mại điện tử, cỗ máy tìm kiếm trên web hoặc máy chủ SQL) và các ứng dụng ưu tiên về hiệu suất (ví dụ: tổng đài thông tin liên lạc hoặc thiết bị thăm dò không gian). C++ hầu hết được thực thi dưới dạng là một ngôn ngữ biên dịch, có thể chạy trên nhiều nền tảng khác nhau như Windows, Mac OS, Linux, Ubuntu và các phiên bản Unix. Nhiều nhà cung cấp cung cấp các trình biên dịch C++, bao gồm Tổ chức Phần mềm Tự do, Microsoft, Intel và IBM.

2.3. Thư viện lập trình

- Trình biên dịch C/C++ MinGW
- Thư viện C/C++ chuẩn
- Thư viện Win32 API (Application Programming Interface) nằm trong Windows.h cung cấp các thư viện liên kết động (.dll) chứa các hàm truy cập tài nguyên trong hệ thống Windows. Ví dụ kiểu dữ liệu HANDLE số nguyên 32 bit được hệ điều hành dùng để định danh cho một đối tượng tài nguyên nào đó như: file, socket, ...

III. Xây dựng chương trình

Các hàm được sử dụng:

- **Tên hàm:**

```

BOOL CreateProcessA(
    LPCSTR          lpApplicationName,
    LPSTR           lpCommandLine,
    LPSECURITY_ATTRIBUTES lpProcessAttributes,
    LPSECURITY_ATTRIBUTES lpThreadAttributes,
    BOOL            bInheritHandles,
    DWORD           dwCreationFlags,
    LPVOID          lpEnvironment,
    LPCSTR          lpCurrentDirectory,
    LPSTARTUPINFOA  lpStartupInfo,
    LPPROCESS_INFORMATION lpProcessInformation
);

```

- **Tham số:**

lpApplicationName	Tên chương trình
lpCommandLine	Tham số dòng lệnh
lpProcessAttributes	Thuộc tính an ninh tiến trình
lpThreadAttributes	Thuộc tính an ninh luồng

bInheritHandles	Cho phép kế thừa các thẻ thiết bị
dwCreationFlags	Cờ tạo tiến trình
lpEnvironment	Trở tới khối môi trường
lpCurrentDirectory	Đường dẫn đầy đủ đến chương trình
lpStartupInfo	Cấu trúc thông tin tiến trình mới
lpProcessInformation	Thông tin về tiến trình mới

- **Chức năng:** Tạo một tiến trình mới

- **Tên hàm:**
BOOL TerminateProcess(
HANDLE hProcess,
UINT uExitCode
);
- **Tham số:**

hProcess	Thẻ tiến trình đóng
uExitCode	Mã kết thúc tiến trình đóng
- **Chức năng:** Hủy một tiến trình và các luồng của nó

- **Tên hàm:**
DWORD WaitForSingleObject(
HANDLE hHandle,
DWORD dwMilliseconds
);
- **Tham số:**

hHandle	Thẻ đối tượng
dwMilliseconds	Thời gian chờ đợi
- **Chức năng:** Đợi đến khi đối tượng được báo hiệu hoặc hết thời gian

- **Tên hàm:**
BOOL CloseHandle(
HANDLE hObject
);
- **Tham số:**

hObject	Thẻ đối tượng
---------	---------------
- **Chức năng:** Đóng một đối tượng

- **Tên hàm:**
void GetLocalTime(
LPSYSTEMTIME lpSystemTime
);
- **Tham số:**

lpSystemTime	Một con trỏ đến cấu trúc SYSTEMTIME để nhận ngày và giờ cục bộ hiện tại
--------------	---

- **Chức năng:** Truy xuất ngày và giờ địa phương hiện tại

- **Tên hàm:**
int system(const char *command);
- **Tham số:**

command	Lệnh mà hệ điều hành cung cấp
---------	-------------------------------
- **Chức năng:** gọi một lệnh hệ điều hành từ chương trình C / C ++

- **Tên hàm:**
bool checkFile(const char *name);
- **Tham số:**

name	Tên file
------	----------
- **Chức năng:** Kiểm tra file có tồn tại hay không

- **Tên hàm:**
void openApp(char *cmd, const char *path, MODE mode);
- **Tham số:**

cmd	Câu lệnh
path	Đường dẫn tới ứng dụng
mode	Chế độ mở
- **Chức năng:** Mở một ứng dụng và thêm vào danh sách tiến trình

- **Tên hàm:**
void addNewProcess(DWORD pid, const char *cmd);
- **Tham số:**

pid	Thông tin của tiến trình vừa được mở
cmd	Câu lệnh
- **Chức năng:** Thêm tiến trình vào danh sách quản lý

- **Tên hàm:**
void closeApp(DWORD pid);
- **Tham số:**

pid	Thông tin của tiến trình vừa được mở
-----	--------------------------------------
- **Chức năng:** Đóng tiến trình và xóa khỏi danh sách quản lý

- **Tên hàm:**
void sigintHandler(int sig_num);

- **Tham số:**

sig_num	Tín hiệu
---------	----------

- **Chức năng:** Nhận tín hiệu ngắt rồi kết thúc tiến trình

- **Tên hàm:**

string workingdir();

- **Tham số:**
- **Chức năng:** In ra đường dẫn thư mục hiện tại

IV. Cài đặt và kiểm thử

4.1. Cài đặt môi trường

- Cài đặt VScode: <https://code.visualstudio.com/download>
- Cài đặt MinGW C/C++: <https://sourceforge.net/projects/mingw-w64/>
- Build chương trình: `g++ -o shell .\main.cpp`
- Chạy chương trình: `.\shell`

4.2. Kiểm thử

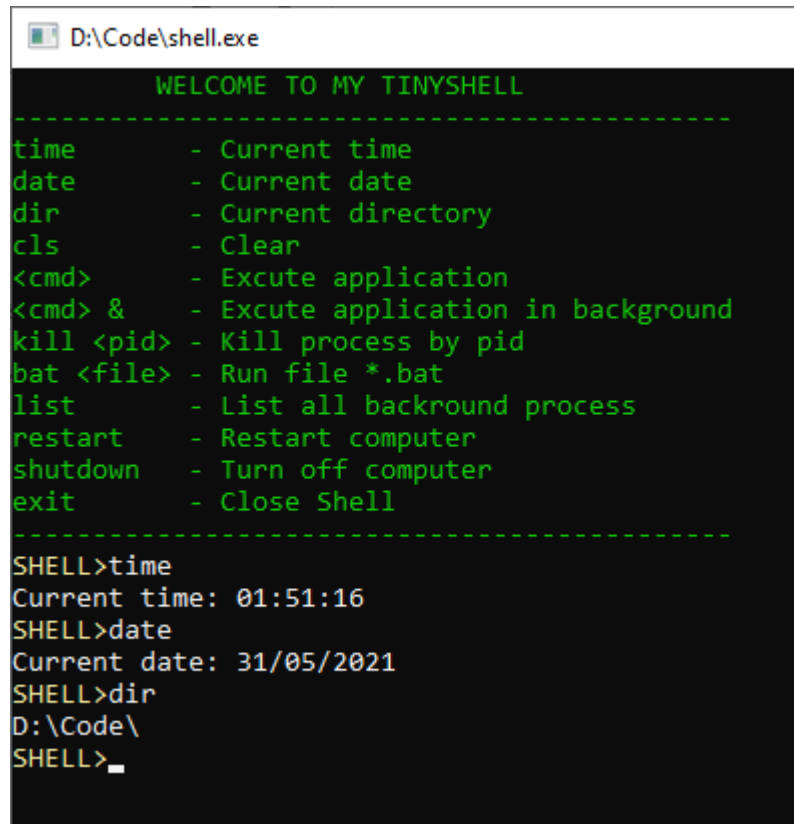
D:\Code\shell.exe

```

WELCOME TO MY TINY SHELL
-----
time      - Current time
date      - Current date
dir       - Current directory
cls       - Clear
<cmd>     - Excute application
<cmd> &   - Excute application in background
kill <pid> - Kill process by pid
bat <file> - Run file *.bat
list      - List all backround process
restart   - Restart computer
shutdown  - Turn off computer
exit      - Close Shell
-----
SHELL>_

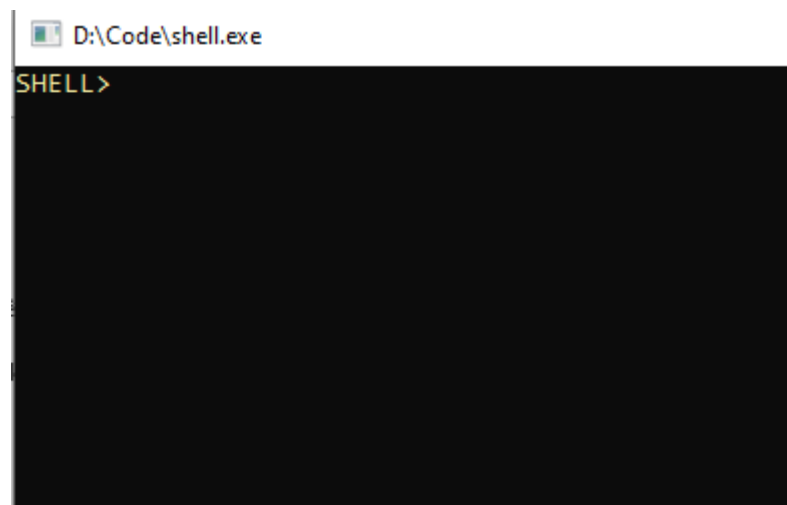
```

Hình 1: Giao diện chương trình



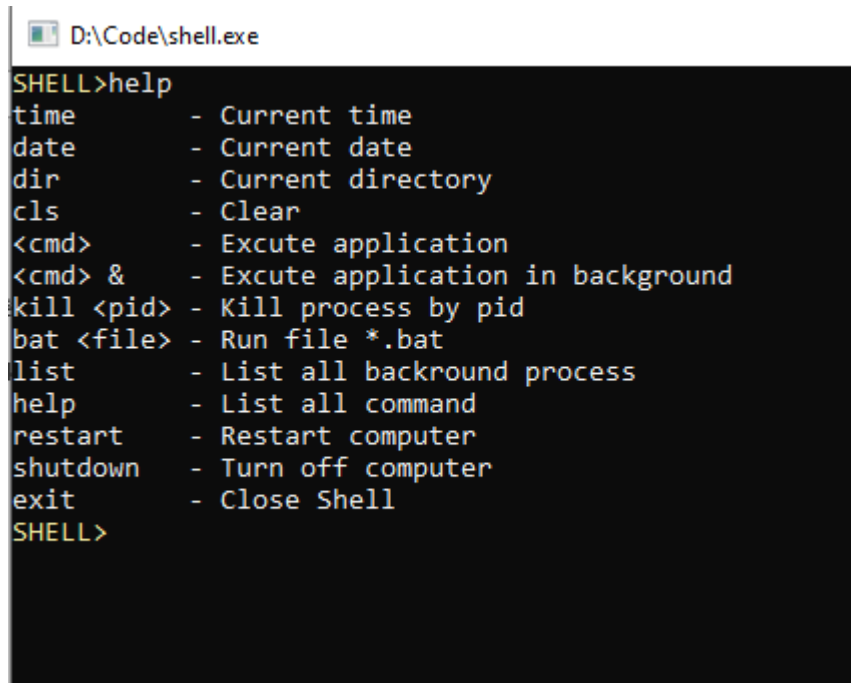
```
D:\Code\shell.exe
WELCOME TO MY TINYSHELL
-----
time      - Current time
date      - Current date
dir       - Current directory
cls       - Clear
<cmd>     - Excute application
<cmd> &   - Excute application in background
kill <pid> - Kill process by pid
bat <file> - Run file *.bat
list      - List all backround process
restart   - Restart computer
shutdown  - Turn off computer
exit      - Close Shell
-----
SHELL>time
Current time: 01:51:16
SHELL>date
Current date: 31/05/2021
SHELL>dir
D:\Code\
SHELL>_
```

Hình 2: Các lệnh time, date, dir



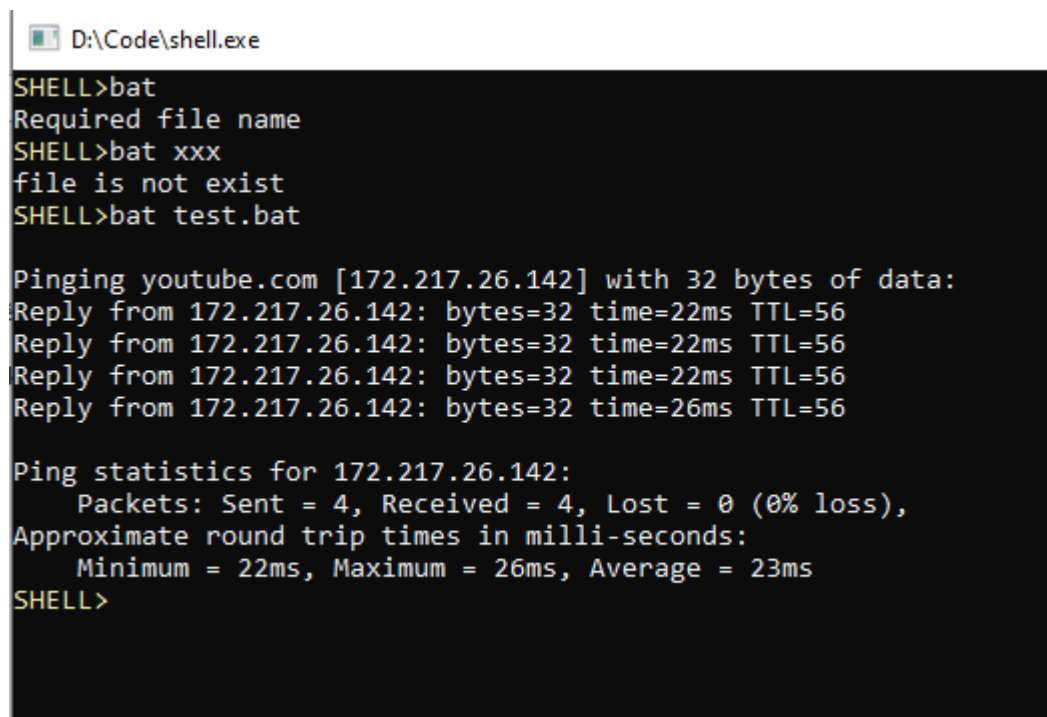
```
D:\Code\shell.exe
SHELL>
```

Hình 3: Lệnh cls reset lại giao diện chương trình



```
D:\Code\shell.exe
SHELL>help
time      - Current time
date      - Current date
dir       - Current directory
cls       - Clear
<cmd>     - Excute application
<cmd> &   - Excute application in background
kill <pid> - Kill process by pid
bat <file> - Run file *.bat
list      - List all backround process
help      - List all command
restart   - Restart computer
shutdown  - Turn off computer
exit      - Close Shell
SHELL>
```

Hình 4: Lệnh help

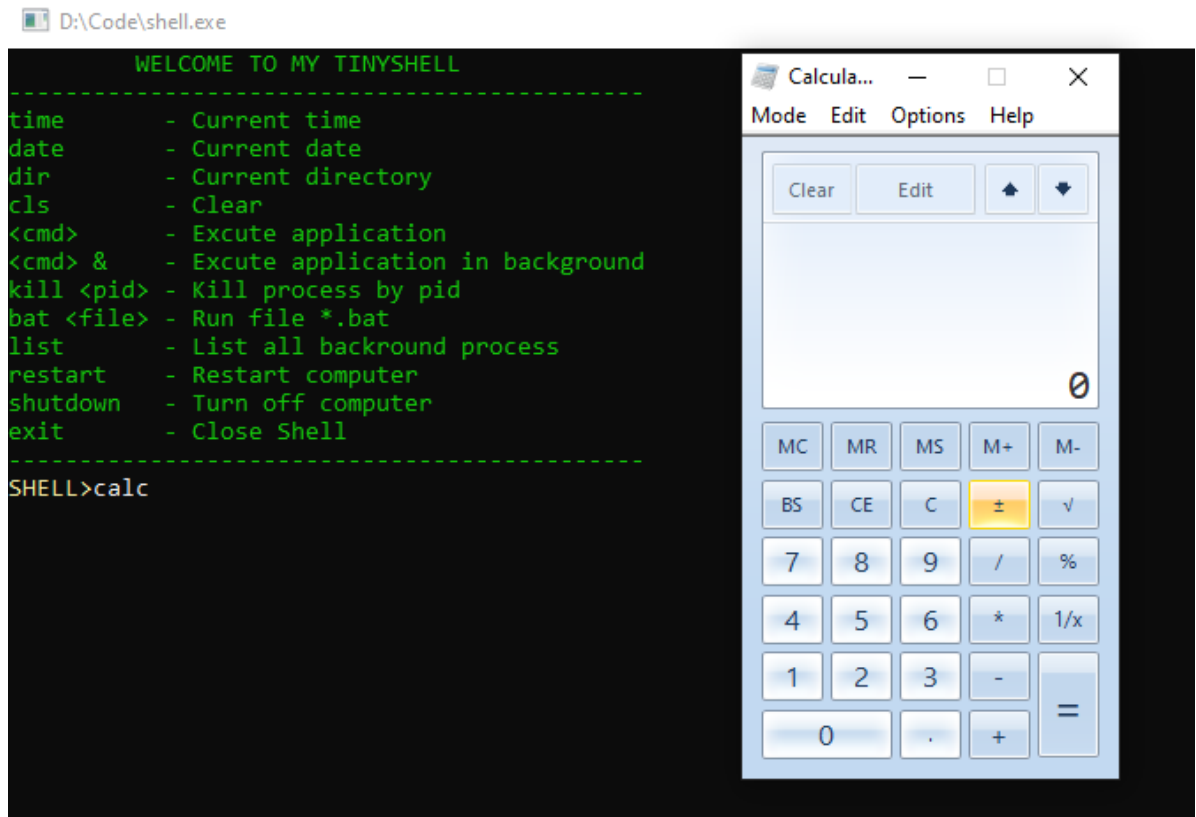


```
D:\Code\shell.exe
SHELL>bat
Required file name
SHELL>bat xxx
file is not exist
SHELL>bat test.bat

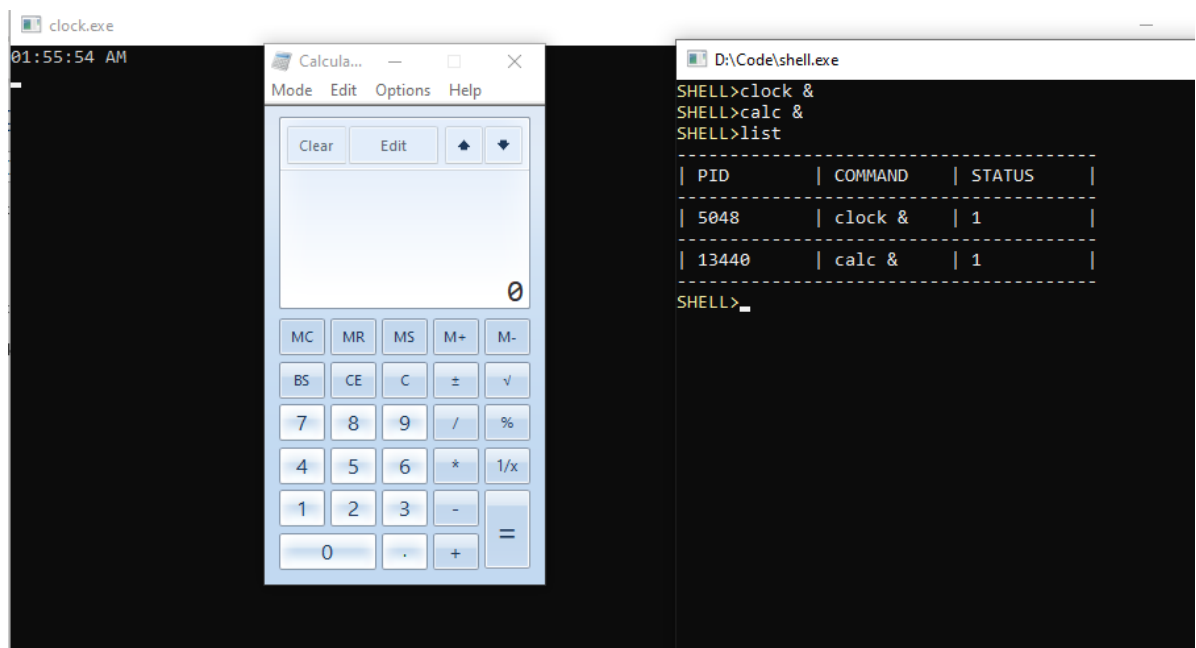
Pinging youtube.com [172.217.26.142] with 32 bytes of data:
Reply from 172.217.26.142: bytes=32 time=22ms TTL=56
Reply from 172.217.26.142: bytes=32 time=22ms TTL=56
Reply from 172.217.26.142: bytes=32 time=22ms TTL=56
Reply from 172.217.26.142: bytes=32 time=26ms TTL=56

Ping statistics for 172.217.26.142:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 22ms, Maximum = 26ms, Average = 23ms
SHELL>
```

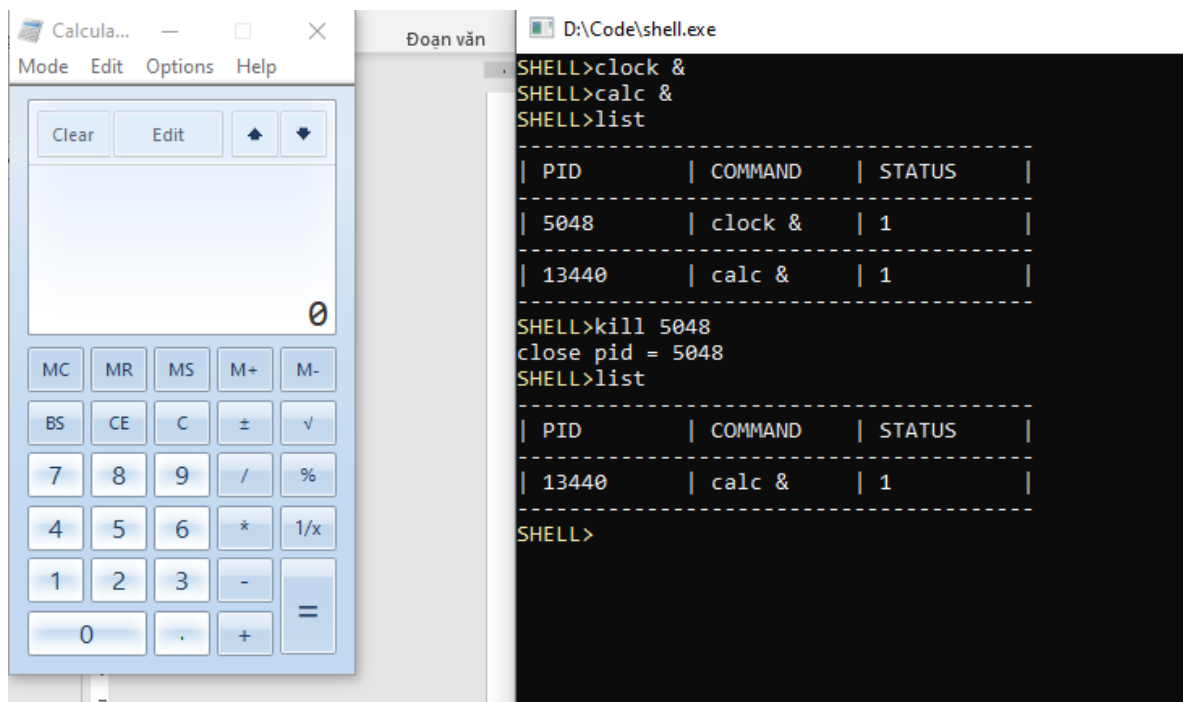
Hình 5: Chạy file *.bat



Hình 6: Chạy ứng dụng clock



Hình 7: Chạy ứng dụng calc và clock chế độ nền



Hình 8: sau khi kill ứng dụng clock

4.3. Hướng phát triển

- Tăng hiệu năng của chương trình thông qua việc tối ưu hóa mã nguồn.
- Thêm các chức năng như lấy danh sách file, thêm, sửa, xóa file.

V. Kết luận

Sau khi nhận đề tài này, dưới sự hướng dẫn của thầy, em đã thu nhận được nhiều kiến thức quý báu sẽ giúp ích cho việc học tập các học phần sau này.

- Nâng cao kỹ năng tự học
- Có hiểu biết sâu sắc hơn WIN API.
- Nâng cao kỹ năng lập trình C/C++. Có kỹ năng phân tách chức năng thành các hàm, giúp mã nguồn trong sáng, rõ ràng, dễ quản lý.
- Có kỹ năng viết báo cáo, trình bày báo cáo, mã nguồn chương trình một cách khoa học, mạch lạc hơn.