**Cách kiểm tra một file thuộc ngôn ngữ nào & công cụ dịch ngược phù hợp**

Nếu bạn có một tệp thực thi (**binary, ELF, PE, Mach-O**) hoặc mã nguồn (**Python, Java, C, Go, Rust,...**), bạn có thể sử dụng các công cụ sau để xác định **ngôn ngữ lập trình** và chọn công cụ **dịch ngược (decompile) hoặc biên dịch lại (compile).**

**1️⃣ Kiểm tra loại file**

Chạy lệnh:

bash

Sao chépChỉnh sửa

file crackme100

✔ Nếu kết quả có:

* **ELF 64-bit** → Code viết bằng **C, C++, Rust, Go** trên Linux.
* **PE32 executable** → Code viết bằng **C, C++, C#, VB.NET** trên Windows.
* **Python script** → Python script, ASCII text → Code viết bằng **Python**.
* **Java class** → Java class file, version 52.0 → Code viết bằng **Java/Kotlin**.

**2️⃣ Kiểm tra các chuỗi đặc trưng của ngôn ngữ**

Chạy:

bash

Sao chépChỉnh sửa

strings crackme100 | less

✔ Nếu có chuỗi **python**, **Py\_Initialize** → Code viết bằng **Python**.  
✔ Nếu có **JVM, Java** → Code viết bằng **Java/Kotlin**.  
✔ Nếu có **\_Z (mangling name)** → Code viết bằng **C++**.  
✔ Nếu có **Go build** → Code viết bằng **Go**.

**3️⃣ Kiểm tra thư viện (library)**

Chạy:

bash

Sao chépChỉnh sửa

ldd crackme100 # Linux (ELF)

bash

Sao chépChỉnh sửa

objdump -p crackme100 | grep "DLL" # Windows (PE)

✔ Nếu có **libc.so** → Code viết bằng **C/C++**.  
✔ Nếu có **.NET assemblies** → Code viết bằng **C# / VB.NET**.  
✔ Nếu có **libpython.so** → Code viết bằng **Python**.

**4️⃣ Công cụ dịch ngược (Decompile) theo từng ngôn ngữ**

| **Ngôn ngữ** | **Công cụ Decompile** | **Công cụ Compile** |
| --- | --- | --- |
| **C/C++** | Ghidra, IDA Pro, Radare2 | gcc, clang, g++ |
| **Python** | uncompyle6, decompyle3, pyinstxtractor | python, pyinstaller |
| **Java/Kotlin** | JD-GUI, Bytecode Viewer, cfr | javac, kotlinc |
| **C# / VB.NET** | dnSpy, ILSpy, dotPeek | csc, dotnet build |
| **Go** | Ghidra, IDA Pro, go-reverse | go build |
| **Rust** | Ghidra, Radare2 | rustc, cargo |

**5️⃣ Kiểm tra kiến trúc CPU**

Chạy:

bash

Sao chépChỉnh sửa

readelf -h crackme100 # Linux

✔ Nếu **x86-64** → Chương trình biên dịch cho CPU 64-bit.  
✔ Nếu **ARM** → Có thể là **Android** hoặc nhúng (**Embedded**).

**📌 Kết luận**

* 🔹 **Tệp viết bằng ngôn ngữ nào?** → Dùng file, strings, ldd để kiểm tra.
* 🔹 **Dịch ngược (Decompile)?** → Dùng công cụ phù hợp với từng ngôn ngữ (Python, C, Java, Go,...).
* 🔹 **Cần biên dịch lại?** → Dùng trình biên dịch (gcc, javac, csc, go build).

🚀 **Bạn có thể thử chạy các lệnh trên và gửi kết quả, tôi sẽ phân tích giúp bạn!**