

제1장 리눅스 소개 및 설치

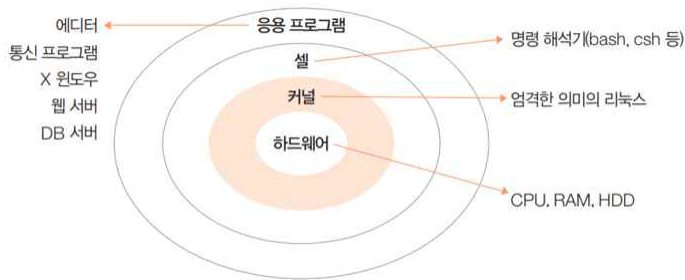
1. 리눅스의 개요

1) 리눅스 특징

- ① 1991년 8월 리누스 토발즈에 의해 리눅스 커널 0.01 버전을 개발
- ② 리눅스는 공개 소프트웨어이며 무료로 사용할 수 있음
- ③ 유닉스와 완벽한 호환성을 유지하며, 서버용 운영체제로 많이 사용

2) 리눅스 구조

- ① 커널(Kernel)은 HW 장치의 지원정보 및 HW를 제어하는 코드 등이 포함된 리눅스의 핵심
- ② 셸(Shell)은 사용자가 입력한 명령을 해석해 커널로 전달하는 명령어 해석기
- ③ 응용 프로그램은 리눅스 운영체제를 운영관리하기 위한 각종 프로그램



3) 리눅스 배포판

- ① 여러 단체 및 기업에서 커널에 컴파일러, 셸, 기타 응용프로그램들이 통합된 배포판을 제작
- ② 리눅스 배포판의 종류에는 크게 레드햇 계열, 데비안 계열, 슬랙웨어 계열로 나뉜다.
- ③ 가장 대표적인 리눅스 배포판으로 CentOS, 우분투 리눅스, 페도라 리눅스 등 이 있다.

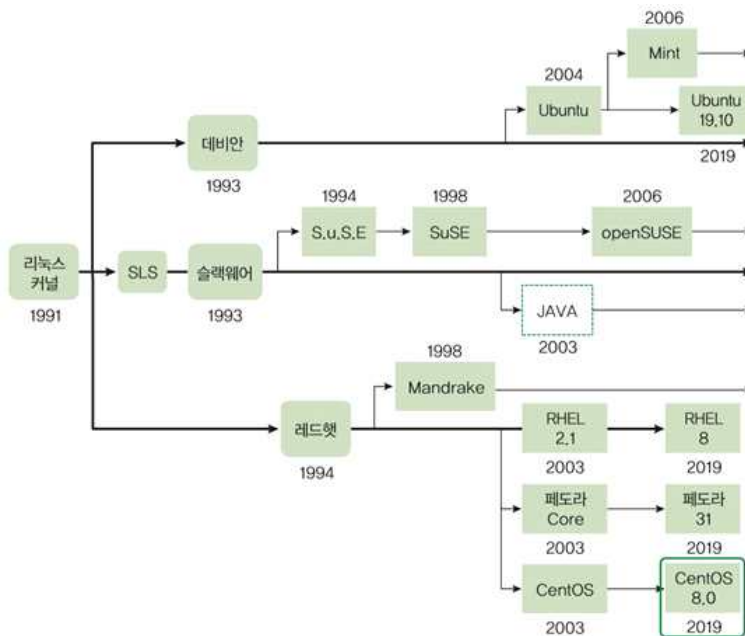


그림 1-4 주요 리눅스 배포판의 계통도

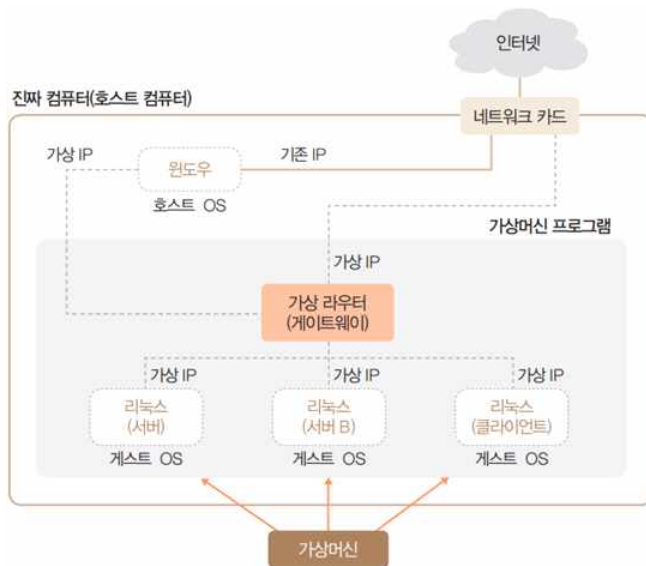
2. 가상머신

1) 개념

- ① 컴퓨터(호스트)에 설치된 운영체제에 가상의 컴퓨터를 만들고, 그 안에 또 다른 운영체제를 설치할 수 있는 프로그램
- ② PC에 설치된 Windows를 Host OS 이고 가상머신에 설치한 OS를 Guest OS라고 한다.
- ③ 가상머신의 종류에는 VMware, Virtual PC, VirtualBox 가 있다.

2) 가상머신 생성

- ① Host 컴퓨터의 자원을 빌려 사용하므로, Host의 성능에 영향을 주고 Host의 성능에 영향을 받는다.
- ② Host와 Guest 간의 연결 및 통신이 가능하다.



[실습하기] 아래 표를 참고하여 가상머신을 생성하시오.

가상머신 이름	Guest OS	HDD	RAM	기타
CentOS8_Desktop	CentOS 8 64-bit	20 GB	2 GB	2core
CentOS8_Server	CentOS 8 64-bit	20 GB	1 GB	1core

3. 리눅스 설치

1) CentOS

- ① 전 세계적으로 가장 유명한 리눅스 배포판 중 하나로 레드햇(RedHat)사에서 제작했다.
- ② 유료 레드햇 엔터프라이즈 버전을 무료 CentOS로 배포(<https://www.centos.org/>) 하고 있다.

☞ [실습하기] 아래 표를 참고하여 CentOS를 설치하시오.

가상머신 이름	소프트웨어 선택	설치 대상	KDUMP	네트워크
CentOS8_Desktop	워크스테이션(GNOM 포함)	자동 파티션	비활성	연결
CentOS8_Server	최소설치	자동 파티션	비활성	연결

2) 주요 디렉터리(파티션)

마운트 포인트	내용
필수	/ root 파티션으로 시스템 설정 등 주요 내용이 저장된다.
	/boot boot 파티션으로 리눅스 부팅 커널이 저장된다. 대략 500MB를 할당한다.
	swap 메모리(RAM)가 부족할 때 사용되는 영역으로 대개 RAM 크기의 2배 정도로 설정한다.
선택	/usr 명령어, 응용 프로그램 등이 주로 저장된다.
	/var 로그 파일 등이 주로 저장된다.
	/tmp 시스템 사용 중에 발생하는 임시 파일이 저장된다.
	/home 사용자 홈 디렉터리가 위치한다.

3) 시작과 종료

- ① 'shutdown' 명령어의 옵션 중 'now' 부분에 시간을 지정하면 지정한 시간에 시스템을 종료
- ② 옵션 없이 바로 종료하는 'halt' 명령어보다 'shutdown' 명령어로 안전한 시스템 종료를 권장
- ③ 로그아웃은 현재 접속한 사용자의 접속을 종료를 의미하고 Ctrl+D 단축키를 많이 사용

구분	명령어	설명
시스템 종료	#poweroff	시스템 전원을 차단하여 시스템 종료
	#shutdown -P now	다양한 옵션을 이용한 시스템 종료 명령어
	#halt	옵션 없이 바로 시스템 종료
재부팅	#reboot	옵션 없이 시스템 재부팅
	#shutdown -r now	다양한 옵션을 이용한 재부팅
로그아웃	#logout	현재 사용자의 시스템 접속을 종료
	#exit	

제2장 리눅스 기본

1. 리눅스 명령어

1) 프롬프트와 명령어 형식

- ① 사용자의 명령입력을 기다리는 표시로 일반 관리자는 \$, 시스템 관리자는 #으로 표시됨(Bash Shell)
- ② 리눅스 명령어는 [명령] [옵션] [인자] 형식을 가지며 각 사이는 여러 칸 띄어쓰기 가능
- ③ 명령어의 일부만 입력한 후에 Tab키를 눌러 나머지 명령어를 자동완성

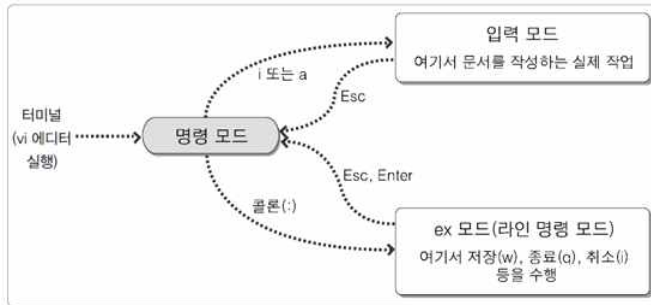
2) 기본 명령어

명령어	사용 예	설명
ls	#ls #ls /etc/sysconfig #ls -a #ls -l	현재 디렉터리의 파일 목록을 조회 #ll 명령어로 더 많이 사용
cd	#cd 또는 #cd . #cd .. #cd ~centos #cd /etc/sysconfig	현재 디렉터를 이동 ./ : 현재 디렉터리 ../ : 상위 디렉터리 ~/ : 홈 디렉터리
pwd	#pwd	Print Working Directory 현재 작업 중인 디렉터리의 경로를 출력
touch	#touch abc.txt	크기가 0인 새 파일을 생성
cp	#cp aaa.txt bbb.txt #cp -r aaa bbb	파일이나 디렉터를 복사
mv	#mv aaa.txt /etc/sysconfig #mv aaa.txt bbb.txt	파일이나 디렉터리의 이름을 변경 또는 이동
rm	#rm aaa.txt #rm -r abc #rm -rf abc	파일이나 디렉터를 삭제 r 옵션 : recursive f 옵션 : force
mkdir	#mkdir abc	새로운 디렉터리 생성
rmdir	#rmdir abc	비어있는 디렉터리 삭제
cat	#cat a.txt #cat a.txt > b.txt #cat a.txt >> b.txt	파일의 내용을 화면에 출력(concatenate) a.txt 내용을 b.txt에 생성 및 쓰기(덮어쓰기) a.txt 내용을 b.txt에 생성 및 추가
head, tail	#head aaa.txt #head -3 aaa.txt #tail -5 aaa.txt	파일의 앞 10행 내용 또는 마지막 10행 내용 출력
find	#find /home -name 'Hello*'	파일 검색
clear	#clear	터미널 화면을 깨끗하게 지워줌
history	#history	지금까지 입력한 명령어를 모두 출력

2. VI 에디터

1) VI 명령모드

- ① 윈도우즈의 메모장처럼 간단한 편집 프로그램으로 'visual' 이란 뜻으로 리눅스의 기본 에디터
- ② 리눅스에서 활용도가 가장 높은 에디터로 많은 연습으로 숙지



2) VI 에디터 명령어

구분	명령어 / 키	설명
실행하기	vi 파일명	파일명을 vi편집기로 열기
	view 파일명	파일명을 vi편집기로 읽기모드로 열기
	vi -R 파일명	파일명을 vi편집기로 읽기모드로 열기
입력/명령 모드	i	현재 커서의 위치에서 입력모드 전환
	a	현재 커서의 다음 위치에서 입력모드 전환
	o	현재 커서의 다음 행에 입력모드 전환
	R	현재 커서의 위치에서 덮어쓰기 모드로 전환
	esc	명령모드로 전환
커서 이동	w	다음 단어의 첫 문자로 커서 이동
	e	다음 단어의 끝 문자로 커서 이동
	b	이전 단어로 이동
	gg	첫 행으로 이동
	G	마지막 행으로 이동
	Ctrl+e / Ctrl+y	한 줄씩 아래로 이동 / 한 줄씩 위로 이동
	:숫자	해당 숫자 행으로 이동
삭제	x (소문자)	현재 커서가 위치한 글자 삭제
	X (대문자)	현재 커서가 위치한 앞글자 삭제
	dd	한줄 삭제
	숫자dd	현재 커서부터 숫자만큼의 행 삭제
	u	이전에 수행한 명령을 취소(윈도우 Ctrl+z)

복사/붙여넣기	yy	현재 행을 버퍼로 복사
	숫자yy	현재 커서의 행에서 부터 숫자만큼 행 복사
	p (소문자)	버퍼에 있는 내용을 커서 뒤에 붙여넣기
	P (대문자)	버퍼에 있는 내용을 커서 앞에 붙여넣기
블록지정	v (소문자)	블록지정
	V (대문자)	줄 단위 블록지정
	y	블록 지정된 부분 버퍼로 복사
	d	블록 지정된 부분 삭제
	shift + v + g	전체선택
검색	/name	현재 커서 이후부터 문자열 name 검색
	n	다음 아래 방향으로 문자열 검색
	N	다음 위 방향으로 문자열 검색
문자열 교체	:s/"apple"/"orange"/g	현재 행의 'apple'를 'orange'로 교체
	:%s/"apple"/"orange"/g	파일 전체 'apple'를 'orange'로 교체
파일저장	:w	수정한 내용 저장
	:w 파일명	파일명에 저장하기
	:q	저장하지 않고 종료
	:q!	저장하지 않고 강제종료
	:wq	파일을 저장하고 종료
기타	:set nu	행 번호 표시
	:set nonu	행 번호 취소

3) swap 파일

- ① vi 에디터로 편집중인 파일이 비정상적으로 종료되었을 때 생성되는 임시저장 파일
- ② 한번 편집했던 파일의 swap파일을 제거해야 다시 편집할 수 있게 된다.


파일명	명령어	기타
.파일명.swp	#rm -f .파일명.swp	파일명 앞에 .은 리눅스에서 숨김파일

3. 사용자 관리와 파일속성

1) 사용자와 그룹


- ① 리눅스는 다중 사용자 시스템으로 1대의 리눅스에 여러 명이 동시에 접속해서 사용 할 수 있다.
- ② 리눅스를 설치하면 기본적으로 root라는 이름을 가진 슈퍼유저가 있으며 최고 관리자 이다.
- ③ 모든 사용자는 하나 이상의 그룹에 소속된다.

명령어	사용 예	설명
useradd adduser	#useradd newuser #useradd -g group1 newuser	새로운 사용자를 추가
passwd	#passwd newuser	사용자의 비밀번호를 지정하거나 변경
usermod	#usermod -g group2 newuser	사용자 속성을 변경
userdel	#userdel newuser #userdel -r newuser	사용자를 삭제
groups	#groups #groups newuser	사용자가 소속된 그룹을 조회
groupadd	#groupadd newgroup	그룹을 생성
groupdel	#groupdel newgroup	그룹을 삭제

 [실습하기] 사용자를 추가하시오.


```
# useradd test1
# adduser test2
```

test1, test2 사용자와 test1, test2 그룹이 생성, 홈 디렉터리 /home/test1, /home/test2 확인

 [실습하기] test1 사용자의 기본 그룹을 bigdata로 지정하시오.

```
# usermod -g bigdata test1
```

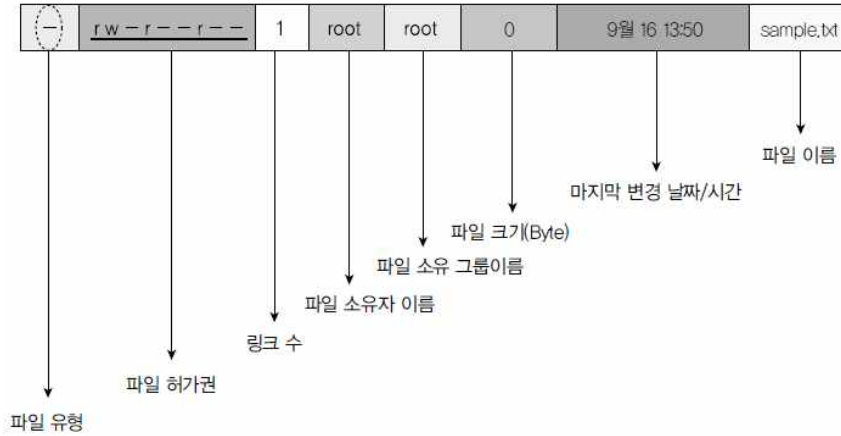
bigdata 그룹이 미리 생성되어 있어야 함

 [실습하기] 사용자와 그룹 정보를 확인 하시오.

```
#vi /etc/passwd
#vi /etc/group
```

2) 파일과 디렉터리의 속성과 파일유형

- ① 리눅스는 각각의 파일과 디렉터리마다 소유권과 허가권이 있다.
 ② 파일유형이 디렉터리일 경우 'd', 일반적인 파일일 경우에는 '-', 링크일 경우에는 'l' 이 표시된다.



3) 파일 허가권과 파일 소유권

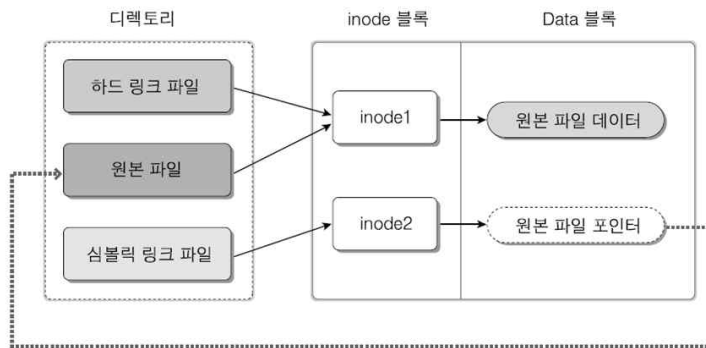
- ① 'rw-', 'r--', 'r--' 3개씩 끊어서 읽는다.
 ② 8진수 숫자로도 표시 가능하며 아래 표와 같다.
 ③ 디렉터리는 일반적으로 소유자/그룹/기타사용자 모두에게 실행권한이 설정된다.
 ④ 파일 소유권은 파일을 소유한 사용자와 그룹을 의미한다.

소유자(User)			그룹(Group)			그 외 사용자(Other)		
r	w	-	r	-	-	r	-	-
4	2	0	4	0	0	4	0	0
6			4			4		

명령어	사용 예	설명
chmod	#chmod 777 abc.txt #chmod -r 777 abc.txt	파일 허가권 변경
chown	#chown centos abc.txt #chown centos.centos abc.txt	파일 소유권 변경
chgrp	#chgrp centos abc.txt	파일 소유권 그룹 변경

4) 링크

- ① 파일의 링크는 하드링크와 심볼릭 링크가 2가지가 있다.
- ② inode는 리눅스의 파일 시스템에서 사용하는 자료구조로 파일이나 디렉터리의 여러 정보가 있다.



명령어	사용 예	설명
ln	#ln abc.txt hardln #ln -s abc.txt softln	링크 생성 -s 옵션 심볼릭 링크

☞ [실습하기] 아래의 작업을 수행하시오.

```
#ln -s /home/bigdata/workspace workspace/
```

5) 마운트

- ① 물리적인 장치인 하드디스크 파티션, CD/DVD, USB 등을 사용하기 위해 특정한 위치(디렉토리)에 연결하는 과정을 마운트라고 한다.

명령어	사용 예	설명
mount	#mount /dev/cdrom /media/cdrom	#mount [장치명] [대상디렉토리]
umount	#umount /media/cdrom	#umount [대상디렉토리]

4. 리눅스 패키지(프로그램) 설치

1) RPM (Redhat Package Manager)

- ① RPM은 윈도우의 'setup.exe'와 비슷한 CentOS용 설치파일로 *.rpm 이며, '패키지'라고 부른다.
- ② RPM의 가장 큰 단점으로 의존성 문제로 이를 해결하기 위해 YUM 명령어 이다.

구분	명령어	설명
설치	#rpm -Uvh 패키지파일명.rpm	U -> 패키지 설치 v -> 설치과정 확인 h -> 설치 진행과정을 #기호로 화면 출력
삭제	#rpm -e 패키지명	e -> 패키지 삭제(erase)
조회	#rpm -qa 패키지명 #rpm -qi 패키지명	qa -> 시스템에 패키지 설치여부 확인 qi -> 시스템에 설치된 패키지의 상세정보 확인

2) YUM

- ① RPM 패키지 설치의 의존성 문제를 해결하기 위한 RPM 설치도구 이다.
- ② 인터넷을 통해 필요한 패키지를 패키지 저장소(Repository)에서 자동으로 다운로드하는 방식이다.

구분	명령어	설명
설치	#yum install 패키지명 #yum install -y 패키지명	패키지 설치
삭제	#yum remove 패키지명	패키지 삭제
조회	#yum info 패키지명 #yum check-update	패키지 정보조회 업데이트 가능한 패키지 목록조회

☞ [실습하기] 아래 표를 참고하여 패키지를 설치하시오.

패키지명	설명
net-tools	네트워크 작업을 위한 명령어 패키지
wget	웹 서버 파일을 다운로드 하기 위한 패키지
vim	vi 에디터 확장 패키지(설치 후 설정 작업)

☞ [실습하기] 패키지 업데이트

```
# yum update
```

☞ [실습하기] vim 설정 작업

```
# vi /etc/profile
```

```
alias vi=vim
```

← 마지막 줄에 추가

```
# vi /etc/vimrc
```

```
set number
set tabstop=2
set shiftwidth=2
set autoindent
```

← 마지막 줄에 추가

```
# source /etc/profile
```

5. 파일 아카이브와 압축

1) 파일 아카이브(Archive)

- ① 리눅스에서 파일 아카이브는 압축과 다른 디렉토리를 포함한 모든 파일들을 하나의 파일로 만든다.
- ② 일반적으로 tar+gzip 형태를 많이 이용한다.

명령어	사용 예	설명
tar	#tar -cvf abc.tar abc	abc 파일을 abc.tar 파일로 묶기
	#tar -cvf abc.tar aaa bbb ccc	aaa, bbb, ccc파일을 abc.tar 파일로 묶기
	#tar -cvzf abc.tar.gz abc	abc 파일을 묶기 + gzip 압축
	#tar -xvf abc.tar	abc.tar 파일 풀기
	#tar -xvzf abc.tar.gz	abc.tar.gz 파일 압축해제 + tar 풀기

2) 파일 압축(Compress)

- ① 리눅스에서는 파일이나 디렉터리들을 묶는 것(archive)과 압축(compress)하는 기능이 나뉘어져 있다.
- ② 일반적으로 tar와 gzip을 사용한 .tar.gz의 확장자를 가지는 압축 아카이브를 많이 사용한다.

명령어	사용 예	설명
xz	#xz 파일명	'파일명.xz' 압축파일 생성
	#xz -d 파일명.xz	'파일명.xz' 압축파일 해제
bzip2	#bzip2 파일명	'파일명.bz2' 압축파일 생성
	#bzip2 -d 파일명.bz2	'파일명.bz2' 압축파일 해제
gzip	#gzip 파일명	'파일명.gz' 압축파일 생성
	#gzip -d 파일명.gz	'파일명.gz' 압축파일 해제
zip	#zip 파일2.zip 파일1	압축할 파일1을 '파일2.zip'로 생성
unzip	#unzip 파일명.zip	'파일명.zip'을 압축해제

6. 시스템 관리 및 설정

1) 디스크 사용량, 공간 확인

- ① df 명령을 사용해서 시스템 전체 디스크 사용량을 확인 할 수 있다.
- ② du 명령을 사용해서 디렉터리별 디스크 사용량을 확인 할 수 있다.

명령어	사용 예	설명
df	#df -h #df -m	디스크의 사용량을 출력
du	#du -h /home #du -sh /home	/home 디렉터리의 사용량을 출력

2) 시스템 모니터링

- ① 프로세스의 CPU, MEMORY 사용량 등 시스템의 전반적인 상황을 모니터링하는 명령어
- ② 기본적으로 3초 간격으로 화면이 갱신되며 Ctrl+C로 종료한다.

명령어	사용 예	설명
top	#top #top -d 1	시스템 모니터링
kill	#kill 27551 #kill -9 27551	pid 27551 프로세스 종료 pid 27551 프로세스 강제종료

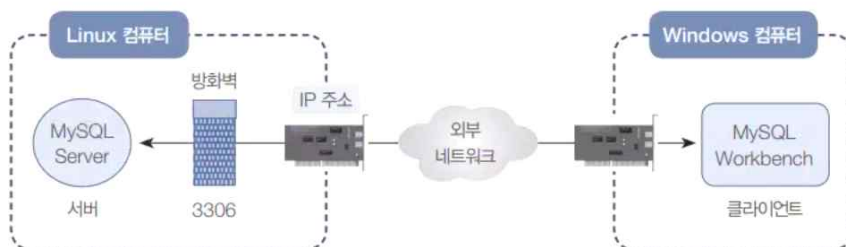
3) 서비스

- ① 서비스는 데몬(daemon)이라고 부르는 서버 프로세스를 말한다.
- ② 대표적인 서비스의 종류에는 httpd, mysqld, firewalld 등이 있으며 백그라운드로 실행된다.
- ③ 일반적으로 실행/종료/재시작은 'systemctl start/stop/restart 서비스명' 으로 설정한다.

구분	사용 예	비고
시작/중지/재시작	#systemctl start/stop/restart 서비스명	
상태확인	#systemctl status 서비스명	
사용/사용안함	#systemctl enable/disable 서비스명	

※ firewalld(방화벽)

- 방화벽은 허용된 접속을 제외한 외부 사용자들이 내부 네트워크에 접근하지 못하도록 하는 네트워크 방어 도구
- firewalld는 CentOS7부터 이전의 iptables를 대체해 새롭게 선보인 패킷 필터링 방화벽 프로그램
- CentOS뿐만아니라 RHEL과 페도라 등 여러 리눅스 시스템에서 기본 방화벽으로 채택되어 사용됨



4) 날짜 및 시간

- ① date 명령어로 리눅스 시스템의 날짜 및 시간을 설정
- ② 인터넷 시간으로 동기화를 위해 rdate 명령어를 사용

명령어	사용 예	설명
date	#date #date -s "13:00:01" #date -s "20180101 13:00:01" #date -s "2018-01-01 13:00:01"	시스템 날짜 및 시간을 조회 시스템 날짜 및 시간을 수정
rdate	#rdate -p time.bora.net #rdate -s time.bora.net	타임서버 시간 조회 타임서버 시간으로 동기화

5) cron

- ① cron은 주기적으로 반복적인 잡(Job)을 자동으로 실행 할 수 있도록 시스템 작업을 예약
- ② cron과 관련된 데몬(서비스)은 crond이고 관련 파일은 /etc/crontab 이다.
- ③ crontab은 분 시 일 월 요일 사용자 실행명령 형식을 갖는다.

`30 23 1 * * /usr/bin/ls -l ~user1 > ~user1/cron.out`
 ①분 ②시 ③일 ④월 ⑤요일 ⑥작업 내용

crontab 파일 형식	설명
01 * * * * root 작업내용	매일 매시간 1분에 작업내용 실행
02 4 * * * root 작업내용	매일 새벽 4시 2분 작업내용 실행
03 4 * * 0 root 작업내용	일요일 새벽 4시 3분 작업내용 실행
42 4 1 * * root 작업내용	매월 1일 새벽 4시 42분 작업내용 실행

☞ [실습하기] cron 작업

1. cron 작업 내역 확인

```
#vi /var/log/cron
```

2. hello.py 파일 생성 및 출력 확인

```
#cd ~
#vi hello.py
--- hello.py 내용 ---
print('Hello Linux!')
```

3. cron 작업 설정

```
#vi /etc/crontab
--- crontab 마지막 줄 입력 ---
* * * * * root python3 /root/hello.py
```

제3장 셸 스크립트 프로그래밍

1. 셸 스크립트

1) 셸(Shell)

- ① 셸은 사용자가 입력한 명령을 해석해 커널로 전달하는 명령어 해석기
- ② CentOS는 기본적으로 bash Shell을 사용하며, 셸 스크립트는 셸 환경에서 실행되는 프로그래밍이다.

2) 셸 환경 설정 파일

- ① 셸 설정파일로 사용자가 시스템에 로그인을 하면 자동적으로 실행되는 파일이다.
- ② 셸 설정파일은 시스템 전체의 초기화를 위한 파일과 개인 유저 환경을 위한 초기화 파일이 있다.
- ③ 설정파일 수정 후 source 명령어로 현재 로그인된 셸에 적용

파일명	설명
/etc/profile	시스템 전역 셸 변수 초기화
/etc/bashrc	셸 함수와 Alias를 위한 시스템 전역 변수 정의
~/.bashrc	유저 개인의 Alias 및 변수 설정 파일

☞ [실습하기1-1] Hello World 출력하기

```
# vi /home/shell/1_1_hello.sh
```

```
01 #!/bin/sh
02 echo "Hello World"
03 exit 0
```

```
# sh 1_1_hello.sh
```

```
# ./1_1_hello.sh
```

☞ [실습하기1-2] 환경변수 확인하기

```
# vi /home/shell/1_2_env.sh
```

```
01 #!/bin/sh
02 echo "USER : $USERNAME"
03 echo "HOST : $HOSTNAME"
04 exit 0
```

```
# ./1_2_env.sh
```

※ 각 스크립트 파일을 실행하기 위해 파일허가권 '실행권한'을 추가해야 됨

```
# chmod 755 hello.sh
```

```
# chmod +x name.sh
```

☞ [실습하기1-3] 셸 환경 설정파일 실습하기

```
# vi /etc/profile
```

```
#마지막 라인
export NAME="홍길동"
echo "Hello $NAME!"
```

```
# source /etc/profile
```

```
# echo $NAME
```

```
# logout
```

2. 변수

1) 기본규칙

- ① 모든 변수는 문자열(String)로 취급한다.
- ② 변수를 참조할 때는 변수명 앞에 \$ 키워드를 붙인다.
- ③ 변수를 대입할 때 '=' 좌우에는 반드시 공백이 없어야 한다.

2) 변수 연산

- ① 변수에 들어있는 값에 +, -, *, / 등 연산을 하려면 'expr' 키워드를 사용한다.
- ② 반드시 수식과 함께 백틱(`)으로 묶어야 한다.
- ③ 수식에 괄호 또는 곱(*)을 연산을 위해서 해당 기호 앞에 역슬러쉬(\)를 붙인다.

☞ [실습하기2-1] 셸 스크립트 변수 실습하기

```
# vi /home/sh/2_1_variable.sh
```

```
01 #!/bin/sh
02 value="Hello World"
03 echo $value
04 echo "$value"
05 echo '$value'
06 echo \ $value
07 exit 0
```

```
# ./2_1_variable.sh
```

☞ [실습하기2-2] 셸 스크립트 연산 실습하기

```
# vi /home/sh/2_2_operator.sh
```

```
01 #!/bin/sh
02 num1=100
04 num2=$num1+200
05 echo "num2 : $num2"
06 num3=`expr $num1 + 200`
05 echo "num3 : $num3"
05 num4=`expr \( $num1 + 200 \) / 10 \* 2`
05 echo "num4 : $num4"
07 exit 0
```

```
# ./2_2_operator.sh
```

3. 조건문

1) if

형식

```
if [ 조건 ]
```

```
then
```

```
    참일 경우 실행
```

```
fi
```

2) if ~ else

형식

```
if [ 조건 ]
```

```
then
```

```
    참일 경우 실행
```

```
else
```

```
    거짓일 경우 실행
```

```
fi
```

☞ [실습하기3-1] 셸 스크립트 조건문 실습하기

```
# vi /home/sh/3_1_if.sh
```

```
01 #!/bin/sh
02 if [ "woo" = "woo" ]
03 then
04     echo "참입니다."
05 fi
06 exit 0
```

※ '[조건]'의 사이의 각 단어에는 모두 공백이 있어야 함

```
# ./3_1_if.sh
```

☞ [실습하기3-2] 셸 스크립트 조건문 실습하기

```
# vi /home/sh/3_2_if_else.sh
```

```
01 #!/bin/sh
02 if [ "woo" != "woo" ]
03 then
04     echo "참입니다."
05 else
06     echo "거짓입니다."
05 fi
06 exit 0
```

```
# ./3_2_if_else.sh
```


4. 반복문

1) for-in

형식

```
for 변수 in 값1 값2 값3...
```

```
do
```

```
    반복할 문장
```

```
done
```

2) while

형식

```
while [조건]
```

```
do
```

```
    반복할 문장
```

```
done
```

☞ [실습하기4-1] 셸 스크립트 반복문 실습하기

```
# vi /home/sh/4_1_for.sh
```

```
01 #!/bin/sh
02 sum=0
03 for((k=1;k<=10;k++))
04 do
05     $sum=`expr $sum + $k`
06 done
07 echo "1부터 10까지 합 : $sum"
08 exit 0
```

```
# ./4_1_for.sh
```

☞ [실습하기4-2] 셸 스크립트 반복문 실습하기

```
# vi /home/sh/4_2_while.sh
```

```
01 #!/bin/sh
02 sum=0
03 k=1
04 while [ $k -le 10 ]
05 do
06     $sum=`expr $sum + $k`
07     $k=`expr $k + 1`
08 done
09 echo "1부터 10까지 합 : $sum"
10 exit 0
```

```
# ./4_2_while.sh
```

5. 스크립트 실습

☞ [실습하기5-1] 백업 스크립트 실습하기

vi /root/backup.sh

```
01 #!/bin/sh
02 date=$(date +%Y-%m-%d-%H-%M-%S)
03 name="backup-$date.tar.gz"
04 tar -cvzf /root/$name /home/shell
05 exit 0
```

vi /etc/crontab

```
* * * * * root /root/backup.sh
```

제4장 리눅스 서버 구축

1. SSH 서버

1) SSH

- ① SSH는 Secure Shell로 원격지의 서버에 접속하기 위한 프로그램 또는 프로토콜
- ② SSH는 전통적인 텔넷 접속에서 보안을 강화한 접속 프로토콜로 openssh를 설치
- ③ CentOS8은 기본적으로 OpenSSH가 설치되어 있어 따로 설치 할 필요가 없음



2) SSH 접속 클라이언트 Putty

- ① 원격 웹 클라이언트 putty 접속 실습을 위해 가상머신의 고정 ip 설정
- ② Windows용 SSH 클라이언트인 Putty 사용법을 숙지

☞ [실습하기] CentOS 원격 서비스 구축하기

1-1. CentOS 설치, 기본 필수 패키지 설치 및 설정

- Centos8 최소 설치, 설치과정 생략, 패키지 설정 생략

1-2. 기본 필수 패키지 설치 및 업데이트

```
# yum -y install net-tools
# yum -y install wget
# yum -y install vim
# yum -y update
```

1-3. Hostname(시스템명) 설정

```
# vi /etc/hostname
```

```
#기존내용 삭제 후 가상머신 이름과 동일하게 입력
Server101
```

1-4. selinux 해제

```
# vi /etc/selinux/config
```

```
SELINUX=disabled
```

1-5. ssh 설정

```
# vi /etc/ssh/sshd_config
```

```
84: GSSAPIAuthentication no ← yes에서 no로 수정
```

1-6. 종료

```
# poweroff
```

1-7. VirtualBox 가상머신 네트워크 구성

가상머신 호스트 전용 어댑터 IP 대역 확인(VirtualBox 6.x 기준)

- VirtualBox > 파일 > 호스트 네트워크 관리자 > VirtualBox Host-Only Ethernet Adapter > IPv4 주소 확인

※ 일반적으로 처음 VirtualBox를 설치하면 IPv4 는 192.168.56.1 로 지정되어 있음

※ DHCP 서버에 사용함으로 체크되어 있다면 체크해제 할 것



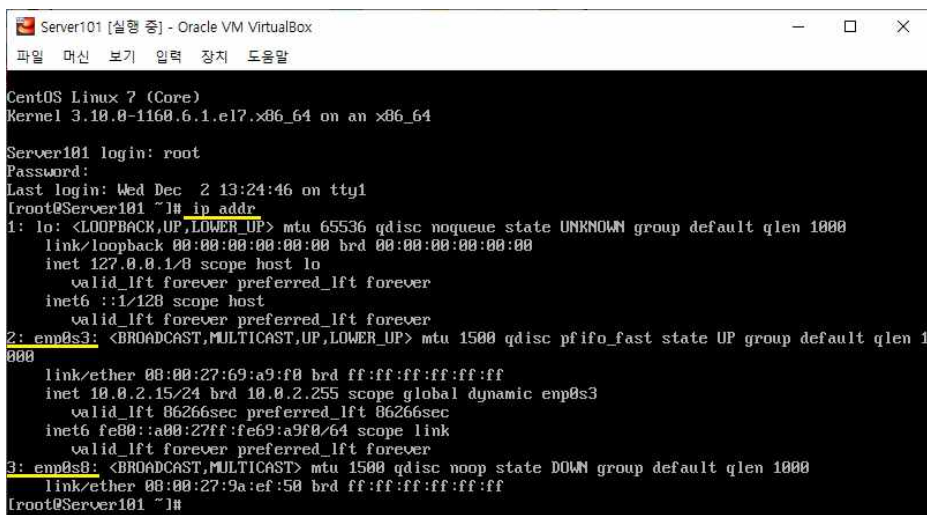
호스트 전용 어댑터 추가

- 가상머신 마우스오른쪽 버튼 > 설정 > 네트워크 > 어댑터 2 탭 > 다음에 연결됨 : 호스트 전용 어댑터 선택



1-8. 리눅스 부팅 후 연결 네트워크 확인

ip addr



※ enp0s3 : NAT 네트워크로 외부 인터넷 연결을 위한 네트워크 어댑터, 어댑터명 다를 수 있음

※ enp0s8 : 호스트 전용 어댑터로 고정 IP 연결을 위한 네트워크 어댑터, 어댑터명 다를 수 있음

1-9. 호스트 전용 어댑터 파일 생성 및 고정 IP설정

```
# cd /etc/sysconfig/network-scripts
# cp ifcfg-enp0s3 ifcfg-enp0s8
# vi ifcfg-enp0s8
```

```
4: BOOTPROTO="static"
12: NAME="enp0s8"           ← ip addr로 확인한 호스트 전용 어댑터 이름 입력
14: DEVICE="enp0s8"        ← ip addr로 확인한 호스트 전용 어댑터 이름 입력
16: IPADDR="192.168.xxx.xxx" ← IP 대역번호에 맞춰 입력, 끝자리는 101부터 시작
17: NETMASK="255.255.255.0"
```

※ GATEWAY 주소 넣으면 외부 인터넷 연결 안됨!

1-10. 네트워크 재시작 및 설정 정보 확인

```
# systemctl restart network (CentOS 7 경우)
# systemctl restart NetworkManager.service (CentOS 8 경우)
# ifconfig
```

```
Server101 [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

CentOS Linux 7 (Core)
Kernel 3.10.0-1160.6.1.el7.x86_64 on an x86_64

Server101 login: root
Password:
Last login: Wed Dec 2 13:26:48 on tty1
[root@Server101 ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe69:a9f0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:69:a9:f0 txqueuelen 1000 (Ethernet)
    RX packets 115 bytes 11920 (11.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 140 bytes 12304 (12.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe9a:ef50 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9a:ef:50 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 486 (486.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14 bytes 1016 (1016.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

1-11. ping 테스트

```
# ping www.google.com
```

```
[root@Server101 ~]# ping www.google.com
PING www.google.com (172.217.174.196) 56(84) bytes of data:
64 bytes from hkg07s34-in-f4.1e100.net (172.217.174.196): icmp_seq=1 ttl=110 time=46.9 ms
64 bytes from hkg07s34-in-f4.1e100.net (172.217.174.196): icmp_seq=2 ttl=110 time=46.9 ms
64 bytes from hkg07s34-in-f4.1e100.net (172.217.174.196): icmp_seq=3 ttl=110 time=46.0 ms
64 bytes from hkg07s34-in-f4.1e100.net (172.217.174.196): icmp_seq=4 ttl=110 time=46.8 ms
```

※ 테스트 종료는 Ctrl + c 입력

1-12. 재부팅 후 PUTTY 접속확인

```
# reboot
```

2. 웹 서버

1) Apache

- ① 웹(Web) 서버 운영을 위해 리눅스를 가장 많이 사용
- ② 리눅스 환경에서 가장 많이 사용하는 웹 서버 구축 패키지 APM(Apache, PHP, MySQL) 설치

☞ [실습하기] 웹 서비스 구축하기

1-1. Apache 설치

```
# yum install httpd
```

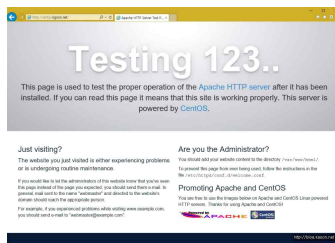
1-2. Apache 시작 및 상태 확인, 부팅 자동실행 등록

```
# systemctl start httpd
# systemctl status httpd
# systemctl enable httpd
```

1-3. 방화벽 설정/적용

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

1-4. <http://192.168.xxx.xxx> 확인



1-5. Apache 기본설정 및 홈 디렉터리 설정

```
#vi /etc/httpd/conf/httpd.conf
```

```
69: User nobody      ← 수정
70: Group nobody     ← 수정
98: ServerName 192.168.xxx.xxx:80  ← 주석해제 후 수정
122: DocumentRoot "/home/html"     ← 수정
134: <Directory "/home/html">      ← 수정
```

1-6. Apache 홈 디렉터리 생성 및 시작페이지 생성

```
# mkdir /home/html
```

```
# vi /home/html/index.html
```

```
<h1>Hello Apache!!!</h1> ← 입력
```

1-7. Apache 재시작 후 사이트 확인

```
# systemctl restart httpd
```

<http://192.168.xxx.xxx> 확인

2) PHP

- ① PHP는 웹 서버에서 사용되는 웹 프로그래밍 언어로 동적 HTML 개발에 사용
- ② 웹 프로그래밍 언어에는 JSP, ASP, PHP 등

☞ [실습하기] PHP 설치하기

2-1. PHP 설치

```
# yum install php php-mysqlnd
```

2-2. PHP 버전 확인

```
# php -v
```

2-2. Apache 재시작

```
# systemctl restart httpd
```

2-3. PHP 테스트 페이지 작성

```
#vi /home/html/hello.php
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>Hello PHP</title>
  </head>
  <body>
    <h3>Hello PHP!</h3>
    <?php
      $message = "Hello PHP!";
      echo "<h4>".$message."</h4>";
    ?>
  </body>
</html>
```

2-4. 테스트 페이지 접속 확인

<http://192.168.xxx.xxx/hello.php>

3) MariaDB

- CentOS 7 부터 기본 DBMS가 MySQL에서 MariaDB로 변경
- 기존 MySQL과 완벽 호환

☞ [실습하기] MariaDB 설치/설정/테스트하기

3-1. MariaDB 설치

```
# yum install mariadb mariadb-server
```

3-2. MariaDB 상태확인 및 시작

```
# systemctl status mariadb
# systemctl start mariadb
# systemctl enable mariadb
```

3-3. MariaDB 설정

```
# mysql_secure_installation
```

Enter current password for root (enter for none)?	최초 root 비번이 없으므로 Enter
Set root password?	y 입력 후 root 패스워드 설정(2회)
Remove anonymous users?	y 입력 후 익명접근 차단
Disallow root login remotely?	n 입력 후 root 접속 허용
Remove test database and access to it?	y 입력 후 test DB 삭제
Reload privilege tables now?	y 입력 후 현재까지 설정 적용

3-4. MariaDB 접속 및 기본 쿼리실습

```
# mysql -u root -p
mysql> show databases;
mysql> create database `mydb`;
mysql> use `mydb`;
mysql> show tables;

mysql> CREATE TABLE `USER` (
mysql>     `uid` VARCHAR(10) PRIMARY KEY,
mysql>     `name` VARCHAR(10),
mysql>     `age` INT,
mysql>     `hp` CHAR(13)
mysql> );
mysql> INSERT INTO `USER` VALUES ('a101', '홍길동', 21, '010-1234-1111');
mysql> SELECT * FROM `USER`;
```

3-5. MariaDB 외부접속용 root 계정 생성

```
mysql> create user 'root'@'%' identified by '비밀번호';
mysql> grant all privileges on *.* to 'root'@'%';
mysql> flush privileges;
mysql> exit
```

3-6. MariaDB 방화벽허용

```
# firewall-cmd --permanent --add-port=3306/tcp
# firewall-cmd --reload
```

3-7. HeidiSQL 접속확인

3. FTP 서버

1) FTP

- ① FTP(File Transfer Protocol)는 서버와 클라이언트간의 파일 전송 프로토콜
- ② CentOS8은 기본적으로 sftp가 설치되어 있어 따로 설치 할 필요가 없음

☞ [실습하기] FTP 접속하기

1-1. 파일질라 다운로드/설치

<https://filezilla-project.org/>

1-2. 파일질라 접속 확인

- 프로토콜 - SFTP 선택
- 호스트 - IP 주소 입력
- 로그인 유형 - 비밀번호 묻기 선택
- 사용자 - 아이디 입력



4. 기타

1) Java

- ① 리눅스 자바는 Oracle JDK, OpenJDK
- ② 자바 라이선스 문제 등으로 OpenJDK를 설치

☞ [실습하기] Java 설치/테스트하기

1-1. Java 설치 리스트 확인

```
# yum list java*
```

1-2. Java 설치

```
# yum install java-11*
```

1-3. 설치확인

```
# java -version
```

1-4. 예제 작성 및 컴파일 테스트

```
# vi Hello.java
# javac Hello.java
# java Hello
```

2) Python

- ① 파이썬은 1991년 네덜란드의 귀도 반 로섬에 의해 개발된 대화형 인터프리터 언어
- ② 파이썬은 플랫폼 독립적이며 대부분의 운영체제에서 동작

☞ [실습하기] Python 설치/테스트하기

2-1. python 버전 리스트 확인

```
# yum list python*
```

2-2. python 설치

```
# yum install python38*
```

2-3. 설치확인

```
# python3 -V
```

2-4. 예제 작성 및 테스트

```
# python3
>>> print('Hello World')
>>> hello = 'Hello Python'
>>> print(hello)
```

3) Tomcat

- ① 톰캣은 웹 서버와 연동하여 실행 할 수 있는 자바환경을 제공하여 JSP 실행환경을 제공
- ② 톰캣은 웹 애플리케이션 서버(WAS)로 다양한 기능을 컨테이너에 구현하여 기능을 수행

☞ [실습하기] Tomcat 설치/테스트하기
설치 매뉴얼 참고