

- Subject : The Flow of Linux Porting
- Created date : 2017. 05. 25
- Recently modified date : 2017. 05. 30
- Creator : Choi Jun-Ho
- Contact : peluza6332@gmail.com
- Worked dates to make working of subject : 2017. 05. 22 ~ 2017. 05. 23
- Documentation status : **finished**
- Summary :
개발 보드에 리눅스 포팅하는 것에 대한 이해와 그 방법
- Contents
 - Understand of the Linux porting
 - The things of needs to Linux porting
 - Abstract Procedure

- Understanding of the Linux porting

하드웨어에 리눅스 포팅한다는 것은 그 하드웨어에 리눅스 운영체제를 설치한다는 것을 말한다.

이렇게 하는 이유는 리눅스 운영체제를 설치하여 그 하드웨어와 그것에 연결된 모든 장치들을 운영체제를 통해 제어할 수 있게 하기 위함이다.

- The things of needs to Linux porting

하드웨어에 리눅스 포팅을 위해서는 그 하드웨어가 켜질 때 해당 운영체제가 주 메모리에 적재되고 실행되어야 한다.

하드웨어에 리눅스 포팅을 위해서는 단순 리눅스만 필요한 것이 아니라 위에서 언급했듯이 하드웨어가 켜지면서 스스로 리눅스를 올려야 하기 때문에 그 일련의 처리에 대한 제어도 할 수 있어야 한다.

따라서 하드웨어에 리눅스를 포팅하기 위해서는 하드웨어가 켜졌을 때 스스로 운영체제를 메모리에 복사하고 실행할 그것이 있어야 하며 실행할 리눅스와 마지막으로 운영체제가 실행되면서 각종 저장 파일들을 운용할 파일 시스템이 필요하다.

이때 하드웨어가 켜졌을 때 알아서 실행시킬 리눅스와 그 리눅스가 사용할 파일 시스템을 가져오게 하는 것을 bootloader라고 하며 이것은 리눅스, 파일 시스템과 더불어 필수적인 것이다.

그리고 이때 사용할 리눅스 자체를 리눅스 커널 이미지(linux kernel image 또는 kernal binary)라고 부르며 파일 시스템은 그대로 filesystem라고 부른다.

이들은 각각 파일화 될 때 u-boot.bin, zImage(또는 uImage), ramdisk 형태를 띈다.

여기에 리눅스 커널이 각종 장치들을 좀 더 효율적이고 체계적으로 관리하기 위해서 리눅스 커널 3.0부터 도입한 디바이스 트리(.dts, .dtb)가 리눅스 운영체제 부팅 시 필요하다.

이외에도 u-boot, kernel binary(zImage)는 크로스 컴파일러를 통해 해당 하드웨어에 맞는 실행 파일들로 컴파일된다.

현재 운영체제와 타겟 하드웨어의 컴파일 방식이 다를 수 있으므로 맞는 크로스 컴파일러를 정해 컴파일 해주어야 한다.

따라서 리눅스 포팅을 위해 필요한 것들을 정리하자면

- Bootloader(u-boot)
- Linux kernel image(zImage)
- Linux kernel device tree blob(devicetree.dtb)
- filesystem(ramdisk.gz)

이며 파일 형태는 필요에 따라 변경될 수 있다.

- Abstract Procedure

-> 작업용 컴퓨터 리눅스 운영체제 사용

하드웨어의 리눅스 운영체제 부팅에 사용될 파일을 만드는 작업을 할 작업 컴퓨터에 리눅스 운영체제 설치.

-> 크로스 컴파일러, 환경변수 설정

리눅스 운영체제를 포팅할 하드웨어(개발 보드)의 사용 환경(ARM, etc, ...)에 따라 적절한 크로스 컴파일러와 그 외 환경변수 설정
(서로의 컴파일 환경이 다르기 때문)

-> 부트로더 빌드

하드웨어에 전원이 들어오면 물리 메모리의 처음에 가장 먼저 로드될 파일.
이 파일이 로드되어 커널과 파일 시스템 등을 적재한다.

-> 커널 빌드

실제 하드웨어에 사용될, 옵션을 조정한 커널을 만든다.

-> 디바이스 트리 빌드

커널 3.x 이상 버전부터 도입된 디바이스 트리를 빌드한다.

-> 비트스트림 파일 생성

타겟 하드웨어에서 사용할 비트스트림을 생성한다.

-> 부트 이미지 생성

비트스트림, 부트로더, 커널, 디바이스 트리를 사용하여 부트 이미지를 생성한다.

-> 파일 시스템 빌드

포팅한 리눅스 운영체제가 사용할 파일 시스템 빌드

이때 파일 시스템이 압축되어 질 형식 또한 사용자가 정하는 것이다.

-> 타겟 하드웨어의 비휘발성 메모리에 부트 이미지 파일과 파일 시스템 적재

부트 이미지의 순차적인 실행을 통해 리눅스 커널이 메모리와 파일 시스템이 적재된다.

