Parsing PEs - Part 01

This project will allow us to explore the PE file format in great detail. In addition, we’ll also be discussing process internals, advanced assembly language topics and dynamic API resolution. This will be the part 1 of a multi-part project.

The goal of part 1 is to locate the base address in a **32-bit process** of **NTDLL**. NTDLL is a core Windows library and is present in all processes. In part 2 we will then parse the PE file format to locate the export table. Finally, in part 3 will be dynamically locate function addresses for a specific API.

This project will use the Windows Operating System exclusively. All code, demos and instructions are expected to be used within a Windows OS. Ideally you will use a Windows 7 machine. However, a more recent version of Windows (such as Windows 10) should suffice. Please note that structures, process information may vary.

These projects will require you to perform your own research and ask questions if you get stuck. I will NOT provide demonstrations on every aspect of the project - start earlier and don’t hesitate to reach out for help whenever you need it.

**Part 1**

The goal of part 1 is to locate the base address (i.e. image base) of NTDLL. To do this, we will utilize the FS register and a structure called the Process Environment Block (PEB). You can find more information about the PEB structure on MSDN: <https://msdn.microsoft.com/en-us/library/windows/desktop/aa813706(v=vs.85).aspx>.

Use inline assembly or straight assembly (demod in class), load the value of NTDLL base into memory or a register, and print the value as proof that you were successful. For example:



**Deliverables**

Turn in your source code along with a PDF or Word document containing screenshot(s) of successful execution of your program.