

Linux

0.start

linux?

핀란드 헬싱키대학교의 학생이었던 리누스 토발즈가 처음 개발

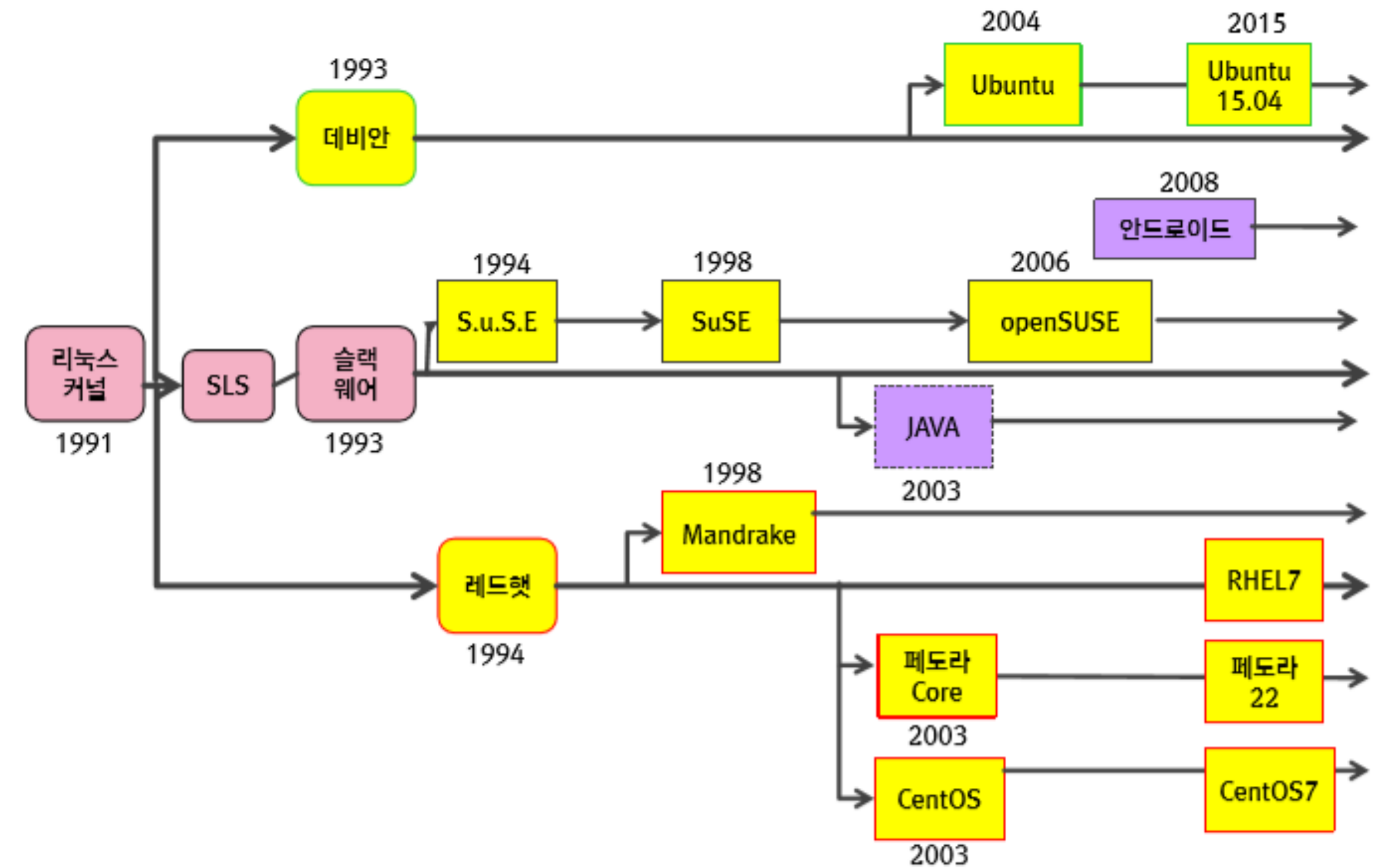
- MINIX(교육용 운영체제)를 참조
- GNU Project (GNU's Not Unix) → GNU/Linux
GNU GPL(GNU General Public License)
- distrowatch

0.start

배포판

RedHat 계열

- RHEL : 유료
- CentOS : RHEL을 무료로
- Fedora : 빠른 업데이트, 낮은 안정성



0.start

배포판

Debian 계열

- Debian : server, embedded에 주로 사용
안정성을 위해 업데이트 잘 안함
- Ubuntu : Debian에 편의성 추가

SUSE 계열

- Suse : 유럽에서 많이 사용

0.start

표준

POSIX (**P**ortable **O**perating **S**ystem **I**nterface) : OS 표준

LSB (Linux Standard Base) : Linux 표준

- POSIX 표준에 맞추고, 일부 확장
- file system 계층 등의 구조를 표준화

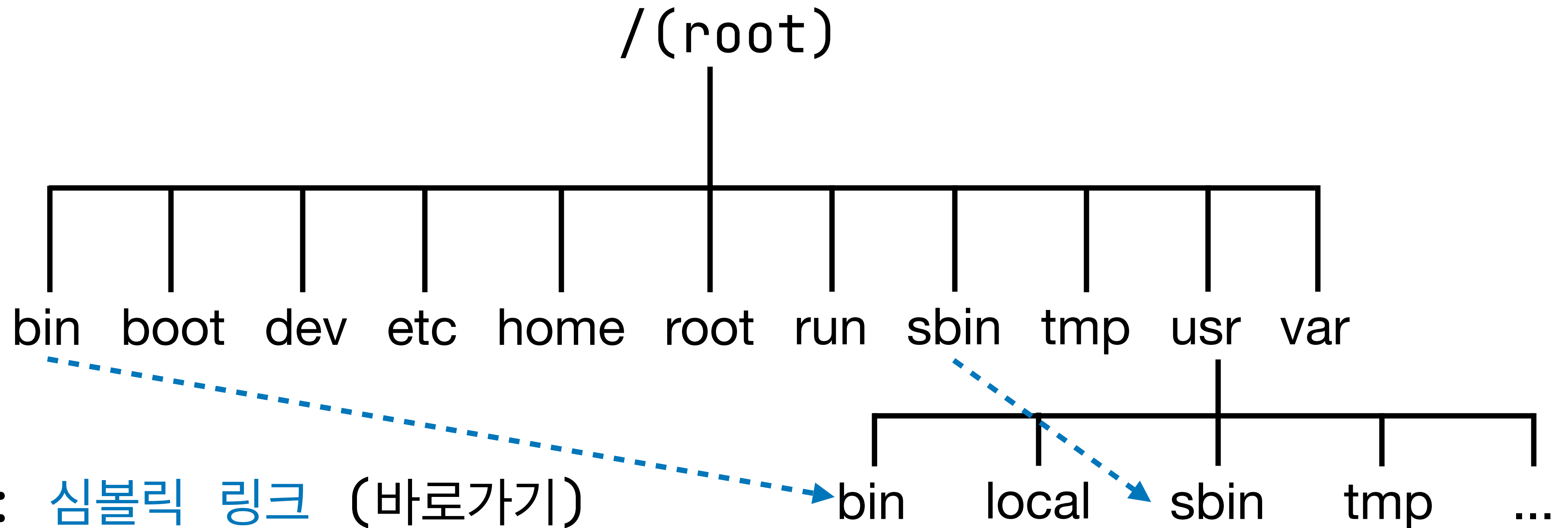
0.start

linux file system

- 단일 디렉토리 트리로 구성
- / : 최상단 디렉토리
- '/' : 디렉토리 구분자

0.start

linux file system



0.start

linux file system

/bin : binary (실행파일 등)

/sbin : system (or superuser) binary (관리자를 위한 실행파일 등)

/boot : booting과 관련된 파일 (bootloader, 커널 등)

/dev : device. 각종 장치 파일 등 (cd rom 등..)

/etc : 주요 설정. 시스템 환경 설정 등 (ex. /etc/hosts = dns 역할)

/home : 개인 사용자 디렉토리

0.start

linux file system

/root : 관리자의 home 디렉토리

/run : 실행 정보 파일

/tmp : temporary. 임시 파일 (잠시 만들어지고 사라지는 파일들)

/usr : 작은 root (/usr/local = 일반적인 오픈소스 기본 설치 디렉토리)

/var : variable (log파일 등)

1.user

사용자 관리

- 비밀번호가 없는 사용자는 사용 불가
- superuser로 로그인 하는 것은 지양
/etc/sudoers 파일을 이용하여 특정 사용자에게 관리 명령 부여 가능
/var/log/secure 파일에 sudo 명령 logging
- 한 사용자가 여러 개의 그룹에 포함 가능
기본 그룹 : 사용자 이름과 동일한 그룹으로 생성됨
보조 그룹 : 디렉토리 및 파일에 접근 권한 부여를 위해 사용

1.user

사용자 관리

su [사용자명] (switch user)

해당 사용자로 변경
(관리자의 비밀번호 필요)

sudo [명령어] (superuser do)

관리자 권한으로 실행
관리자가 sudo 사용하도록 허용해야 함

id [사용자명]

사용자 정보 확인

passwd [사용자명]

사용자 비밀번호 설정

1.user

사용자 관리

useradd [사용자명]	사용자 생성
usermod [option] [사용자명]	사용자 정보 수정
userdel [-r] [사용자명]	사용자 삭제

option	설 명
-g	기본 그룹 지정
-G -aG	보조 그룹 지정 / 추가
-d -dm	홈 디렉토리 지정 / 이동
-s	로그인 셸
-L / -U	사용 중지 / 중지 해제

2.file

파일 권한 관리

drwxrwxrwx : **owner**(생성한 사용자) **group**(그룹) **other**(타그룹)

d : directory

r : read

w : write

x : execute

ex) drwxr-xr-- : directory이고,
해당 파일을 생성한 사용자는 모든 권한을,
생성자와 같은 그룹의 사용자는 읽기와 실행 권한을,
타그룹 사용자는 읽기 권한을 가진다.

2.file

파일 권한 관리

chmod [옵션] [파일명]

ex) chmod g+w test1 : group에 쓰기 권한 추가
 chmod g-e test1 : group에 실행 권한 삭제

option	설 명
-u	owner
-g	group
-o	other
+	권한 부여
-	권한 삭제

2.file

파일 권한 관리

chmod [옵션] [파일명]

- 숫자로 한번에 추가 / 삭제 가능

r	:	4
w	:	2
x	:	1

ex) chmod **777** test1 : -**rwX****rwX****rwX**
chmod 700 test1 : -rwx-----
chmod 755 test1 : -rwxr-xr-x

2.file

파일 권한 관리

`chown [옵션] [owner][:group] [파일명]`

ex) `chown qclass test1`

`test1` 파일의 소유자를 `qclass`로 변경한다 (소유 그룹은 변경하지 않음)

`chwon :qclass test1`

`test1` 파일의 소유 그룹을 `qclass`로 변경한다 (소유자는 변경하지 않음)

`chown qclass:qclass test1`

`test1` 파일의 소유자를 `qclass`로, 소유 그룹을 `qlcass`로 변경한다

2. file

파일 관리

`cp file1 file2`

file 복사

`mv file1 file2`

file 이동 (변경)

`rm file1`

file 삭제

2.file

파일 관리

`mkdir dir1`

directory 생성

`cp -r dir1 dir2`

directory 복사

`mv dir1 dir2`

directory 이동 (변경)

`rm -r dir1`

directory 삭제

3.package

파일 관리

RPM : Redhat Package Manager

- *.rpm 확장자
- windows의 setup(installer)와 비슷한 역할
- 의존성(dependency) 문제로 인해 관련된 package들을 모두 직접 설치 필요

3.package

파일 관리

rpm -i[옵션] rpm-package : rpm-package 설치

rpm -u[옵션] rpm-package : rpm-package 업그레이드

option	설 명
-v	설치 시 message 출력
-h	설치 진행상황을 #으로 표시
--nodept	의존성 무시
--force	기존에 설치되어 있거나, 이전 버전으로 교체 시 사용
--test	실제로 설치하지 않고 설치 점검 수행

3.package

파일 관리

YUM : **Y**ellowdog **U**pdater **M**odified

- rpm 기반의 package tool
- 패키지를 분석하여 의존성 해결 (자동 설치)
- `/etc/yum.repos.d`에 패키지 저장소 파일 저장

3.package

파일 관리

yum [-y] install package-name package 설치

yum update package-name package 업데이트

yum upgrade package-name

yum [-y] erase package-name package 삭제

yum [-y] remove package-name

yum info package-name package 정보 확인

3.package

파일 관리

APT : Advanced Packaging Tool

- 데비안 계열의 패키지 인스톨러
- `/etc/apt/source.list` 에 패키지 인덱스 저장
- `apt` / `apt-get`은 큰 차이 없음 (사용 옵션 차이)
`apt-get`에서 자주 사용하는 옵션들을 묶어서 `apt`로 사용

3.package

파일 관리

apt install package-name

package 설치

apt update package-name

package 업데이트

apt upgrade package-name

apt remove package-name

package 삭제

apt purge package-name

설정파일들도 포함하여 삭제

apt show package-name

package 정보 확인

4.text editor

vi

VI : **VI**sual display editor

- 가벼운 text editor

- command mode
editor(insert) mode
last line mode

커서 이동, 명령어 입력

글 작성

하단에 ' :' 을 통해 명령어 입력

4.text editor

vim

VIM : VI iMproved

- vi 호환 text editor
- cui용 뿐만 아니라 gui도 있음 (대부분 cui 사용)
- vim script : vim에 통합된 스크립트형 언어

4. text editor

vim

vim [파일명]

:wq! 저장하고 종료

:q! 저장하지 않고 종료

version 1.1
April 1st, 06

vi / vim 단축키 모음

Esc
명령 모드

~ 대소문자 전환	! 외부 명령	@ 매크로 실행	# 이전 검색	\$ 줄 끝으로 이동	% 일치하는 줄로 찾기	^ 줄의 첫 글자	& :s 번복	* 다음 검색	(문장 시작) 문장 끝	_ 아래줄로 이동	+ 다음 줄
· 매크로 이동	1	2	3	4	5	6	7	8	9	0 줄의 처음	- 이전 줄	= 자동 들여쓰기
Q 실행 모드	W 다음 WORD	E 끝 WORD	R 수정 모드	T 뒤로 검색	Y 줄단위 복사	U 줄 단위 실행취소	I 줄 시작에서 삽입	O 행 위에 삽입	P 커서 이전에 붙여넣기	{ 문단 시작	}	문단 끝
q 매크로 기록	w 다음 단어	e 단어 끝	r 한 문자 교체	t 한 문자 검색	y 복사	u 실행취소	i 편집 모드	o 행 아래에 삽입	p 커서 이후에 붙여넣기	[기타]	기타
A 줄 끝에 덧붙이기	S 줄 삭제후 편집모드	D 줄 끝까지 삭제	F 뒤로 검색	G 파일끝/줄로 이동	H 화면 상단	J 줄 합치기	K 도움말	L 화면 하단	: ex 명령줄	" 레지스터 지정	열 이동	
a 덧붙이기	s 단어 삭제후 편집모드	d 한 줄 삭제	f 한 문자 찾기	g 확장 명령	h ←	j ↓	k ↑	l →	; t/T/f/F 명령 반복	' 매크로 이동	\ 사용 안함	
Z 종료	X 백스페이스	C 줄 끝까지 바꾸기	V 줄단위 비주얼모드	B 이전 WORD	N 이전 (찾기)	M 화면 가운데	< 3 내어쓰기	> 3 들여쓰기	? 찾기 (위로)			
Z 확장 명령	X 글자 삭제	c 바꾸기	v 비주얼 모드	b 이전 단어	n 다음 (찾기)	m 마크 설정	t/T/f/F 역순 검색	. 명령 반복	/ 찾기			

동작

명령

연산자

확장

q·

words: 구분자로 공백, 특수기호 모두 사용
WORDS: 구분자로 공백 문자만 사용

words: quux(foo, bar, baz);
WORDS: quux(foo, bar, baz);

동작

명령

연산자

확장

q·

words: 구분자로 공백, 특수기호 모두 사용
WORDS: 구분자로 공백 문자만 사용

words: quux(foo, bar, baz);
WORDS: quux(foo, bar, baz);

주요 명령행 명령 ('ex'):

:w (저장), :q (종료), :q! (저장하지 않고 종료)
:e f (파일 f 열기),
:s/x/y/g (파일 전체에서 'x' 를 'y' 로 교체),
:h (vim 도움말), :new (새 파일)

그외 중요한 명령들:
CTRL-R: 재실행 (vim),
CTRL-F/-B: 페이지 위로/아래로,
CTRL-E/-Y: 줄 스크롤 위로/아래로,
CTRL-V: 블록비주얼 모드 (vim 전용)

비주얼 모드:
커서를 움직여 지정한 범위에 연산자를 적용합니다. (vim 전용)

참고:

(1) 복사/붙여넣기/지우기 명령어를 사용하기 전에 "x"를 입력하여 레지스터(클립보드)를 지정하세요. (x는 a에서 z 또는 * 을 사용할 수 있음) {예: "ay\$ 를 입력하면 현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에 저장합니다.}

(2) 어떤 명령을 입력하기 전에 횟수를 지정하면, 횟수만큼 반복하게 됩니다.(예: 2p, d2w, 5l, d4j)

(3) 연속으로 입력하는 명령은 현재의 라인에 반영됩니다. 예시: dd(현재 라인 지우기), >>(들여쓰기)

(4) ZZ 는 저장후 종료, ZQ는 저장하지 않고 종료.

(5) zt : 커서가 위치한 곳을 제일위로 올리거나, zb : 바닥으로, zz : 가운데로

(6) gg : 파일의 처음으로(Vim 전용), gJ : 커서가 위치한 곳의 파일 열기(Vim 전용)

vi/vim 에 대한 더 많은 강좌나 팁을 얻으려면 www.viemu.com (ViEmu, MS 비주얼 스튜디오를 위한 vi/vim 에뮬레이션)을 방문하십시오.

5.archive

tar gzip bzip2

종 류	속 도	압 축 륜	확 장 자
tar	빠름	낮음	.tar
gzip	보통	보통	.gz
bzip2	느림	높음	.bz2

* gzip은 여러 개의 파일을 압축하지 못함 → 일반적으로 tar와 같이 사용
(*tar.gz)

5.archive

tar gzip bzip2

tar [옵션] [파일명]

c

x

f

v

z

j

tar 압축

tar 압축 해제

여러 개의 파일

처리 정보 출력

gzip 사용

bzip2 사용

5.archive

tar gzip bzip2

```
tar -xvzf jdk-8u212-linux-x64.tar.gz
```

x : 압축 해제
v : 처리 정보 출력
z : gzip(.gz) 사용
f : 여러 개의 파일

6.bonus

command

ls	목록 출력
ls -l	권한 소유자 그룹 파일크기 수정날짜 파일명
ls -a	숨김 파일(.*) 표시
ls -lh	파일크기를 단위(k, mb, ...)로 표시
cd [경로]	해당 경로로 이동
cd /	최상위(root) 디렉토리
cd ..	현재 디렉토리의 상위 디렉토리
cd -	현재 디렉토리로 이동 전 디렉토리
cd ~	현재 사용자의 홈 디렉토리
pwd	현재 디렉토리의 절대경로 확인

6.bonus

command

cat [파일명] 파일 내용 보기

명령 | 명령 명령의 결과를 다른 명령의 입력으로 사용

명령 < 파일명 파일로부터 입력 (키보드 입력 대신)

명령 > 파일명 파일로 새로 저장 (모니터 대신)

명령 >> 파일명 파일로 추가 저장

6.bonus

한글

```
yum -y update
```

```
yum install -y ibus ibus-hangul
```

```
reboot
```

```
# 설정 - 지역 및 언어 - 입력 소스 - 한국어(Hangul) 및 다른 언어 삭제
```

```
# shift + spacebar / 한영키 사용 가능
```

6.bonus

java

linux용 java (tar.gz) download → 압축해제

vim ~/.bashrc

JAVA PATH 추가

export JAVA_HOME=[자바 경로]

export PATH=\$JAVA_HOME/bin:\$PATH

export CLASSPATH=\$CLASSPATH:\$JAVA_HOME/jre/lib/ext:
\$JAVA_HOME/lib/tools.jar

source ~/.bashrc

6.bonus

tomcat

tomcat download → 압축해제

```
vim /qclass/tomcat/conf/server.xml  
port="8080"          # 8787 로 변경
```

* JAVA_HOME 이 등록되어 있지 않으면, 실행 불가