

BÀI 5: THƯ VIỆN TĨNH & THƯ VIỆN ĐỘNG

Static vs Dynamic Libraries



Mục tiêu bài học

1. **Phân biệt:** Sự khác nhau giữa Static Lib (`.a`) và Dynamic Lib (`.so`).
2. **Kỹ năng:** Tự tạo và biên dịch thư viện từ source code C.
3. **Deploy:** Hiểu cơ chế `LD_LIBRARY_PATH` để chạy ứng dụng dùng thư viện động.
4. **Debug:** Sử dụng `ldd` và `nm` để kiểm tra liên kết.

1. Thư viện là gì? Tại sao cần?

- **Định nghĩa:** Là tập hợp các hàm (function) đã được biên dịch thành mã máy, có thể tái sử dụng cho nhiều chương trình khác nhau.
- **Lợi ích:**
 - Không phải viết lại code (Ví dụ: `printf` nằm trong thư viện `libc`).
 - Dễ bảo trì và nâng cấp.
 - Giảm thời gian biên dịch lại.

2. So sánh: Static (.a) vs Dynamic (.so)

Đặc điểm	Static Library (<code>libname.a</code>)	Dynamic/Shared Library (<code>libname.so</code>)
Cơ chế	Code thư viện được Copy nhét thẳng vào file chạy (<code>.exe</code>).	File chạy chỉ chứa tham chiếu . Code thư viện nằm riêng bên ngoài.
Dung lượng App	Lớn (Béo phì).	Nhỏ (Thon gọn).
Bộ nhớ RAM	Tốn RAM (Mỗi app load 1 bản copy riêng).	Tiết kiệm (Nhiều app dùng chung 1 bản trong RAM).
Cập nhật	Phải biên dịch lại toàn bộ App.	Chỉ cần thay file <code>.so</code> mới.
Đuôi file	Linux: <code>.a</code> / Windows: <code>.lib</code>	Linux: <code>.so</code> / Windows: <code>.dll</code>

3. Static Library (Thư viện tĩnh)

- **Cách tạo:** Dùng công cụ `ar` (Archiver) để gom các file `.o` lại.
- **Quy trình:**
 1. Compile: `gcc -c mymath.c -o mymath.o`
 2. Archive: `ar rcs libmymath.a mymath.o`
 3. Link: `gcc main.c -L. -lmymath -o app`

“**Ưu điểm:** App chạy độc lập, copy đi đâu cũng chạy được, không sợ thiếu thư viện.
Phù hợp với Bootloader hoặc các tool hệ thống nhỏ.”

4. Dynamic Library (Thư viện động)

- Cách tạo: Cần cờ **-fPIC** (Position Independent Code) và **-shared** .
 - Quy trình:
 1. Compile: **gcc -c -fPIC mymath.c -o mymath.o**
 2. Create SO: **gcc -shared -o libmymath.so mymath.o**
 3. Link: **gcc main.c -L. -lmymath -o app**
- “ **Lưu ý:** Khi chạy app, hệ điều hành sẽ tìm file **.so** . Nếu không thấy -> Lỗi **error while loading shared libraries** . ”



PHẦN THỰC HÀNH (LAB 05)

Tạo bộ thư viện tính toán (MyMathLib)

Bước 1: Chuẩn bị Code

Tạo 3 file: `calc.c` , `calc.h` , `main.c`

```
// calc.c
int add(int a, int b) {
    return a + b;
}

// calc.h
int add(int a, int b);

// main.c
#include <stdio.h>
#include "calc.h"

int main() {
    printf("Sum: %d\n", add(10, 20));
    return 0;
}
```

Bước 2: Tạo & Sử dụng Static Lib

```
# 1. Tạo file object (.o)
gcc -c calc.c -o calc.o

# 2. Đóng gói thành thư viện tĩnh (.a)
# ar = archive, r = replace, c = create, s = index
ar rcs libmycalc.a calc.o

# 3. Biên dịch App, liên kết với thư viện tĩnh
# -L. : Tìm thư viện ở thư mục hiện tại
# -lmycalc : Link với file libmycalc.a (bỏ đầu lib, bỏ đuôi .a)
gcc main.c -L. -lmycalc -o app_static

# 4. Kiểm tra & Chạy
ls -lh app_static # Dung lượng sẽ lớn hơn bình thường xíu
./app_static
```

Bước 3: Tạo & Sử dụng Dynamic Lib

```
# 1. Tạo file object với -fPIC (Mã độc lập vị trí)
```

```
gcc -c -fPIC calc.c -o calc.o
```

```
# 2. Tạo thư viện chia sẻ (.so)
```

```
gcc -shared -o libmycalc.so calc.o
```

```
# 3. Biên dịch App
```

```
gcc main.c -L. -lmycalc -o app_dynamic
```

```
# 4. Chạy thử
```

```
./app_dynamic
```

“ **STOP!** Bạn có gặp lỗi này không?

error while loading shared libraries: libmycalc.so: cannot open shared object file

”

Bước 4: Xử lý lỗi Runtime (Quan trọng)

Linux chỉ tìm thư viện ở `/lib` hoặc `/usr/lib`. Nó không tìm ở thư mục hiện tại.

Cách 1: Dùng biến môi trường (Tạm thời - Khuyên dùng khi dev)

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.  
./app_dynamic
```

Cách 2: Copy vào hệ thống (Vĩnh viễn)

```
sudo cp libmycalc.so /usr/lib/  
sudo ldconfig # Cập nhật cache thư viện  
./app_dynamic
```

5. Công cụ LDD (List Dynamic Dependencies)

Làm sao biết một chương trình cần những thư viện `.so` nào?

```
ldd app_dynamic
```

Kết quả:

```
linux-vdso.so.1 => (0x00007ff...)
libmycalc.so => not found <-- LỖI Ở ĐÂY
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6
```

“ Khi làm Embedded: Copy app xuống board mà không chạy được, hãy dùng `ldd` (trên board) để xem có thiếu thư viện nào không. ”



Bài tập về nhà

1. Viết **Makefile** để tự động hóa việc build cả bản Static và Dynamic cho bài Lab trên.
2. **Cross-compile:**
 - Tạo **libmycalc.so** cho ARM (**arm-linux-gnueabihf-gcc**).
 - Build **app_dynamic** cho ARM.
 - Copy cả 2 file xuống Board.
 - Set **LD_LIBRARY_PATH** trên Board và chạy thử.
3. Tìm hiểu thư mục **/etc/ld.so.conf.d/**.

Q & A

Chúc mừng bạn đã hoàn thành Giai đoạn 1!

Chuẩn bị tinh thần cho Giai đoạn 2: System Porting (Khó hơn x10 lần)