

BÀI 1: TỔNG QUAN & CÀI ĐẶT MÔI TRƯỜNG

Introduction to Embedded Linux & Environment Setup

Mục tiêu bài học hôm nay

Sau buổi này, các bạn sẽ:

1. **Hiểu bản chất:** Embedded Linux khác gì với Linux trên Desktop và Vi điều khiển (MCU)?
2. **Nắm kiến trúc:** 4 thành phần cốt lõi của một hệ thống nhúng.
3. **Xây dựng môi trường:** Tự cài đặt được Ubuntu trên máy ảo và các công cụ cần thiết.
4. **Sinh tồn:** Sử dụng được 5 lệnh Linux quan trọng nhất.

1. Embedded Linux là gì?

"Là việc sử dụng hạt nhân Linux (Kernel) trong các hệ thống nhúng."

Đặc điểm nhận dạng:

- **Tài nguyên giới hạn:** RAM/Flash nhỏ (vài MB đến vài GB).
- **Chuyên biệt:** Chỉ làm một nhiệm vụ cụ thể (Camera, Router) thay vì đa năng như PC.
- **Tùy biến cao:** Chỉ cài những gì cần thiết (Stripped down OS).

So sánh: MCU vs Embedded Linux

Đặc điểm	Vi điều khiển (STM32, AVR)	Embedded Linux (RPi, BeagleBone)
CPU	MHz (Đơn nhân)	GHz (Đa nhân)
RAM	Kilobytes (KB)	Megabytes/Gigabytes (MB/GB)
OS	Bare-metal / RTOS	Linux Kernel (Đa nhiệm, Đa người dùng)
Ưu điểm	Real-time, Giá rẻ, Ít tốn điện	Xử lý ảnh, AI, Mạng, Database
Khởi động	Tức thì (ms)	Chậm (vài giây đến phút)

2. Kiến trúc hệ thống (Big Picture)

Một hệ thống Embedded Linux luôn gồm 4 phần:

1. **Bootloader (U-Boot):** Chạy đầu tiên, khởi tạo RAM, nạp Kernel.
2. **Kernel (Linux):** Quản lý phần cứng (CPU, Memory, Driver).
3. **Root Filesystem (RootFS):** Chứa thư viện, file cấu hình (`/etc` , `/bin`).
4. **User Application:** Chương trình của bạn chạy trên cùng.

3. Khái niệm Host vs Target

Trong lập trình nhúng, chúng ta không code trực tiếp trên thiết bị.

- **Host (Máy chủ):** Laptop mạnh mẽ của bạn (Ubuntu). Dùng để viết code, biên dịch.
- **Target (Máy đích):** Board mạch yếu (Raspberry Pi/BeagleBone). Dùng để chạy ứng dụng.
- **Cross-Compilation:** Dùng trình biên dịch trên Host (x86) để tạo ra mã máy cho Target (ARM).

Ví dụ: `arm-linux-gnueabihf-gcc` chạy trên Intel nhưng tạo file `.exe` cho chip ARM.

PHẦN THỰC HÀNH (LAB 01)

Cài đặt môi trường "Kiểm cơm"

Bước 1: Chuẩn bị máy ảo (Virtual Machine)

Tại sao dùng máy ảo? -> *An toàn, dễ snapshot, chuẩn môi trường.*

1. Tải & Cài đặt **VMware Workstation Player** (hoặc VirtualBox).
2. Tải file ISO **Ubuntu 20.04 LTS** hoặc **22.04 LTS**.
3. Tạo máy ảo mới:
 - **RAM:** Tối thiểu 4GB (Khuyến nghị 8GB).
 - **Disk:** 40GB - 50GB.
 - **Network:** NAT (Để có internet tải gói).

Bước 2: Cài đặt các gói cần thiết

Mở Terminal (`Ctrl+Alt+T`) và chạy các lệnh sau:

1. Cập nhật danh sách gói

```
sudo apt update
```

```
sudo apt upgrade
```

2. Cài đặt công cụ lập trình cơ bản

```
sudo apt install build-essential git vim net-tools
```

3. Cài đặt thư viện cho build kernel

```
sudo apt install libncurses5-dev libssl-dev bison flex
```

Bước 3: Làm quen dòng lệnh (Survival Commands)

Hãy gõ thử và xem kết quả:

- `pwd` : Tôi đang đứng ở đâu?
- `ls -la` : Liệt kê tất cả file (kể cả file ẩn).
- `cd Documents` : Đi vào thư mục Documents.
- `cd ..` : Lùi ra ngoài một cấp.
- `mkdir Training_Linux` : Tạo thư mục mới.
- `rm -rf Training_Linux` : Xóa thư mục (Cẩn thận!).

Bước 4: Demo: Native GCC vs Cross-GCC

Tạo file `hello.c` :

```
#include <stdio.h>
int main() {
    printf("Hello World\n");
    return 0;
}
```

Thử nghiệm:

1. Biên dịch bằng `gcc` : Chạy được trên PC.

```
gcc hello.c -o hello_x86
```

2. Biên dịch bằng `arm-linux-gnueabi-gcc` :

```
arm-linux-gnueabi-gcc hello.c -o hello_arm
```

- Chạy trên PC -> Báo lỗi `Exec format error` (Đúng!).

! Các lỗi thường gặp khi mới bắt đầu

- Quên `sudo` : Khi cài phần mềm hoặc sửa file hệ thống -> Lỗi "Permission denied".
- Gõ sai lệnh: Linux phân biệt hoa thường (`Desktop` khác `desktop`).
- Không có mạng: Máy ảo chưa để chế độ NAT, không `apt install` được.
- Copy/Paste không được: Cần cài `open-vm-tools-desktop` để copy từ Windows vào Ubuntu.



Bài tập về nhà

1. Hoàn thiện việc cài đặt máy ảo Ubuntu.
2. Tạo cấu trúc thư mục cho khóa học:

```
~/Linux_Training/  
├── slides/  
├── labs/  
└── projects/
```

3. Tìm hiểu lệnh `chmod` để cấp quyền chạy cho file script.
4. **(Nâng cao):** Thử viết một shell script `setup.sh` để tự động chạy các lệnh cài đặt ở Bước 2.