

8장

리눅스 부팅 과정

전원 ON - 바이오스 - 부트 로더 - 커널 초기화 - systemd 서비스 - 로그인 프롬프트
출력

바이오스 : 하드웨어 검사 - 부팅장치 선택 - MBR 로드 - 부트로더 로드
하드웨어의 상태 확인
부팅 디스크의 앞 512B 로딩
로딩 된 MBR이 부트로더를 로딩

부트로더 : 어떤 운영체제 이용할건지 선택하는 단계 (까만화면에 카운트다운) : 리눅스
커널을 메모리에 로딩하는 역할 수행
우분투의 부트로더는 GRUB

커널 초기화 : 시스템에 연결된 장치 검사 후 프로세스와 스레드 생성
모두의 부모 init 프로세스 실행
커널 중요한 일 수행(스케줄링, 메모리 관리 등)

systemd 서비스 : 부팅 메시지 출력하면서 리눅스 동작

init은 전통적인 1번 프로세스이지만 systemd의 심볼릭 링크로 바뀜

systemd : 리눅스 시스템, 서비스의 관리자

장점

inetd와 호환성
셸과 독립적으로 부팅
마운트 제어?
서비스에 명령
안전한 종료

유닛

service : 시스템 서비스 유닛으로 데몬을 시작, 종료, 재시작, 로드시킴
target : 유닛을 그룹핑

런레벨(init 명령어) : 시스템 상태에 따라 셸을 실행하는데, 이에 필요한 구분자

run level	target	
0	poweroff.target	: 종료
1	rescue.target	: 응급 복구 모드
2~4	multi-user.target	: 다중 사용자 모드
5	graphical.target	: 그래픽얼 다중 사용자 모드

6 reboot.target : 재시작

init 1 = systemctl isolate runlevel1.target = systemctl isolate rescue

systemd 명령 : systemctl

systemctl : 동작중인 유닛 출력

systemctl start : 유닛 실행

systemctl status : 유닛 상태 확인

systemctl is-active : 실행중인지 확인

systemctl get-default : 현재 target과 런레벨 확인

systemctl set-default <name>.target : 런레벨 바꾸기

shutdown 옵션 [시간:분] 메시지

-r : 재부팅

-h : 종료

-k : 겁만줘

-c : 종료취소

데몬 프로세스 : 백그라운드에서 동작하는 프로세스

독자형 데몬 : 혼자 동작

슈퍼데몬 : 다른 데몬 관리

1번 데몬 = systemd

주요 데몬

atd : 예약한 명령 실행

cron : 예약한 명령 실행

dhcpcd : 동적 IP 주소 부여

httpd : 웹서비스

sshd : 원격 보안 접속

ftpd : 파일 송수신

9장

패키지 관리 : 리눅스판 파일 설치 - 간단단

패키지 카테고리(4개 -> 2개만)

main : 우분투에서 공식적으로 지원, 자유배포

restricted : 우분투에서 지원되나 완전 자유 ~~ x

아키텍처 = cpu명?

우분투 패키지 저장소 = /etc/apt/sources.list 파일에 저장

패키지 관리 : apt-cache, apt-get

의존성 : 부모자식 느낌으로 자식 실행 위해 부모 전부 필요 -> 자동화됨

apt-cache stats : apt 캐시 통계 정보

apt-cache pkgnames : 사용 가능한 패키지 이름 보기 - 이름만 먼저 땡겨와서 볼 수 있음

apt-cache search : 이름 검색

apt-cache show : 정보 검색

apt-cache showpkg : 의존성 검색

apt-get update : 이름들 끌어오기

apt-get install : 패키지 업그레이드 / 설치

apt-get remove : 패키지 삭제

apt-get purge : 설정파일 함께 삭제

apt-get remove --purge : 한꺼번에 실행

apt-get autoremove : 정리

apt-get clean : 공간 정리

apt-get download : 패키지 설치 말고 다운로드 (서버에서 배포 끝난 이후에도 쓰고싶을 때)

apt 명령어 = apt-cache + apt-get

dpkg = 날것의 apt 명령 : 의존성 고려 x, 일일이 부모 패키지 신경쓸 때 사용

-i : 설치

-l : 검색

snap : 용량 많아요 - 안씀

파일 아카이브/압축

아카이브 : 묶기

압축 : 용량줄이기

cvf : 묶어

tvf : 확인

xvf : 풀어

uvf : 추가

cvzf : 압축해서 묶어

xvzf : 다 풀어

tar cvf 이름 파일 위치 : 이름을 갖는 아카이브를 위치에 만들어

tar cvzf 이름 파일 위치 : 이름을 갖는 아카이브를 "압축해서" 위치에 만들어

gcc : c언어 컴파일

10장

사용자 계정 관련 파일

/etc/passwd

로그인ID:x:UID:GID:설명:홈디렉터리:로그인 셸 정보 담김

x자리가 비밀번호 자리 - 옛날엔 비밀번호가 8로 담김

UID : 0~999, 65534번은 시스템 사용자 UID, 일반 사용자는 1000번부터 할당(root = 0번)

/etc/shadow (권한 :640)

로그인ID : x : 최종변경일 : MIN : MAX : WARNING : INACTIVE : EXPIRE :

Flag

x : 암호화 된 암호

MIN : 최소 사용기간

MAX : 최대 사용기간

WARNING : 경고날

INACTIVE : 만료되도 며칠간 사용 o

EXPIRE : 만료날

Flag : 나중을 위한 빈칸

/etc/login.defs : 위 내용들 결정하는 파일

기본값

PASS_MAX_DAYS : 99999

UID_MIN , UID_MAX : 1000 ~ 60000

SYS_UID_MIN , SYS_UID_MAX : 100 ~ 999

/etc/group : 그룹 정보 파일

/etc/gshadow : 그룹 암호 파일

사용자 계정 관리 명령

useradd : 계정 생성

-u Uid : UID 지정

-o : UID 중복 허용

-g gid : 기본 그룹의 GID 지정

-d 디렉터리 명 : 디렉터리 지정

-s 셸 : 기본 셸 지정
-c 설명 : 설명 추가
-m : 홈디렉터리 자동 생성

ex) useradd -m -u 1004 -g 1007 -s /bin/bash user 100

/etc/skel 디렉터리 역할 : 사용자 계정의 홈 디렉터리에 공통으로 배포할 파일들 들어감

adduser : apt마냥 더 쉬운거

--uid UID : UID
--gid GID : 그룹
--home DIR : 어디 홈디렉터리
--shell shell : 어떤셸

ex) adduser --uid 10004 --gid 10004 --home /home/user10004 --shell
/bin/bash user10004

usermod : 계정 수정

-u : UID
-g : GID
-d : DIR
-s : SHELL
-c : cerMyeong

userdel 옵션 유저명 : 계정 삭제

-r : 홈디렉터리 같이 삭제

패스워드 에이징★★★★★★★★★★★★★

MIN : passwd -n NUM , chage -m
MAX : passwd -x NUM , chage -M
WARNING : passwd -w NUM , chage -W

ex) sudo chage -W 30 user1

계정 생성

useradd 옵션 유저명

sudo useradd -m -d /home/test01 -s /bin/bash -u 2100 -G 3 -c "test01
user" test01

sudo passwd test01

까지해야 로그인 가능

adduser 옵션 유저명

```
sudo adduser --uid 2200 --gid 4 -gecos "test02 user" --shell /bin/bash  
test02
```

하면 암호 쓰게하고 만들어짐

그룹 관리 명령

groupadd 옵션 그룹명 : 그룹 생성

-g : gid (없으면 마지막 다음번호 지정)

-o : 중복

ex) groupadd -g 1004 chunSa

addgroup

--gid

groupmod 옵션 그룹명

-g : gid 변경

-o : gid 중복 허용

-n : 그룹 이름 변경

groupdel : 그룹 삭제

gpasswd : 관리

-a : 계정 추가

-d : 계정 삭제

-r : 암호 삭제

사용자 정보 관리 명령

UID : 로그인 한 계정

EUID : 썩은 계정

su - (계정) : EUID 들어가기

who : UID 출력

whoami : EUID 출력

passwd

-l : 암호 잠그기

-u : 암호 잠금 해제

-d : 암호 삭제

chown 계정(:그룹) 파일/디렉 : 파일/디렉의 소유자/소유그룹 변경

chgrp 그룹 파일/디렉 : 그룹 변경

12장

inetd : 인터넷 슈퍼 데몬

텔넷 : 암호화 안됨

내꺼 텔넷 열기 : telnet 0 = telnet localhost = telnet [내 IP]

내 IP : ifconfig -> inet

ssh : 암호화 됨

port number

ftp : 21

ssh : 22

telnet : 23

mail : 25

서버가 살아있는지 확인 : telnet [ip] [port number]

ftp

put : ftp 연 자리에 있는 파일 업로드

get : ftp 연 자리에 파일 다운로드

mget : 여러개 다운로드

13장

데이터베이스

관련성 있는 데이터들을 중복성을 최소화해서 모아놓은 것

데이터 : 저장되는 값

테이블 : 데이터 모임

필드 : 테이블의 열(학번 쪽, 이름 쪽 등등)

레코드 : 테이블의 행(튜플, 같은놈 정보들)

키 : 구분 가능한 필드(학번, 주민등록번호 등등 유일한 정보 곳)

show databases : 목록출력

create database 이름 : 생성

drop database 이름 : 삭제

use 이름 : 데이터베이스 선택

create table 이름(속성 자료형, 속성 자료형 ,...)

자료형 종류 : varchar, char, int, float, date, time

explain 테이블명 : 테이블에 뭐있는지 목록으로 출력

select 필드명, 필드명, ... from 테이블명 where 조건 : 테이블에서 조건에 맞는 필드를
모조리 긁어오라

ex) select 계좌번호, 비밀번호, 예금주, 잔액 from kbBank where 잔액>=10000000;

kbBank table에서 잔액이 10000000 이상인 튜플의 계좌번호, 비밀번호, 예금주, 잔
액 정보를 불러오기

explain(필드정보), insert(튜플삽입), alter(변경), update(튜플 수정), delete(튜플 삭제)

ex)

explain st_info;

insert into st_info values(3748, "호빵맨", "HeroEngineering")

alter table st_info add age int : 테이블에 필드를 추가

update st_info set age=3500 where st_id=3748 : 학번이 3748인놈 age를 3500으로 "
업데이트"