

Booting, GRUB

—

@nenw

at 01.05, 2021 Winter Wheel Seminar

부팅 (Roughly)

부팅 또는 부팅업은 컴퓨터에서 사용자가 운영 체제를 시동할 때 운영 체제를 시작하는 부트스트래핑 과정이다.

- Wikipedia, 부팅

부팅 (Roughly)

전원버튼을 누른 후부터 운영체제가 시작돼서 사용할 수 있을 때가 되기 전까지의 과정

Firmware

Bootloader

Kernel

Initramfs

Init Process

간략하게 나타낸 리눅스의 부팅과정

Firmware

BIOS

Basic Input-Output System

예전에 나온 펌웨어 규격

MBR을 이용한 부팅 (~ 2TiB) (GPT로도..... 잘 하면 어떻게든 가능하긴 함)

UEFI

Unified Extensible Firmware Interface

최근에 나온 펌웨어 규격

리눅스에서 efibootmgr 명령어로 부팅설정을 관리할 수 있음

GPT를 이용한 부팅 (MBR로도 가능하긴 함)

Secure Boot 등 많은 차이점이 존재 (보안이 향상되는게 언제나 좋은 것만은 아님)

Firmware / Boot Sequence

POST (Power-On Self Test)

CPU나 기타 장치 등이 제대로 작동하나 검증
비프음 등으로 알림

하드웨어 초기화

부팅 디스크 > MBR 또는 ESP > 부트로더 로딩

Firmware / MBR, GPT

MBR (Master Boot Record)

하드디스크의 첫 섹터

최대 주 파티션 4개 / 2TB까지만 가능함

부팅에 필요한 코드, 파티션 테이블 등으로 구성

(최대 440 바이트라 사실상 부트로더로 넘겨주기만 함)

GPT (GUID Partition Table)

최대 주 파티션 128개 / 8ZB까지 가능함 (보통 사람들에게 있어선 사실상 무제한)

ESP (EFI System Partition)

Protective MBR

Firmware / GPT

ESP (EFI System Partition)

파티션 중 하나이지만, 부팅에 사용되는 파티션이라는 점이 중요

보통은 첫번째에 위치하나, 꼭 그럴 필요는 없음

디렉토리 구분자가 '/'가 아니라 '\'임

최소 32MiB, 그러나 그냥 넉넉하게 100MiB 정도 할당하면 편함

(특히 /boot에 마운트할 경우, 부트 이미지로 꽂아서 고통받은 경험 때문에 500MiB까지도 할당해본 적 있음)

주로 /boot, /efi, /boot/efi 등에 마운트됨

부트로더들 (*.efi 파일들)이 있음

(/boot 에 마운트해서 Initrd, Kernel 등을 넣어둘 수도 있음, systemd-boot 등을 사용하는 경우, 이것이 필수)

FAT32 포맷, 파티션 타입 "EF 00 (EFI System)"

Bootloader / GRUB

GRand Unified Bootloader

역할

리눅스나 기타 OS의 부팅을 시작하기 위한 용도

GRUB 외에도 rEFInd 등의 많은 부트로더가 있으나, 보통 GRUB을 많이 씀

설정 파일 위치

/etc/default/grub

/etc/grub.d/

/boot/grub/grub.cfg (편집 비권장, grub-mkconfig로 자동생성됨)

주의! 잘못 건드렸다가는 부팅용 USB 없이는 오도가도 못하는 상황이 올 수 있습니다.

팁! 부팅 시에 GRUB이 안 뜨고 바로 리눅스로 넘어갈 때 Shift키를 누르고 있으면 GRUB이 뜹니다.

Bootloader / GRUB

설정 파일 구조

```
set name='value'  
insmod  
linux  
initrd  
echo  
search  
chainloader  
menuentry "My Linux" { }
```

Bootloader / GRUB

설정 파일 구조

```
set name='value'
```

환경변수 설정

root: 기본적으로 파일을 어떤 디스크에서 불러올지

timeout: 몇 초 지나면 알아서 부팅할지

등등...

[\(참조 링크\)](#)

Bootloader / GRUB

설정 파일 구조

insmod (모듈 명)

모듈을 로딩

part_gpt: GPT 파티션 용 모듈

chain: chainloader (이후 설명)를 위해 필요한 모듈

등등...

Bootloader / GRUB

설정 파일 구조

linux (경로) (플래그)

리눅스 커널을 로딩

플래그: 부팅의 행동을 변화

single: 싱글유저 모드로 부팅

(안전모드 (runlevel 1) / 비밀번호 잊어버렸을 때 ㅎㅎ)

fastboot: 빠르게 부팅

(부팅 시에 시행하는 파일시스템 검사 등이 스킵됨, Ctrl + Alt + F1 to TTY)

rw: 부팅 시에 장치를 읽기/쓰기 가능하게 로딩

init=(값): 시작할 init proces를 바꿈

selinux=(값): 부팅 시에 SELinux를 켜고 끄

nomodeset: Kernel mode setting을 끄

(GUI 환경에서 화면이 안 나오거나 할 때 시도)

등등등.....

Bootloader / GRUB

설정 파일 구조

initrd (경로)

initrd의 경로를 설정 (후에 설명)

Bootloader / GRUB

설정 파일 구조

echo (텍스트)

화면에 텍스트를 출력

Bootloader / GRUB

설정 파일 구조

search

원하는 파일시스템을 찾음 (hd1,gpt3)과 같은 형식

```
search --no-floppy --fs-uuid --set=root 1234-5678
```

→ UUID가 1234-5678인 파일 시스템을 찾아 root 변수에 저장

(ls -al /dev/disk/by-uuid)

```
search --no-floppy --file --set=root /asdf/zxcv
```

→ /asdf/zxcv라는 이름의 파일을 가진 파일 시스템을 찾아 root 변수에 저장

Bootloader / GRUB

설정 파일 구조

chainloader (efi 파일 경로)

다른 부트로더를 로딩

Bootloader / GRUB

설정 파일 구조

```
menuentry "My Aweesome Linux" {  
    insmod part_gpt  
    search --no-floppy --fs-uuid --set=root 1234-5678  
    linux /vmlinuz-linux rw quiet  
    initrd /initramfs-linux.img  
}
```

해당하는 이름의 부팅 메뉴 생성

Bootloader / Bootable USB?

말 나온 김에...

정의: 원하는 OS의 iso 파일을 다운받아 USB에 구워 부팅할 수 있게 만든 것
(스팍스가 운영하는 카이스트 미러에서 빠르게 받을 수 있음)

용도: OS를 설치하거나, 기존 OS가 동작하지 않을 때

Bootloader / Bootable USB?

사용법:

1. 목적에 맞는 iso파일을 받음

- * 사용중인 아키텍처 (x86?)
- * 원하는 OS (ArchLinux?)
- * 안에 들어있어야 하는 것 (설치용? 라이브 부팅용?)

최근에는 같이 담겨있는 경우가 많음

아치리눅스는 라이브 부팅용만 던져주고 명령어로 알아서 설치하라 함...

2. USB에 구움

- * Rufus (Windows), dd (Linux)

3. USB를 꽂고 부팅

4. 펌웨어 설정에서 USB의 부팅 우선순위를 높이기 (또는 부팅할 매체 선택에서 USB 선택하기)

Firmware

Bootloader

Kernel

Initramfs

Init Process

간략하게 나타낸 리눅스의 부팅과정

Initramfs

배경

커널에서 디스크에 바로 접근할 수 없는 경우가 있음

ex) 커널에 포함돼있지 않은 추가적인 드라이버가 있어야 로딩을 할 수 있음

그런데 부팅의 초기 단계에서 그런 디스크를 써야할 일이 생길 수 있음

ex) init 프로세스가 암호화된 디스크 안에 있음

상황정리

부팅이 다 끝나면 그 디스크를 쓸 수 있음

그런데 부팅을 하기 위해 그 디스크를 써야함

= 디스크를 쓰기 위해 부팅을 하기 위해 디스크를 써야함....????

Initrd

해결법

1. 파일시스템을 마운트하기 위해 필요한 파일만 어디에 담아두기
2. 그 파일을 메모리에 올리고 이걸 디스크처럼 인식시키기
3. 그 디스크에서 부팅을 시작

initrd (Initial Root Disk)

파일시스템을 마운트하는데 필요한 것들을 initrd에 담아둠

이 디스크에서 필요한 스크립트를 돌림

이 후에 실제 루트 파일 시스템으로 전환해서 실제 루트의 init 프로세스로 전환

Initramfs

ramdisk

하나의 디스크 장치로 인식하게 해서 initrd를 구현

initramfs (Initial Ram Filesystem)

tmpfs라는 파일시스템을 마운트하고 initramfs의 내용물을 거기에 추출해서 initrd를 구현

배경 (ramdisk 는...)

디스크 장치로 인식돼서 메모리에 계속 상주하게 됨
크기가 작으면, 필요한 스크립트들이 다 안 들어감
크기가 크면, 쓸데없이 메모리를 많이 잡아먹음

Init Process

처음 시작되는 프로세스, 모든 프로세스의 어머니

`/sbin/init`

프로세스 ID 1번 프로세스

이 후 `init`에 의해 다양한 서비스나 스크립트 등이 실행되며 부팅이 완료
runlevel이 높아지는 방향 (`/etc/rc(n).d/`)

init.d

`init` 프로세스에 의해 시작돼야 할 여러가지 서비스와 스크립트들이 위치
(서비스를 시작/종료하는 스크립트를 만들어 관리)

- `/etc/init.d/my-service start`

Init Process / Alternatives

systemd

최근에 init 프로세스를 대체함 (/sbin/init 이 systemd로 링크돼있음)

병렬적 실행 등 많은 개선사항

최근 널리 쓰이는 중

systemd-networkd 등 많은 컴포넌트가 있으며, 많은 부분을 커버함

서비스를 Unit 파일로 관리 (/usr/lib/systemd, /etc/system)

- `systemctl start my-service`

OpenRC

systemd는 유닉스 철학에도 맞지 않는다, 너무 혼자서 많은 걸 하려고 한다 등 의견

→ 젠투 개발자가 제작, 미니멀한 리눅스 배포판을 쓴다면 만나볼...수도?

- `rc-service my-service start`

References

ArchWiki

- systemd
- Arch boot process
- Partitioning
- Kernel parameters

Gentoo Wiki

- Initramfs
- OpenRC
- GRUB2

GNU Manual

- Grub

The Linux Kernel Documentation

- admin-guide/kernel-parameters

Wikipedia

- Power-on self-test
- EFI system partition

Zetawiki

- 리눅스 런레벨

이 외에도 많은 인터넷 글들

Thank You