BÀI 10: BIÊN DỊCH KERNEL & DEBUGGING

Kernel Compilation & Handling Panic

🎯 Mục tiêu bài học

1. **Biên dịch:** Tạo ra file `zImage` và `modules` từ source code.
2. **Triển khai:** Cách copy Kernel và Modules vào đúng chỗ trên thẻ nhớ.
3. **Boot:** Cấu hình U-Boot để load Kernel mới.
4. **Debug:** Phân tích lỗi kinh điển "Kernel Panic - not syncing: VFS".

1. Các thành phần đầu ra (Output Artifacts)

Sau khi build kernel, ta thu được:

1. `vmlinux` : File kernel dạng ELF (Rất lớn, dùng để debug).
2. `zImage` : Kernel đã nén (Compressed) -> Dùng để boot trên ARM 32-bit.
3. `Image` : Kernel không nén -> Dùng trên ARM 64-bit.
4. `uImage` : `zImage` + Header của U-Boot (Dùng cho các U-Boot đời cũ).
5. `*.dtb` : Device Tree Blobs.
6. `*.ko` : Kernel Modules (Driver biên dịch dạng Module).

2. Quy trình biên dịch đầy đủ

```
# 1. Build Kernel Image
make -j4 zImage

# 2. Build Device Tree
make dtbs

# 3. Build Modules (Driver)
make -j4 modules

# 4. Install Modules (Gom các file .ko vào thư mục)
# INSTALL_MOD_PATH là nơi chứa folder lib/modules/ tạm thời
make modules_install INSTALL_MOD_PATH=./rootfs_tmp
```

3. Lỗi "Kernel Panic - VFS"

Đây là lỗi phổ biến nhất khi mới học (Boot thất bại).

Thông báo:

```
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(xxx)
```

Nguyên nhân:

1. Kernel đã chạy xong, nhưng không tìm thấy phân vùng Root Filesystem.
2. Sai tham số `bootargs` trong U-Boot (Ví dụ: `root=/dev/mmcblk0p2` nhưng thẻ nhớ cắm vào lại là `mmcblk1`).
3. Thiếu driver cho thẻ nhớ (MMC) hoặc định dạng file (EXT4) trong Kernel.

🛠️ PHẦN THỰC HÀNH (LAB 10)

Build, Deploy & Boot Kernel

Bước 1: Build

Thực hiện các lệnh build ở phần Lý thuyết (trong folder source kernel đã config ở Bài 8).

Bước 2: Copy vào thẻ nhớ

Giả sử thẻ nhớ đang mount tại `/media/user/BOOT` và `/media/user/ROOTFS`.

1. **Copy Kernel & DTB vào phân vùng BOOT:**

```
cp arch/arm/boot/zImage /media/user/BOOT/
cp arch/arm/boot/dts/am335x-boneblack.dtb /media/user/BOOT/
```

2. **Copy Modules vào phân vùng ROOTFS:**

```
# Copy thư mục lib/modules vừa tạo ở Bước 1 vào thẻ nhớ
sudo cp -r ./rootfs_tmp/lib/ /media/user/ROOTFS/
```

Bước 3: Cấu hình U-Boot (Trên Board)

```
# Reset biến môi trường về mặc định
env default -f -a

# Thiết lập bootargs (Quan trọng!)
# console: In log ra UART
# root: Nơi chứa RootFS (Phân vùng 2 thẻ nhớ)
# rw: Quyền đọc ghi
setenv bootargs console=ttyO0,115200 root=/dev/mmcblk0p2 rw rootwait

# Lệnh boot: Load Kernel -> Load DTB -> Boot
setenv bootcmd 'fatload mmc 0 0x82000000 zImage; fatload mmc 0 0x88000000 am335x-boneblack.dtb; bootz 0x82000000 - 0x88000000'

# Lưu lại và Reset
saveenv
reset
```

Bước 4: Quan sát Log

- Nếu thấy dòng: `Uncompressing Linux... done, booting the kernel.` -> Chúc mừng! U-Boot đã nạp thành công Kernel.
- Nếu sau đó thấy log chạy rất nhiều và dừng lại ở `Kernel Panic` -> Bình thường, vì ta **chưa tạo Root Filesystem** (Bài sau sẽ làm).

Q & A

Hẹn gặp lại ở Bài 11: Root Filesystem!