

## Section 1. 리눅스의 개요

### 리눅스의 특징 및 장단점

#### 특징

- 오픈 소스 운영체제이다.
- 멀티유저(다중사용자), 멀티테스킹(다중작업) 운영체제이다.
- 다중스레드를 지원하는 네트워크 운영체제이다.
- 여러 종류의 파일시스템을 지원하는 운영체제이다.

#### 장단점

- 리눅스는 유닉스와 완벽하게 호환 가능하다.
- 리눅스는 PC용 운영체제보다 안정적이다.
- 하드웨어 기능을 효과적으로 사용한다.
- 리눅스는 오픈 소스 운영체제이다.
- 공개 운영체제이기 때문에 문제점 발생 시 기술지원을 받기 어렵다.
- 한글 지원이 미흡하다.

### 리눅스 디렉터리 종류와 특징

디렉터리	저장 내용
/	파일 시스템이 있는 최상위 디렉터리로 루트 디렉터리라고 함 모든 디렉터리의 출발점인 동시에 다른 시스템과의 연결점이 되는 디렉터리
/boot	부트 디렉터리로 부팅 시 커널 이미지의 부팅 정보 저장 파일
/proc	시스템 정보 디렉터리이며 커널 기능을 제어하는 역할 현재 실행되는 프로세스와 실제로 사용되는 장치, 하드웨어 정보 저장
/lib	공유 라이브러리 디렉터리 커널 모듈 파일들과 프로그램 실행을 지원해 주는 라이브러리 저장
/bin	기본적인 명령어가 저장된 디렉터리 root 사용자와 일반 사용자가 함께 사용할 수 있는 명령의 디렉터리
/dev	시스템 디바이스 파일들을 저장하는 디렉터리 하드디스크 장치 파일, CD-ROM 장치파일 같은 파일 저장
/etc	시스템 환경 설정 파일 저장 디렉터리
/root	시스템 관리자용 홈 디렉터리

/sbin	관리자용 시스템 표준 명령 및 시스템 관리와 관련된 실행 명령어 저장
/usr	사용자 디렉터리로 사용자 데이터나 애플리케이션 저장
/home	사용자 계정 디렉터리로 계정들의 홈 디렉터리가 위치 일반 사용자들이 로그인 시 처음으로 위치하게 되는 디렉터리
/var	가변 자료 저장 디렉터리로 로그 파일이나 메일 데이터 저장
/tmp	각종 프로그램이나 프로세스 작업을 할 때 임시로 생성되는 파일 저장 모든 사용자에게 대해서 읽기와 쓰기가 허용 시티키 비트 설정으로 파일의 소유자만이 자신의 소유 파일을 지울 수 있음
/mnt	파일 시스템을 일시적으로 마운트 할 때 사용
/lost+found	결함이 있는 파일에 대한 정보가 저장되는 디렉터리

## 리눅스 배포판

### 종류

- 슬랙웨어 리눅스 : 배포판 가운데 가장 먼저 대중화된 배포판으로 1992년 패트릭 볼커딩 에 의해 출시되었다.
- 데비안 : 데비안 프로젝트에서 만들어 배포하는 공개 운영체제로 GNU의 공식적인 후원 을 받고있는 유일한 배포판이다.
- 우분투 : 데비안GNU/리눅스에 기초한 운영체제이다.
- 레드햇 : 미국의 레드햇사가 개발하던 리눅스 배포판이다.
- RHEL : 레드햇이 개발하여 판매하고 있는 상용 리눅스 배포판이다.
- 페도라 : 리눅스 커널에 기반한 운영체제와 레드햇의 후원과 개발 공동체의 지원 아래 개발된 배포판이다.
- CentOS : 업스트림 소스인 레드햇 엔터프라이즈 리눅스와 완벽하게 호환되는 무료 기업 용 컴퓨팅 운영체제이다.
- 수세 : 독일에서 출시된 배포판으로 유럽에서 인기를 누리고 있다.

## Section 2. 리눅스 역사

### 넘어감

## Section 3. 리눅스 라이선스

### 종류

- GNU : GNU 는 유닉스가 아니다 (GNU's Not UNIX) 의 약자이다.
- 자유 소프트웨어 재단 : FSF(Free Software Foundation) 으로 1985년 리처드 스톨만이 설립한 재단이다.
- 오픈 소스 소프트웨어 : Open Source Software 1998 년 일부 커뮤니티에서 '자유 소프트웨어' 대신 '오픈 소스 소프트웨어' 라는 용어를 사용하기 시작했다.
- GNU GPL(General Public License) : GPL 은 자유 소프트웨어 재단에서 만든 Free 소프트웨어 라이선스다.
- GNU LGPL(Lesser General Public License) : LGPL 은 GPL 보다는 훨씬 완화된 조건의 공개 소프트웨어 라이선스이다. LGPL 이 적용된 라이브러리를 이용하여 개발하였을 경우 프로그램 소스코드는 공개하지 않아도 된다.
- BSD(Berkeley Software Distribution) 라이선스 : 버클리 캘리포니아 대학의 자유소프트웨어 저작권의 한 가지이다.
- 아파치(Apache) 라이선스 : 아파치 소프트웨어 재단에서 자체적으로 만든 소프트웨어에 대한 라이선스 규정이다.
- MIT(Massachusetts Institute of Technology) 라이선스 : BSD 라이선스를 기초로 작성된 BSD 계열 라이선스 중의 하나이다.
- MPL(Mozilla Public License) : MPL 의 특징은 소스코드와 실행파일의 저작권을 분리했다는 점이다.

## Section 4. 기본 설치 및 유형

### 리눅스 설치의 개요

- 리눅스 설치 파일은 해당 배포본의 홈페이지에서 다운로드 받을 수 있다.
- 리눅스는 단 하나의 제품 또는 한 종류의 제품군만 있는 것이 아니다.
- 리눅스 배포판마다 설치 환경과 설치 과정이 다르다.
- 리눅스 설치 유형은 배포판마다 다르지만 패키지에 따라 데스크탑형, 서버형, 사용자 정의형으로 구분한다.
- 설치 전에 시스템이 있는 모든 파일을 백업해 둔다.
- 멀티 부팅 시스템을 만든다면, 현재 운영체제의 배포 미디어를 가지고 있어야 한다.
- 부팅 드라이브를 다시 파티션 하는 경우라면, 운영체제의 부트로드를 다시 설치해야 할 수도 있고, 더 많은 경우에 운영체제 전체를 해당 파티션에 다시 설치해야한다.

## 리눅스 설치를 위한 하드웨어 정보 파악

- 하드웨어 정보
- 하드웨어 호환성
- 네트워크 설정

## 리눅스 설치하기

- 1단계 : 설치 초기화면 – Test this media & install CentOS7, Install CentOS8
- 2단계 : 설치 초기화면 – Rescue a CentOS system, Run a memory test
- 3단계 : 언어선택
- 4단계 : 설치 요약 확인
- 5단계 : 날짜와 시간 설정
- 6단계 : 설치 소스 – 저장소(repository) 서버 주소 변경
- 7단계 : 설치 소스 – 설치와 관련된 특수 저장 장치 설정
- 8단계 : 설치 소스 – 기본 저장 장치 하드디스크 파티션 설정
- 9단계 : 소프트웨어 선택
- 10단계 : Kdump 설정
- 11단계 : 네트워크 설정과 호스트명 지정
- 12단계 : 패스워드 지정

## Section 5. 파티션

### 파티션 특징과 종류

- 파티션이란, 하나의 물리적 디스크를 여러 개의 논리적인 디스크로 분할하는 것이다.
- 파티션은 주 파티션, 확장 파티션, 논리 파티션, 스왑 파티션으로 구분된다.

주 파티션	부팅이 가능한 <u>기본 파티션</u> 하나의 하드디스크에 최대 4개의 주 파티션 분할 가능 하드디스크를 4개 이상의 파티션으로 사용해야 할 때 하나의 확장 파티션
-------	---

	을 설정하여 확장 파티션 안에 여러 개의 논리 파티션을 분할하여 데이터 저장
확장 파티션	주 파티션 내에 생성, 하나의 물리적 디스크에 1개만 생성 파티션 번호는 1~4번이 할당 데이터 저장 영역을 위한 것이 아니라 논리 파티션을 생성
논리 파티션	확장 파티션 안에 생성되는 파티션 논리 파티션은 12개 이상 생성하지 않는 것을 권고하며 5번 이후의 번호가 붙여짐
스왑 파티션	하드디스크의 일부를 메모리처럼 사용하는 영역 주 파티션 또는 논리 파티션에 생성 프로그램 실행 시 부족한 메모리 용량을 하드디스크로 대신 리눅스 설치 시에 반드시 설치되어야 하는 영역 스왑 영역의 크기는 메모리의 2배를 설정하도록 권고

## 디스크와 장치명

- 분할된 파티션은 디스크의 장치 파일명 뒤에 숫자를 붙인다.
- 리눅스에서 파티션 만들고 마운트할 때 지정된 디바이스명을 사용한다.

## 파일 시스템

- 파일 시스템은 운영체제가 파일을 시스템의 디스크 파티션상에 구성하는 방식이다.
- 일정한 규칙을 가지고 파일을 저장하도록 규칙 방식을 제시한다.
- 파티션에 파일 시스템이 없으면, 파일 시스템 생성을 거쳐야 사용이 가능하다.
- 리눅스는 고유의 파일 시스템뿐만 아니라 다양한 파일 시스템을 지원하고 있다.

## LVM(Logical Volumn Manager)

- 여러 개의 하드디스크를 합쳐서 사용하는 기술로 한 개의 파일 시스템을 사용한다.
- 작은 용량의 하드디스크 여러 개를 큰 용량의 하나의 하드디스크처럼 사용한다.
- 서버를 운영하면서 대용량의 별도 저장 공간이 필요할 때 활용된다.
- 다수 개의 디스크를 묶어서 사용함으로써 파티션의 크기를 줄이거나 늘릴 수 있다.

## RAID

- RAID 는 복수 배열 독립 디스크의 약자이다. LVM은 용량, RAID는 성능(안전성)이 주목적.
- 여러 개의 물리적 디스크를 하나의 논리적 디스크로 인식하여 작동하게 하는 기술이다.

- 여러 개의 하드디스크에 일부 중복된 데이터를 나눠서 저장하는 기술이다.
- RAID 종류는 하드웨어 RAID 와 소프트웨어 RAID 로 나뉜다.
- 데이터를 저장하는 다양한 방법이 존재하며 이 방법들을 레벨이라 한다.
- 레벨이 높을수록 저장 장치의 신뢰성을 높이거나 전체적인 성능을 향상시키는 다양한 목적을 만족시킨다.
- 각 레벨의 장점을 합친 RAID 구성으로는 대표적으로 RAID 0+1과 RAID 1+0 이 있다.

## 파티션 분할

- fdisk 는 파티션 테이블을 관리하는 명령으로 리눅스의 디스크 파티션을 생성, 수정, 삭제 할 수 있는 일종의 유틸리티이다.
- fdisk 명령어 : a, l(파티션 목록), n(새로운 파티션 추가), t(파티션 변경), w, p, q

## Section 6. 부트 매니저

### 부트로드

- 부트스크립트 로더의 준말로 컴퓨터를 사용자가 사용할 수 있도록 디스크나 플래시에 저장 된 운영체제를 읽어 주기억장치에 적재해 주는 프로그램이다.
- 부트로더는 운영체제가 시동되기 이전에 미리 실행되면서 커널이 올바르게 시동되기 위해 필요한 모든 관련 작업을 마무리하고 최종적으로 운영체제를 시동시키기 위한 프로그램이다.
- 임베디드 시스템 부트로더란, PC 의 BIOS 와 OS Loader 의 기능을 수행하는 프로그램으로 시스템이 부팅할 때 가장 먼저 수행된다.
- 운영체제 실행에 필요한 환경을 설정하고 운영체제 이미지를 메모리에 복사한다.
- 부트로더는 부트매니저라고도 부르며 크기가 512바이트로 하드디스크의 첫번째 섹터인 MBR(Master Boot Record) 에 위치한다.
- 주 파티션마다 부트섹터가 할당된다.
- 분할된 주 파티션들은 자신의 부트 레코드를 MBR에 기록하여 실행된다.
- 한 컴퓨터에 다수개의 운영체제가 설치되어 있는 경우 작업 운영체제를 선택하여 부팅할 수 있게 한다.

## 런레벨(init):시스템 부팅 시 리눅스 커널이 최초 발생시키는 프로세스

- 리눅스 부팅 시 작동하는 서비스들이 있다. 런레벨에 따라 작동하는 서비스를 조정 가능 하다.
- 런레벨은 0 에서 6까지 총 7가지이다.
- 7가지 런레벨 중 리눅스가 가동 시 특정 모드의 레벨을 디폴트로 할 경우 파일 /etc/inittab 에 설정한다.
- /etc/inittab 파일 형식은 '코드 런레벨:행동:명령어'이다.
- 현재 실행되는 런레벨을 확인하는 명령어는 runlevel 이다.

## 로그인과 로그아웃

### 로그인

- 리눅스는 X윈도우상에서의 로그인/로그아웃과 콘솔상에서의 로그인/로그아웃이 있다.
- 로그인 과정 : 입력한 패스워드와 파일 /etc/passwd 필드 비교 -> 셸 설정 파일 실행 -> 로그인 셸 실행

### 로그아웃

- 로그아웃은 logout, exit 또는 조합키 Ctrl+D 를 사용한다.
- 관리자는 일정시간 동안 작업을 수행하지 않는 모든 사용자들을 강제로 로그아웃할 수 있다.

## Section 7. 사용자 생성 및 계정 관리

### 리눅스 명령어

#### which

- 명령어의 경로를 확인하는 명령어이다.
- 명령어의 위치를 찾아주거나 alias 를 보여주는 명령어이다.
- 형식 : which 명령어

#### alias

- 자주 사용하는 명령어를 특정 문자로 입력해 두고 명령어 대신 해당 문자를 사용할 수 있게 하는 명령어이다.
- 형식 : alias [ 별명='명령어']

#### unalias

- alias 기능을 해제한다.
- 형식: unalias [옵션][단축명령어]

### 환경변수 PATH

- PATH 는 실행 파일들의 디렉터리 위치를 저장해 놓는 환경변수이다.
- 명령어 echo \$PATH 는 지정된 PATH 값을 확인할 수 있다.
- 기존의 PATH 에 새로운 경로를 추가하는 방법에는 명령어 PATH 나 홈 디렉터리 의 .bash\_profile 에 PATH 를 추가한다. 기존 경로와 새 경로는 : 으로 구분한다.
- 형식 : export 변수명=\$변수명:변수값

## 리눅스 도움말

### man

- 리눅스에서 사용하는 명령어들의 매뉴얼을 제공한다.
- 형식 : man [섹션][옵션] 명령어

### info

- 리눅스 명령어의 사용 방법, 옵션 등을 나타낸다.
- man에 비해 제공되는 명령어가 한정적이다.
- 형식: info 명령어

### whatis

- 명령어에 대한 기능을 간략하게 나타낸다.
- 완전히 키워드가 일치해야만 해당 명령어의 기능을 확인할 수 있다.

### manpath

- man 페이지의 위치 경로를 검색하여 표시해 주는 명령어이다.

### whereis

- 찾고자 하는 명령어의 실행 파일 절대 경로와 소스코드, 설정 파일 및 매뉴얼 페이지를 찾아 출력하는 명령어이다.

### apropos

- man 페이지 설명에서 지정한 키워드를 포함하고 있는 명령어이다.



## 사용자 생성 명령어

### useradd

- 계정을 생성하는 명령어로 명령어 adduser 와 동일한 기능을 갖는다.
- 계정자의 홈 디렉터리는 '/home/계정명' 이다.
- 생성된 계정자 정보는 파일 /etc/passwd, /etc/shadow, /etc/group 에 저장된다.

### passwd

- 생성된 계정자의 패스워드를 입력 및 변경하는 명령어이다.
- 생성된 계정자의 패스워드는 /etc/shadow 파일 안에 암호화되어 기록된다.

### su

- su 는 switch user 의 줄임말이다.
- 현재의 사용자 계정에서 로그아웃하지 않고 다른 사용자 계정으로 로그인하여 해당 사용자의 권한을 획득하는 명령어이다.
- 하이픈을 붙이면 세션유지, 안붙이면 기존의 세션 종료

## 사용자 관련 파일

### /etc/default/useradd

- 명령어 useradd 로 사용자 계정을 추가할 때 사용되는 정보를 읽어오는 파일이다.

### /etc/passwd

- 계정자의 정보를 가지고 있는 파일로 리눅스에 로그인할 때 사용된다.

### /etc/shadow

- 계정자의 패스워드 정보가 암호화되어 있는 파일로 암호화 패스워드 및 계정의 유효 기간 등을 기록하고 있는 파일이다.

### /etc/login.defs

- 사용자 계정 설정과 관련된 기본값을 정의한 파일이다.
- 새로운 계정을 생성할 때 반드시 참조하는 파일이다.

## 사용자 계정 관리

### usermod

- 디렉터리 /home 에 위치한 사용자들의 정보를 변경하는 명령어이다.
- 사용자의 홈 디렉터리 변경, 그룹 변경, 유효기간 등을 변경한다.

### userdel

- 기존 계정 정보를 삭제하는 명령어이다.
- 사용자의 홈 디렉터리 변경, 그룹 변경, 유효기간 등을 변경한다.
- 옵션 없이 userdel 을 사용하면 /etc/passwd, /etc/shadow, /etc/group 에서 해당 계정자의 정보가 삭제된다.

### chage

- 패스워드의 만료 정보를 변경하는 리눅스 명령어이다.

## 그룹관리

### /etc/group

- 사용자 그룹에 대해 정의되어 있는 파일
- 모든 계정은 한 개 이상의 그룹에 포함되어 있다.

### /etc/gshadow

- 그룹의 암호를 MD5 로 하여 저장하며 그룹의 소유주, 구성원 설정이 가능하다.

### groupadd

- 새로운 그룹을 생성하는 명령어이다.

### groupdel

- 기존의 그룹을 삭제하는 명령어이다.
- 그룹 안에 소속되어 있는 계정명이 있을 경우 해당 그룹은 삭제되지 않는다.

### groupmod

- 그룹의 설정을 변경하는 명령어이다.

## 사용자 조회 명령어

### users

- 시스템에 로그인한 사용자 정보를 출력하는 명령어이다.

### who

- 현재 시스템에 접속해 있는 사용자들을 조회하는 명령어이다.
- 사용자 계정명, 터미널정보, 접속시간, 접속한 서버 정보 등을 확인할 수 있다.
- 관리자 root 와 일반 사용자 모드 사용이 가능하다.
- 명령어 'who am I' 또는 'whoami' 는 자신의 정보를 조회할 수 있다.

#### w

- 현재 접속 중인 사용자들의 정보를 나타내는 명령어이다.
- 확인 가능한 정보는 서버의 현재 시간 정보, 서버 부팅 후 시스템 작동 시간, 서버 접속자의 총 수, 접속자별 서버 평균 부하율, 접속자별 서버 접속 계정명, TTY명, 로그인 시간 정보 등이다.
- JCPU 는 w TTY 필드의 장치명에서 사용되는 모든 프로세스의 CPU 사용시간이다.
- PCPU 는 해당 프로세스 결과값에서 WHAT 필드에 나타나는 프로세스명에서 사용하는 CPU 총시간이다.

#### id

- 사용자 계정의 uid, gid, group 을 확인하는 명령어이다.

#### groups

- 사용자 계정이 속한 그룹 목록을 확인하는 명령어이다.

## Section 8. 디렉터리 및 파일

### 디렉터리 관리 명령어

#### pwd

- 현재 작업 중인 디렉터리의 위치를 나타내는 명령어이다.

#### cd

- 디렉터를 이동할 때 사용하는 명령어이다.
- 절대경로는 시작 위치와 상관없이 경로에 모든 디렉터를 표시하며, 절대 경로의 시작은 / 에서부터 시작한다.
- 상대 경로는 현재 작업 중인 디렉터를 기준으로 표시하는 경로이다.

#### mkdir

- 새로운 디렉터를 생성할 때 사용하는 명령어이다.

#### rmdir

- 디렉터리만 삭제하는 명령어로 디렉터리 안에 파일이 존재하는 경우 삭제되지 않는다.

### 파일 관리 명령어

#### ls

- 현재 위치한 디렉터리의 파일 목록들을 나타내는 명령어이다.

#### cp

- 파일 또는 디렉터를 복사하는 명령어이다.

#### rm

- 파일 또는 디렉터를 삭제하는 명령어이다.

#### mv

- 파일 또는 디렉터를 이동하거나 파일명을 변경할 때 사용하는 명령어이다.

#### touch

- 파일 크기가 0바이트인 빈 파일을 생성한다.
- 생성된 파일에 touch명령어를 쓸 경우 서버의 현재 시간으로 파일의 최근 사용한 시간과 최근 수정 시간 등 타임스탬프를 변경 한다.

#### file

- 파일 종류 및 파일 속성값을 나타내는 명령어이다.

#### find

- 현재 디렉터리에서부터 하위 디렉터리까지 주어진 조건의 파일을 찾아 해당 경로를 표시 한다.

#### locate

- 파일 위치를 찾는 명령어이다.

## 텍스트 파일 관련 명령어

#### cat

- 파일의 내용을 출력하는 명령어이다.

#### head

- 파일의 앞부분을 지정한 만큼 출력하는 명령어이다.

#### tail

- 파일의 마지막 행을 기준으로 지정한 행까지의 파일 내용 일부를 출력하는 명령어이다.

#### more

- 파일을 확인하는 명령어로 파일을 읽어 화면에 화면 단위로 끊어서 출력하는 명령어이다.
- 위에서 아래 방향으로만 출력되기 때문에 지나간 내용을 다시 볼 수 없다.

#### less

- 텍스트 파일을 한 번에 한 화면씩 나타내는 명령어이다.
- 기능적으로 more를 확장한 것으로 커서를 파일의 상하좌우로 이동할 수 있다.

#### grep

- 파일에서 특정한 패턴 또는 정규 문자식으로 나타낸 단어를 찾는 명령어이다.

#### wc

- 파일의 라인 수, 단어 수, 알파벳 수를 알려주는 명령어이다.

#### sort

- 명령어 결과나 문서 내용을 정렬하는 명령어이다.

#### cut

- 파일에서 특정 필드를 추출해 낸다. 필드는 구분자로 구분할 수 있다.

#### split

- 하나의 파일을 여러 개의 작은 파일로 분리할 때 사용한다.
- 파일의 내용을 라인 수로 분할할 수도 있고, 용량 단위로 분할할 수도 있다.
- 주로 디스켓에 파일을 나누어 복사하거나 백업할 때 CD-RW 용량 단위로 분할할 경우 유용하다.

## 파일 비교 명령어

#### diff

- 두 개의 파일을 행 단위로 비교하여 다른 부분을 출력하는 명령어이다.
- 두 개의 파일명을 매개변수로 사용하여 화면에 차이점을 나열한다.

#### cmp

- 두 개의 파일을 바이트 단위(문자단위)로 비교하여 출력하는 명령어이다.

#### comm

- 두 개의 파일의 행과 행을 비교하여 출력하는 명령어이다. 옵션이 중요함.

## 리다이렉션과 정규 표현식

### 리다이렉션

- 표준 입력과 표준 출력의 방향을 재지정하는 것이다.
- 표준 입력/출력/에러가 화면이 아닌 파일로 대체한다. 즉, 모니터로 하는 출력이 파일로 재지정 한다.
- 표준 입력 장치는 키보드, 표준 출력 장치는 모니터, 표준 에러 장치는 모니터이다.

### 파이프

- 둘 이상의 명령을 함께 묶어 출력 결과를 다른 프로그램이 입력으로 전환하는 기능이다.
- 현재 명령의 표준 출력을 다음 명령의 표준 입력으로 사용하는 것이다.
- 명령어와 명령어의 연결은 | 기호를 사용한다.
- 명령어1의 출력 결과는 명령어2의 입력으로 처리된다.
- 더 이상 처리할 명령어가 없으면 표준 출력 장치인 화면으로 출력한다.

정규표현식(문자열을 찾는 명령어(grep,find)에서 많이 사용됨)

기호	의미
^	라인의 첫 글자 <code>^a</code>
\$	라인의 끝 글자 <code>a\$</code>
.	한 글자 <code>a...b -&gt; 점이 세 개니까 ab사이에 문자가 3개</code>
*	* 기호 바로 이전의 글자는 정규 표현식이 0회 이상 반복 <code>abc*-&gt;ab,abc,abcc</code>
[ ]	대체 글자 목록을 [ ] 에 나열 "-로 문자의 범위를 지정 <code>[abc]d-&gt;ab,bd,cd</code>
[^]	대체 못할 글자 목록을 [^] 에 나열 "-로 문자의 범위를 지정 <code>[^abc]-&gt;ab,bd,cd가 오면 안됨.</code>
+	+ 기호 바로 이전 글자나 정규 표현식이 1회 이상 반복 <code>abc+-&gt;abc,abcc</code> 반드시 1회는 나와야 한다. *과의 차이.
?	? 기호 바로 이전 글자나 정규 표현식이 없거나 1회만 존재 <code>abc? -&gt; ab,abc 만됨. abcc안됨.</code>
( )	부분 정규 표현식의 시작과 끝을 표시 <code>a(bc)*-&gt;a,abc</code>
	로 구분된 단어들 중 최소 하나 존재 <code>a(b c)-&gt;ab,ac</code>
{m, n}	{ } 기호 바로 이전 글자나 정규 표현식이 m개 이상 n개 이하 반복 <code>a{1,3}b-&gt;ab,aab,aaab</code>

## Section 9. 기타 명령어

### 네트워크 관련 명령어

## ping

- 외부 호스트에 신호를 보내며 신호를 받은 호스트는 응답을 주면서 서로 네트워크가 연결되어 있음을 확인시켜주는 명령어이다.

## traceroute

- 목적지 호스트까지의 경로를 표시하고 그 구간의 정보를 기록하는 명령어이다.
- 목적지 호스트까지의 패킷 전송 지역을 측정하거나 목적지 호스트로 향하는 경로상에 어떤 장애가 있는 경우 위치를 파악할 수 있다.

## nslookup

- 도메인명으로 IP 주소를 조회하거나 또는 IP 주소로 도메인명을 조회하는 명령어이다.

## dig (domain information groper)

- 명령어 nslookup 과 유사한 기능을 가진 명령어로 호스트명에 대한 IP 주소 정보 또는 IP 주소에 대한 호스트명을 조회하는 명령어이다.
- 서버명은 확인하고자 할 네임 서버를 지정하는 것이며 지정하지 않을 경우 /etc/resolv 에 등록된 네임 서버를 이용하여 루트 서버를 조회하게 된다.

## host

- 호스트명을 알고 있는데 IP 주소를 모르거나 그 반대의 경우에 사용하는 명령어이다.
- 호스트명을 이용하면 IP 주소뿐만 아니라 하위 호스트명도 조회할 수 있다.
- 호스트는 시스템에 등록된 DSN 서버를 이용하여 검색하는데 다른 DNS 서버를 이용해서 따로 지정할 수 있다.

## hostname

- 시스템 이름을 확인하거나 변경할 때 사용하는 명령어이다.

## 시스템 종료 명령어

### shutdown

- 시스템을 종료하거나 재부팅하는 명령어이다.
- 일반적으로 현재 수행 중인 프로세스들을 종료하며 sync 를 수행하여 저장되지 않는 데이터를 디스크에 저장하고 모든 파일 시스템을 mount 시킨 후에 시스템을 종료한다.
- root 사용자만이 권한을 가지고 있는 명령어이다.

#### init

- shutdown 명령어 동일한 기능을 가진 명령어이다.

#### reboot

- 시스템을 재부팅하는 명령어이다.

#### halt

- 시스템을 종료하는 명령어이다.

### 기타 명령어

#### cal

- 시스템에 설정된 달력을 출력하는 명령어이다.

#### date

- 시스템의 날짜와 시간을 표시하거나 변경한다.

#### clear

- 터미널의 내용을 지우는 명령어이다.

#### tty

- 현재 사용하고 있는 단말기 장치의 경로명과 파일명을 나타낸다.
- 텔넷 등에서 동일한 계정으로 여러 개 로그인한 경우 확인 시 유용하다.

#### time

- 프로그램이 수행되는데 걸리는 시간을 측정하여 출력하는 명령어이다.
- 세 가지 시간 결과 real, user, sys를 보여준다. real 을 총 수행시간, user 는 CPU가 사용자 영역에서 보낸 시간, sys 는 시스템 호출 실행에 걸린 시간이다.

#### wall

- 모든 로그인된 사용자들에게 터미널을 통해 메시지를 전달받는 명령어이다.

#### write

- 해당 사용자에게 메시지를 전달하는 명령어이다.

#### mesg

- write 를 사용해서 들어오는 메시지 수신 여부를 확인하고 제어하는 명령어이다.