

# Linux.Editor.Toy

SPARCS 15 고지훈(mango)

[sparcs.org/~mango/linuxEDITORtoy.pdf](http://sparcs.org/~mango/linuxEDITORtoy.pdf)

Linux.Editor.Toy

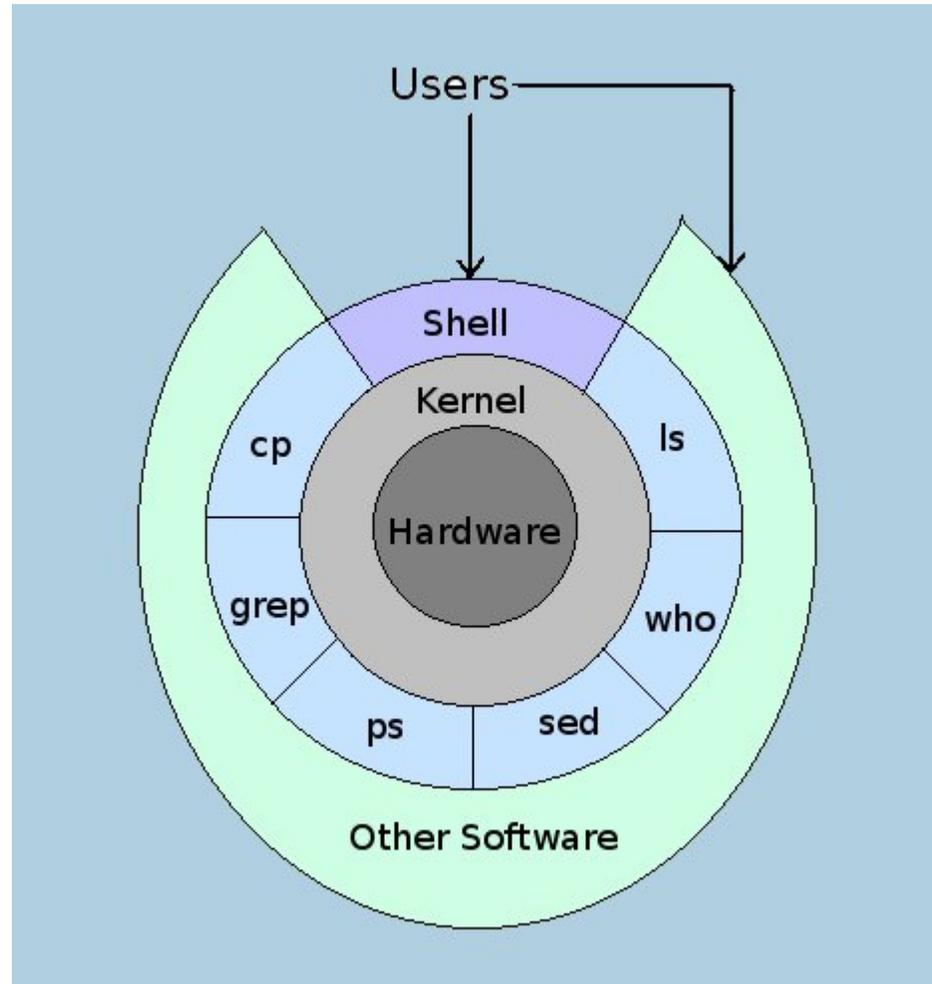
# LINUX

- 하나의 운영체제 (like Windows / macOS)
- 리누스 토발즈가 1991년에 개발함
- 다양한 종류가 있고, LINUX 커널을 쓰는 것들을 모두 통칭 (Ubuntu / Debian / Fedora / RedHat / Arch Linux ...)
- 다른 운영체제보다 가볍고 안정적이어서 서버로 많이 사용
- 대부분 무료로 배포하지만 유료 배포인 것도 있다.

# Operating System

- ~~운영체제 및 실험 (CS330)~~
- 하드웨어와 유저간의 인터페이스
- 소프트웨어를 실행하기 위한 플랫폼을 제공한다
- 시스템과 하드웨어를 관리하는 역할을 한다.

# Operating System \_Structure



# Operating System \_ Structure

- 커널 (kernel)
  - 씨앗, 핵에 비유가 가능
  - 운영체제의 핵심 부분
  - Process, Memory, File System, Device, Network, etc...
  - 등등 많은 것을 관리함
  - 지속적으로 이용되어야 하므로, 보호된 메모리를 이용한다.
- 셸 (shell)
  - (커널을 둘러싼) 껍질, 껍데기로 비유가 가능
  - 사용자와 운영체제 사이에 위치
  - 사용자가 명령어를 입력 -> 명령어를 기계어로 바꿈 -> 커널로 기계어를 전달

# Linux \_Directory \_Structure

- When you type "ls /" ...

```
rnango@sparcsMango:~$ ls /  
/bin /boot /dev /etc /home /media /root ...
```

```
rnango@sparcsMango:/$ cd /home/mango
```

```
rnango@sparcsMango:~$ _
```

# Linux \_Directory \_Structure

- /bin: 기본적인 명령어가 저장된 디렉토리
- 예를 들어, mv, cp, rm 등등..
- /usr : 일반 사용자들이 사용하는 명령어를 저장함
- /boot: 리눅스 부트로더가 존재하는 디렉토리
- /dev: 시스템 디바이스 파일 (하드디스크 장치파일, CD-ROM 장치파일 등등...) (Do you know /dev/sda?)
- /etc: 시스템의 거의 모든 설정파일이 존재
- ex) /etc/ssh/... : ssh 서비스 설정 파일
- /home: 사용자의 홈 디렉토리
- /lib: 라이브러리 파일 (like libc)
- /media, /mnt: 마운트 포인트 (탈부착이 가능한 장치들)
- /root: 루트 사용자의 홈 디렉토리
- /tmp: 공용 디렉토리, 무려 Permission 777

# Linux \_Commands

- 계정 관련 – passwd (비밀번호 변경)
- 매뉴얼 – man [검색할 것] (정말로 매뉴얼)
- 경로 관련 – pwd (현재 작업중인 디렉토리)
- , cd [타겟 폴더] (Change Directory)
- , ls (LiSt) [-l(long) a(all) R(recursive) d(directory)]

# Linux \_Commands

- cat : 파일내용 읽기, 합치기
  - cat [filename] : 파일 읽기
  - cat [a] [b] > [c] : a와 b의 파일 내용을 합친 것을 c라는 파일에 저장
  - cat -n [filename] : 줄 번호가 나오게 출력
- grep (global/regular/expression/print) : 텍스트 검색 기능
  - grep [option] [keyword] [file]
  - -i : 대소문자 구별 하지 않음, -w: 독립적인 단어로 검색
- vi/vim : vim을 실행 (자세한 건 뒤에서)

# Linux \_Commands

- ssh: 원격 컴퓨터에 접속하여 작업
  - ssh [user\_id@computer\_name]
- scp: Secure Copy, 암호화된 명령을 통해 파일 전송
  - scp [file] [user@computer:dictionary]
- alias: 일종의 매크로.
  - ex) alias myDocker= "sudo docker attach mango\_docker"
  - 이제 myDocker라고 입력하면 오른쪽의 명령이 실행됩니다.

# Pipe & Redirection

- Pipe | : [command1] | [command2]
  - [command1]의 output을 [command2]의 input으로 사용
- Redirection (<. >)
  - EX) ./myAlgorithm < input.txt > output.txt

Linux.Editor.Toy

# Text Editor

- 텍스트 편집기
- 텍스트를 편집하는 툴 ( ex) 메모장, 워드패드 in Windows)
- Linux에서는 vi, vim, emacs, nano, gedit 등을 사용한다.
- 여기서는 가장 많이 사용하는 것으로 알려진 vim에 대해서만 다루겠습니다.
- Emacs에 대해 배우시고 싶은 분은 최정운(coearth) 회원의 emacs seminar를 참고하세요.

# VIM

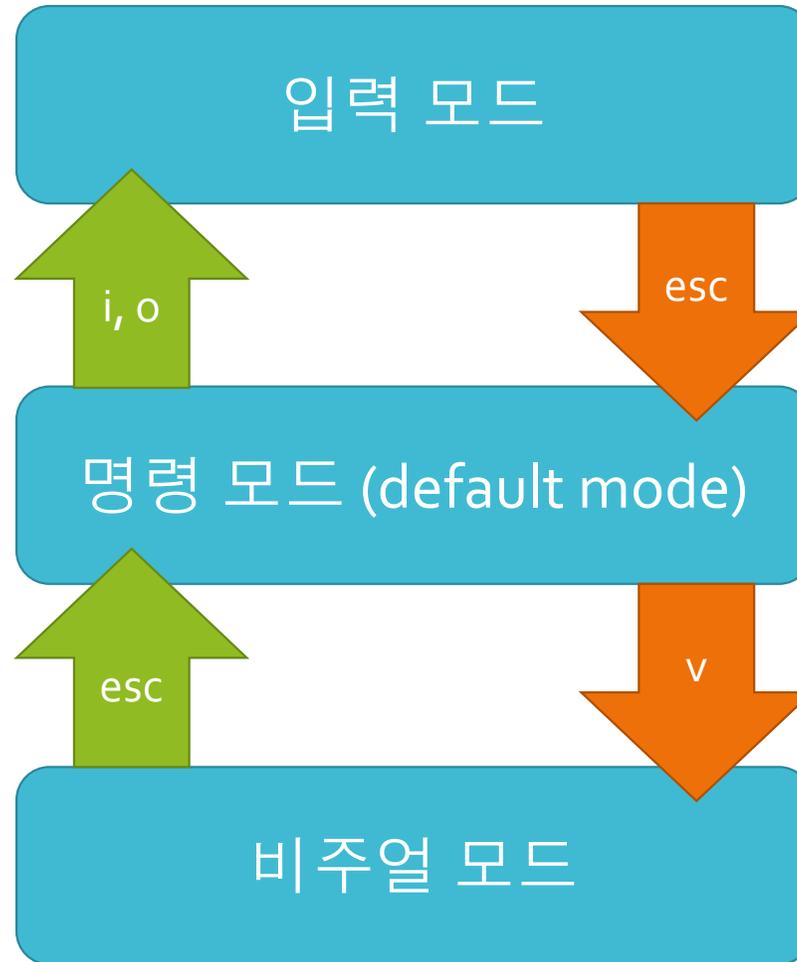
- VI Improved
- vi를 개량하여 좀 더 좋은 editor를 만들었음.
- 그러니 여러분은 vi 대신 vim만 익히셔도 됩니다.
  
- 세 가지 모드: 비주얼 모드, 명령 모드, 입력 모드

# Vim

## \_Three Modes

- 비주얼 모드 (visual mode) : 텍스트를 선택할 수 있음
- 명령 모드 (normal mode) : 각종 명령을 할 수 있음
- 입력 모드 (insert mode) : 새로운 글자를 입력할 수 있음

# Vim \_Three Modes



# Vim \_Cheat Sheet

version 1.1  
April 1st, 06

## vi / vim graphical cheat sheet

Esc

normal mode

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	( begin sentence	) end sentence	"soft" bol down	+ next line
\ goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	w next word	e end word	r replace char	t 'till	y yank <sup>1,3</sup>	u undo	i insert mode	o open below	p paste after	[ misc	]	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	. ex cmd line	!' reg. spec <sup>1</sup>	bol/ goto col	\ not used!
a append	s subst char	d delete <sup>1,3</sup>	f find char	g extra <sup>6</sup>	h ←	j ↓	k ↑	l →	. repeat ; t/T/t/F	' goto mk. bol		
Z quit <sup>4</sup>	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent <sup>3</sup>	> indent <sup>3</sup>	? find (rev.)			
Z extra <sup>5</sup>	x delete char	c change	v visual mode	b prev word	n next (find)	m set mark	, reverse ; t/T/t/F	. repeat cmd	/ find			

- motion** moves the cursor, or defines the range for an operator
  - command** direct action command, if **red**, it enters insert mode
  - operator** requires a motion afterwards, operates between cursor & destination
  - extra** special functions, requires extra input
  - Q** commands with a dot need a char argument afterwards
- bol = beginning of line, eol = end of line, mk = mark, yank = copy
- words: `quux(foo, bar, baz)`  
 WORDs: `quux(foo bar baz)`

**Main command line commands ('ex'):**  
 :w (save), :q (quit), :q! (quit w/o saving)  
 :e f (open file f),  
 :%s/x/y/g (replace 'x' by 'y' filewide),  
 :h (help in vim), :new (new file in vim),

**Other important commands:**  
 CTRL-R: redo (vim),  
 CTRL-F/-B: page up/down,  
 CTRL-E/-Y: scroll line up/down,  
 CTRL-V: block-visual mode (vim only)

**Visual mode:**  
 Move around and type operator to act on selected region (vim only)

- Notes:**
- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,\*) (e.g.: "ay\$ to copy rest of line to reg 'a')
  - (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
  - (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
  - (4) ZZ to save & quit, ZQ to quit w/o saving
  - (5) zt: scroll cursor to top, zb: bottom, zz: center
  - (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to [www.viemu.com](http://www.viemu.com) - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

주의! 그림을 보면 알 수 있듯이 명령어는 대소문자를 구별

# Vim \_Usage

- 기능이 너무 많으니 중요한 것만 다룹니다.
- 자세한 것은 전 슬라이드 그림 참조.

# Vim \_Usage

- 입력 모드
- i : insert – 현재 위치부터 입력
- a : append – 현재 위치 다음부터 입력
- I : 현재 줄의 맨 앞에서부터 입력
- A : 현재 줄의 맨 뒤에서부터 입력
- o : 현재 줄 아래에 줄을 만들고 입력
- O : 현재 줄 위에 줄을 만들고 입력

# Vim \_Usage

- 명령 모드 : 편집에 필요한 각종 명령이 가능한 상태
- gg : 파일 맨 처음으로 이동
- [줄 번호] + gg ( 또는 :[줄 번호] + <Enter> ) : 해당 줄로 이동
- G : 파일 맨 마지막으로 이동
  
- x, dl : 현재 위치의 글자 삭제
- X, dh : 현재 위치 바로 앞의 글자
- D, d\$ : 현재 위치부터 줄의 끝까지 삭제
- dd : 커서가 있는 줄을 삭제 (: [줄 번호] + <Enter> )

# Vim \_Usage

- Ctrl과 함께라면 스크롤이 가능하다.
- +F : 한 화면 다음으로 스크롤
- +B : 한 화면 전으로 스크롤
- +D : 반 화면 다음으로 스크롤
- +U : 반 화면 전으로 스크롤
- +E : 한 줄 다음으로 스크롤
- +Y : 한 줄 전으로 스크롤
- z + (<Enter> or . or - ) : 각각 현재 줄을 화면의 맨 위, 가운데, 맨 아래로 스크롤

# Vim \_Usage

- :w : 저장
  - :w [new filename] : 다른 이름으로 저장
  - :q : 나가기
  - :wq or :x or ZZ : 저장하고 나가기
  - :q! : 저장하지 않고 나가기
  - :help [명령어] : [명령어]에 대한 도움말
- 
- /[string] : [string] 검색
  - /[string]Wc : 대소문자 구분하지 않고 검색
  - ?[string] : 밑에서부터 검색
  - :s/[old]/[new] : 찾아 바꾸기 (old -> new)

# Vim \_Usage

- 비주얼 모드
- v : 글자 단위로 블록 지정
- V : 줄 단위로 블록 지정
  
- 블록을 다 지정한 후에는...
- x, d, c : 잘라내기 (클립보드에 저장, like Ctrl + X)
- y : 복사 (yank, like Ctrl + C)
  
- yy : 현재 줄을 복사
- p : 현재 위치의 다음에 붙여넣기
- P : 현재 위치에 붙여넣기

# Vim \_Setting

- vim을 원하는 대로 커스터마이징 할 수 있다.
- :set nu : 줄 번호 표시
- :set nonu : 줄 번호 표시 X
- :set ai : auto indent
- :set noai : 자동 들여쓰기 해제
- :set mouse=a : 마우스로 커서 선택
- :set mouse=c : 마우스로 커서 선택 X
- :set ts=[수] : 탭의 크기 설정
- :highlight : 하이라이트 설정
- :nohl : 검색 후 하이라이트를 지움
- colorscheme desert : 컬러 테마를 desert로 바꿈

# Vim \_Setting \_\_\_.vimrc

- setting은 일회용이면 매우 불편할 것이다.
- 그래서 한 번 설정하면 계속 사용할 수 있는 방법이 있다!
- ~/.vimrc에서 전 슬라이드의 설정을 적용할 수 있다!
  
- 자세한 것은 구글링을 해보거나 여러 유저들의 .vimrc 관련 github를 뒤져보는 것도 좋은 방법이다.

Linux.Editor.Toy

# Toy

- 장난감?
- 어떤 작업들을 쉽고 편하게 할 수 있도록 만든 툴

```
mango@sparcs: /home
mango@sparcs:/home$ ls sparcs/bin
addsparcs          edalias.py        onel
aduser             eeeg             photo-make-thumbnails
apache2_log_splitter.py  iDEA_backup     post-review
apache2_split_log_permitter.py ip_toy          quota-report
attd2csv          ip_toy_old       README
attd2csv.py       ip_toy_old_old  seminar
backup            mailpatcher      setuid_wrapper.c
change_username   mailpatcher.py   siva.csv
change_username.py Makefile          sivatemp
daemon.py         mkml             siv.csv
daemon.pyc        mkml_backup     sugang
delsparcs        mkml.c          svn-commit.tmp
document-indexer.pl myapache.log     test
edalias           naeja           test.ga
edalias_gather    nugu           test.na
edalias_gather.py nugu-old       unused
edalias_init      nugusave       wakeup
edalias_init.sh   nugusave.sh    wakeup.py
mango@sparcs:/home$
```

# nugu

- SPARCS 주소록 관리 도구
- usage: `nugu [-h] [-l] [-s] [-e] [-r] [target]`
- SPARCS nugu v0.6.0 by samjo, netj, daybreaker
- positional arguments:
  - target        검색할 ID 또는 이름입니다.
- optional arguments:
  - -h, --help    show this help message and exit
  - -l, --list    전체 사용자의 목록을 출력합니다.
  - -s, --search ID/이름에 검색어가 들어있는 사용자를 찾습니다.
  - -e, --edit    사용자의 정보를 수정합니다.
  - -r, --remove 사용자를 삭제합니다. (sudo 필요)
- ex) `nugu -l | grep "mango"`

edalias

- 메일 구독 관련 장난감
- 구독하고 싶다 ? 주석 해제 : 주석 처리 (#)

# mailpatcher

- 이름 옆에 괄호 열고 아이디를 써 주는 장난감
- Input 파일을 받고 이름을 찾아 그 옆에 괄호로 아이디를 써 준다!

# seminar

- 세미나를 올릴 수 있는 장난감이고 아직도 작동하지만...
- 웹에 직접 올릴 수 있는 버튼이 생겨서 지금은 거의 사용하지 않는 것 같습니다... (쓰는 사람이 있을 수도)

geogy

- 도메인을 연결해준다.
- 하지만, 지금은 잘 안되는 듯 하다...
- 관심 있다면 예전 세미나 자료를 뒤져서 검색!

Any  
Question?

```
mango@bit~$ cat footer.txt
```

```
Special Thanks to Chocho (SPARCS 13),  
Egg(SPARCS 13),  
Apple(SPARCS 12),  
Coearth(SPARCS 14)
```