

Embedded System Software

부트로더와 커널

Dept. of Computer Science and Engineering
Sogang University, Seoul, KOREA

목차

- ▶ **bootloader**
- ▶ **kernel image**

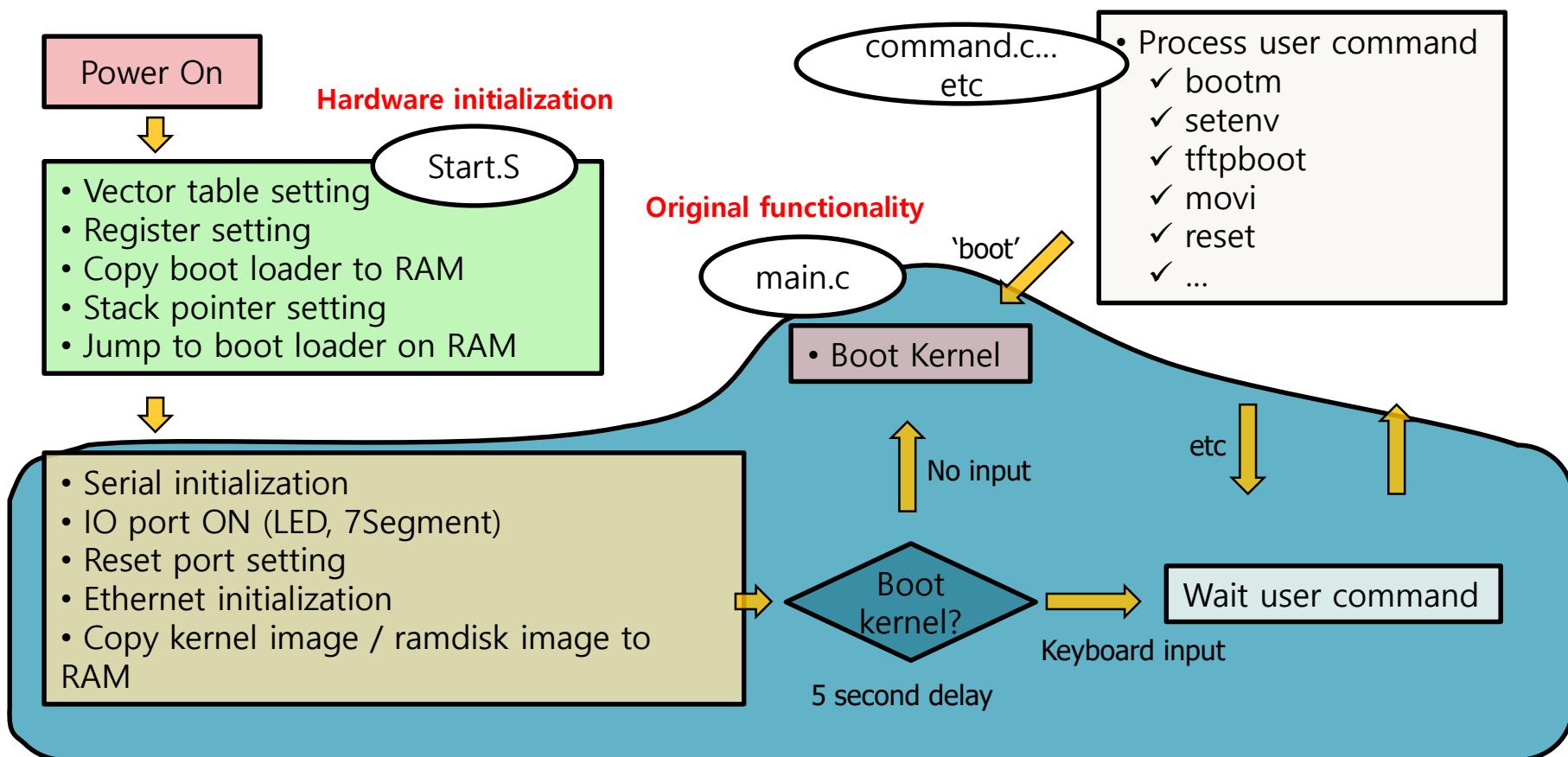


부트로더

- ▶ 전원이 들어오면, 사용이 가능한 하드웨어를 초기화
- ▶ 커널을 메모리로 올려서 리눅스가 부팅될 수 있도록 해주는 역할



Boot Loader in Achro-imx

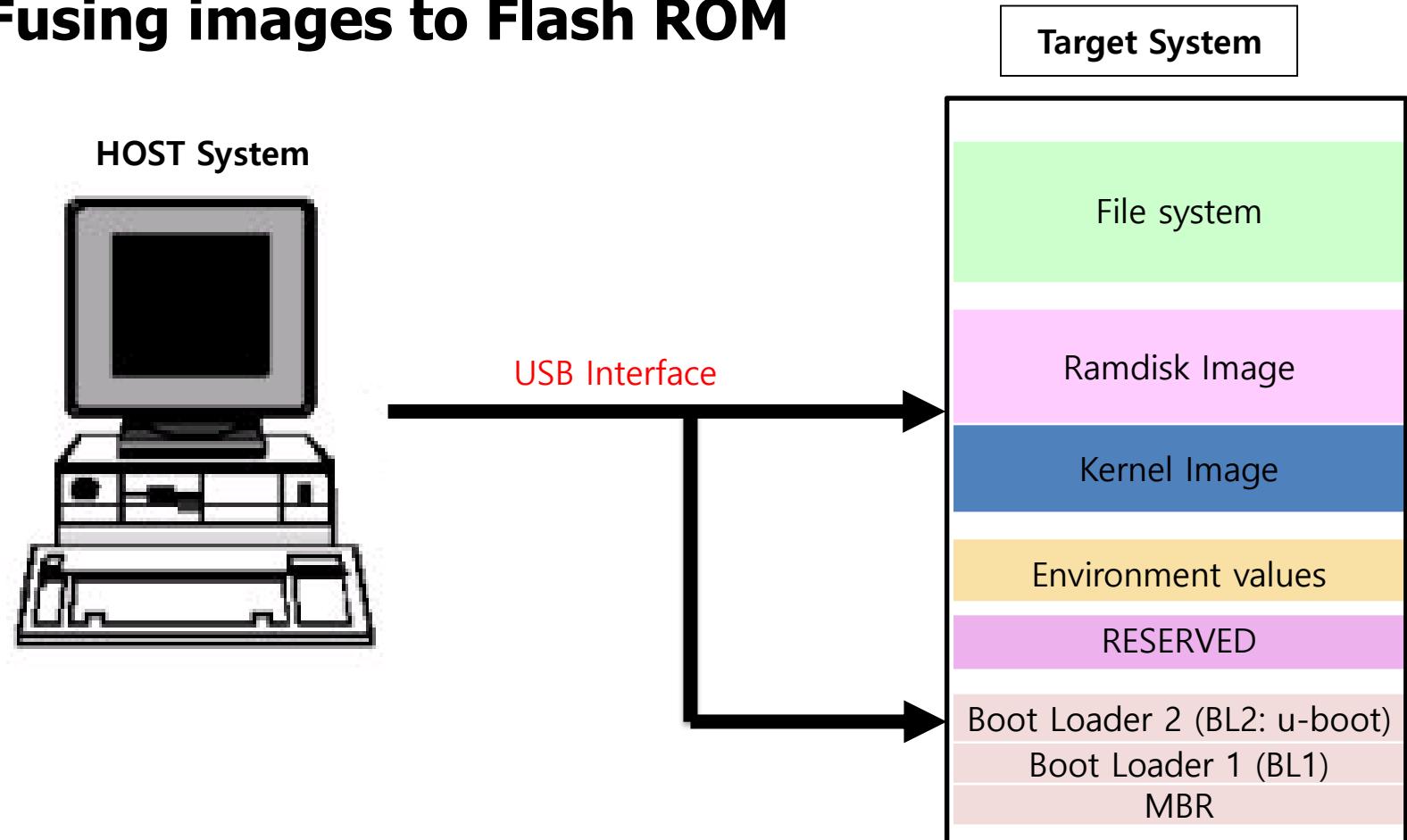


커널의 기능

- ▶ 시스템 자원을 관리해주는 기능
 - 프로세스 관리(Process Management)
 - 메모리 관리(Memory Management)
 - 파일 시스템 관리(File System Management)
 - 디바이스 관리(Device Management)
 - 네트워크 관리(Network Management)

Fusing Images to Flash ROM

▶ Fusing images to Flash ROM



Java library 설치

▶ 필요 패키지 설치(txt 파일 참고해도 됨)

- sudo apt-get install git gnupg flex bison gperf build-essential
- sudo apt-get install zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev
- sudo apt-get install libx11-dev:i386 libreadline6-dev:i386
- sudo apt-get install libgl1-mesa-dev g++-multilib mingw-w64 tofrodos u-boot-tools
- sudo apt-get install python-markdown libxml2-utils xsltproc zlib1g-dev:i386
- sudo apt-get install uuid uuid-dev zlib1g-dev liblz4-dev liblzo2-dev
- sudo apt-get install git-core libncurses5-dev libx11-dev



Java library 설치

▶ 필요 패키지 설치2(txt 파일 참고해도 됨)

- sudo apt-get install gcc-multilib libc6-i386 libc6-dev-i386
- sudo apt-get install lib32z-dev lib32ncurses5-dev
- sudo apt-get install libstdc++6:i386 lib32z1 lib32ncurses5



Java library 설치

Oracle JDK

- **sh jdk-6u45-linux-x64.bin**
- **mkdir /usr/lib/jvm**
- **mv jdk1.6.0_45 /usr/lib/jvm/java-6-oracle**
- **vi /root/.bashrc**

```
#fi

#cross compiler
export CROSS_COMPILE=arm-none-linux-gnueabi-
export PATH=/opt/toolchains/arm-2014.05/bin:$PATH:/usr/lib/jvm/java-6-oracle/bin

export ARCH=arm

#JAVA JDK
export JAVA_HOME=/usr/lib/jvm/java-6-oracle/jre/bin/java
export ANDROID_JAVA_HOME=/usr/lib/jvm/java-6-oracle
```

- **source /root/.bashrc**
- **java -version**

gcc, g++ downgrade

- ▶ 안드로이드 버전이 옛날 버전이라 현재 컴파일러 버전과 **dependency** 문제가 발생(4.8 버전 사용 필요)
 - sudo update-alternatives --remove-all gcc
 - sudo update-alternatives --remove-all g++
 - sudo apt-get install gcc-4.8 g++-4.8
 - sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 50 --slave /usr/bin/g++ g++ /usr/bin/g++-4.8
 - sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-5 10 --slave /usr/bin/g++ g++ /usr/bin/g++-5
- ▶ 버전 확인
 - gcc --version
- ▶ 버전 전환 하고 싶은 경우
 - sudo update-alternatives --config gcc
 - 선택



안드로이드 파일시스템 컴파일

- ▶ 배포한 앱축파일은 이미 **android**를 컴파일한 버전임
 - mkdir /work/
 - tar -zxvf android.tar.gz –C /work/
 - 이후 kernel 컴파일 후 bootimage 생성 때 사용
- ▶ **Compile(실습 x – 시간이 너무 오래 걸려서 생략하기로 함.)**
 - tar -zxvf android.tar.gz –C /work/
 - make clean(처음할 때만) <- 실습 때 입력하면 집에 못 갑니다
 - ./build_android.sh
 - 한번만 (build 오래걸림)



부트로더 컴파일

◆ **Compile**

- tar xvf u-boot.tar.bz2 –C /work/
- cd /work/u-boot/
- make distclean
- make clean
- make achroimx_android_sd_config
- make -j[# of core]

- 보충 : make achroimx_linux_sd_config 대신
make achroimx_android_sd_config로 하면
fastboot 사용 가능하게 config할 수 있음

부트로더 기록 (실습은 x)

▶ 부트로더 기록

- Micro SD card를 PC에 연결한다.
- Linux에서 dmesg로 Micro SD Partition 확인(**sdb,sdc...**)
- 파티션을 나눈다. u-boot source에서 ./part_sd.sh **sdb** 실행
- u-boot.bin 파일이 있는 곳에서 아래 명령어로 sd에 Write
- dd if=u-boot.bin of=/dev/**sdb** bs=512 seek=2 skip=2 conv=fsync
- Write가 되었으면 sd카드를 Target Board에 T-flash에 넣는다.
- Linux에서 미니컴 실행
- PC와 보드를 UART to Serial Cable로 연결
- 개발 보드의 전원을 켠다.

커널 컴파일

▶ **Compile**

- **tar -zxvf kernel.tar.gz –C /work/**
- **cd /work/achroimx_kernel/**
- **make clean**(처음할 때만)
- **export ARCH=arm**
- **make achroimx_defconfig**
- **make -j[# of core]**

- **mkimage**관련 에러시
cp /work/u-boot/tools/mkimage /usr/bin



커널 기록 (usb 이용)

▶ Bootimage 생성 (kernel compile 후)

- cd /work/android
- vim make_bootimage.sh에서 KERNEL_ZIMAGE경로 설정

```
#KERNEL_ZIMAGE="../kernel/arch/arm/boot/zImage"
KERNEL_ZIMAGE="../arch/omapx_kernel/arch/arm/boot/zImage"
#KERNEL_CMDLINE="console=ttyMxc3,115200 init=/init video=mxcfb0:dev=
xcfb2:off fbmem=10M vmalloc=400M androidboot.console=ttyMxc3 android
```

- ./make_bootimage.sh

커널 기록 (usb 이용)

▶ Fastboot 방법 (바로 flash에 기록)

- **# apt-get install android-tools-fastboot**
- > u-boot모드 진입 (device)
- > fastboot (device)
- **# fastboot erase boot**
- **# fastboot flash boot boot.img**
- **# fastboot reboot**

