



리눅스 기초

9월 15일

목차

- 리눅스가 무엇일까?
 - 간략한 유닉스, 리눅스 역사
 - 운영체제 구성요소, 리눅스 배포판
 - 우리는 왜 리눅스를 쓸까?
- 리눅스 실습
 - 셸 명령어 기본
 - 패키지 관리자(apt), 텍스트 편집기(vim), 셸 스크립트
 - 셸 명령어 심화
 - ssh
 - cron

역사



리누스 토발즈

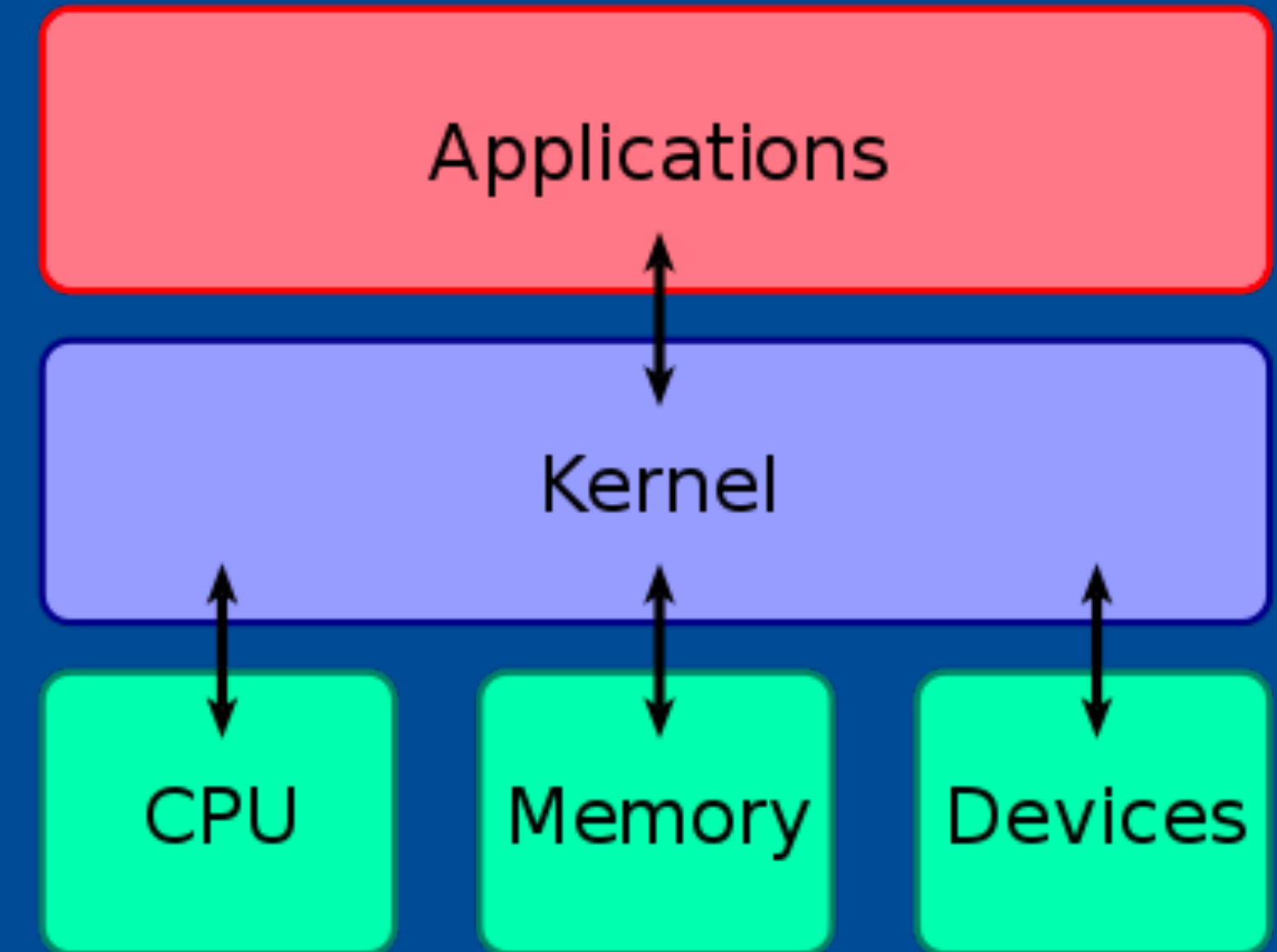
- 1991년에 개발된 신생아 OS
- 유닉스에 비해 역사는 길지 않지만 오픈소스의 힘을 받아 빠르게 성장

리눅스 특징

- 멀티 유저 : 하나의 시스템에 다수의 사용자들이 동시 접속하여 사용 가능
- 멀티태스킹 : 접속자들이 CPU같은 공용 자원을 나누어 프로그램을 실행할 수 있음
- 오픈소스 -> 무료 & 빠른 버그 수정
- 다양한 배포판 존재 (서버용, 개발용, PC용)
- 가상 메모리 사용 (램 부족하면 하드 사용)

커널이란?

- 운영 체제의 핵심
- 하드웨어 자원을 어플리케이션에 할당 해주는 부분



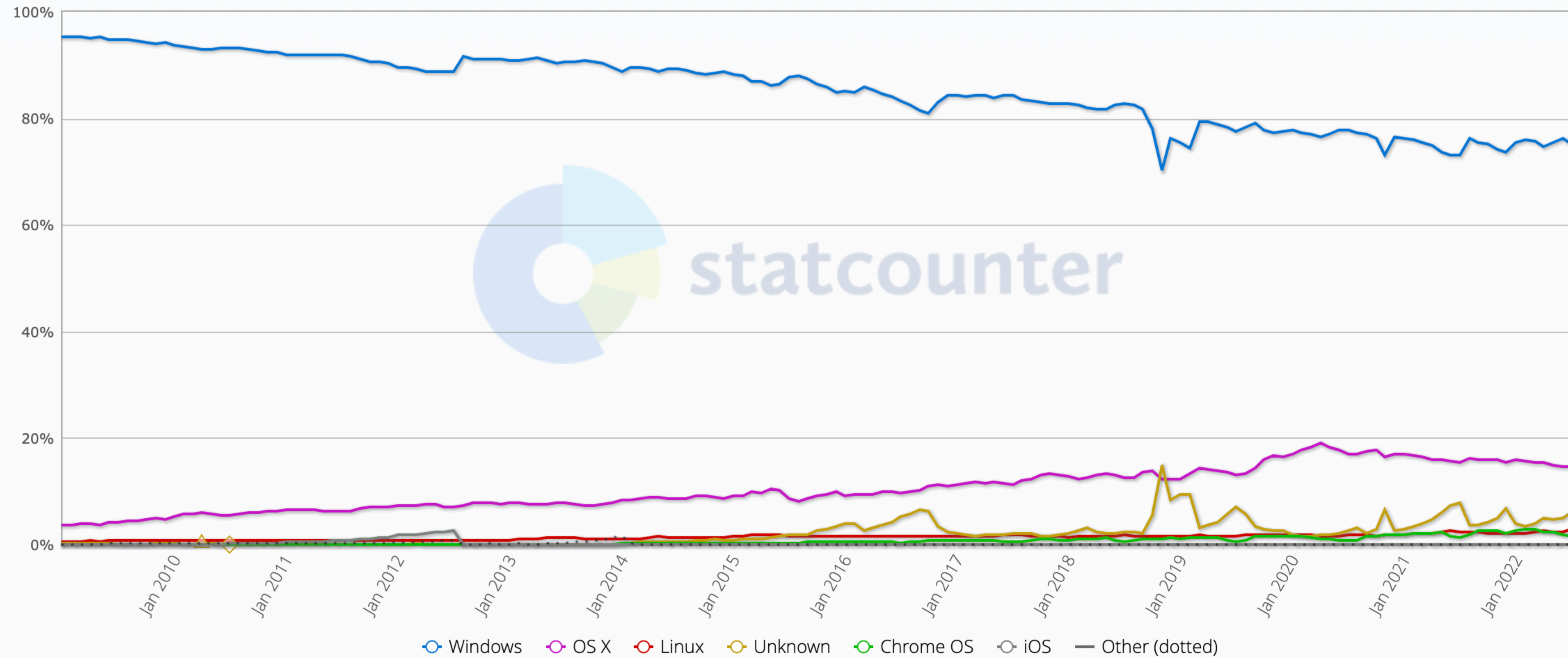
리눅스 배포판

- 리눅스 커널을 기반을 둔 오픈소스 유닉스 계열 운영체제
- ex) 데비안, 페도라, 우분투 / 레드햇 엔터프라이즈 리눅스(RHEL) 등

우리는 왜 리눅스를 공부해야할까

Desktop Operating System Market Share Worldwide

Jan 2009 - Aug 2022

[Edit Chart Data](#)[Save Chart Image \(.png\)](#)[Download Data \(.csv\)](#)[Embed HTML](#)

```
<div id="desktop-os_combined-ww-monthly-200901-202208" width="600
```


우리는 왜 리눅스를 사용해야할까

Linux Statistics (Editor's Choice)

- All of the world's 500 supercomputers run on Linux. (It's FOSS)
- 85% of all smartphones are powered by Linux. (Hayden James)
- One of the leading contributors to Linux Kernel is Intel. (It's FOSS)
- Out of 25 top websites in the world, only two aren't powered by Linux. (ZDNet)
- There are over 600 Linux distros available. (Tecmint)
- The US boasts the most significant number of Ubuntu users. (Ubuntu)
- Ubuntu powers nearly 40% of all websites with a known system. (W3Techs)

번외) 왜 개발자들은 Mac을 쓸까?

- 유닉스 기반 OS (명령어 비슷)
- 편리한 환경 세팅 (brew)
- 안정적, 보안성
- 성능
- 멋있어서

리눅스 기초

- 셸
- 기본 명령어
- 패키지 매니저(apt)
- 텍스트 편집기(vim)
- 셸 스크립트

```
Last login: Tue Sep  6 23:30:19 on ttys000
→ ~ ls
Applications Downloads Movies Public ibc-rs seminar
Desktop      IdeaProjects Music  code  install
Documents    Library   Pictures go     key
→ ~ █
```

셸 기본 명령어

- ls: 파일 리스트 출력
- cd: 디렉토리 이동
- mkdir: 디렉토리 생성
- cat: 파일 내용 조회
- rm: 파일 삭제
- echo: 입력을 표준출력으로 출력
- cp: 파일 복사
- mv: 파일 이름 변경, 이동
- chmod: 권한 설정
- 명령어 메뉴얼 보기: man 명령어
- 구글에 셸 명령어 검색해서 사용하자

ex00

```
→ ex00 ls -al
total 0
drwxr-xr-x  3 mihnhyuk  staff   96  9  6 23:51 .
drwxr-xr-x  5 mihnhyuk  staff  160  9  6 23:50 ..
drwxr-xr-x  2 mihnhyuk  staff   64  9  6 23:51 folder
→ ex00 █
```

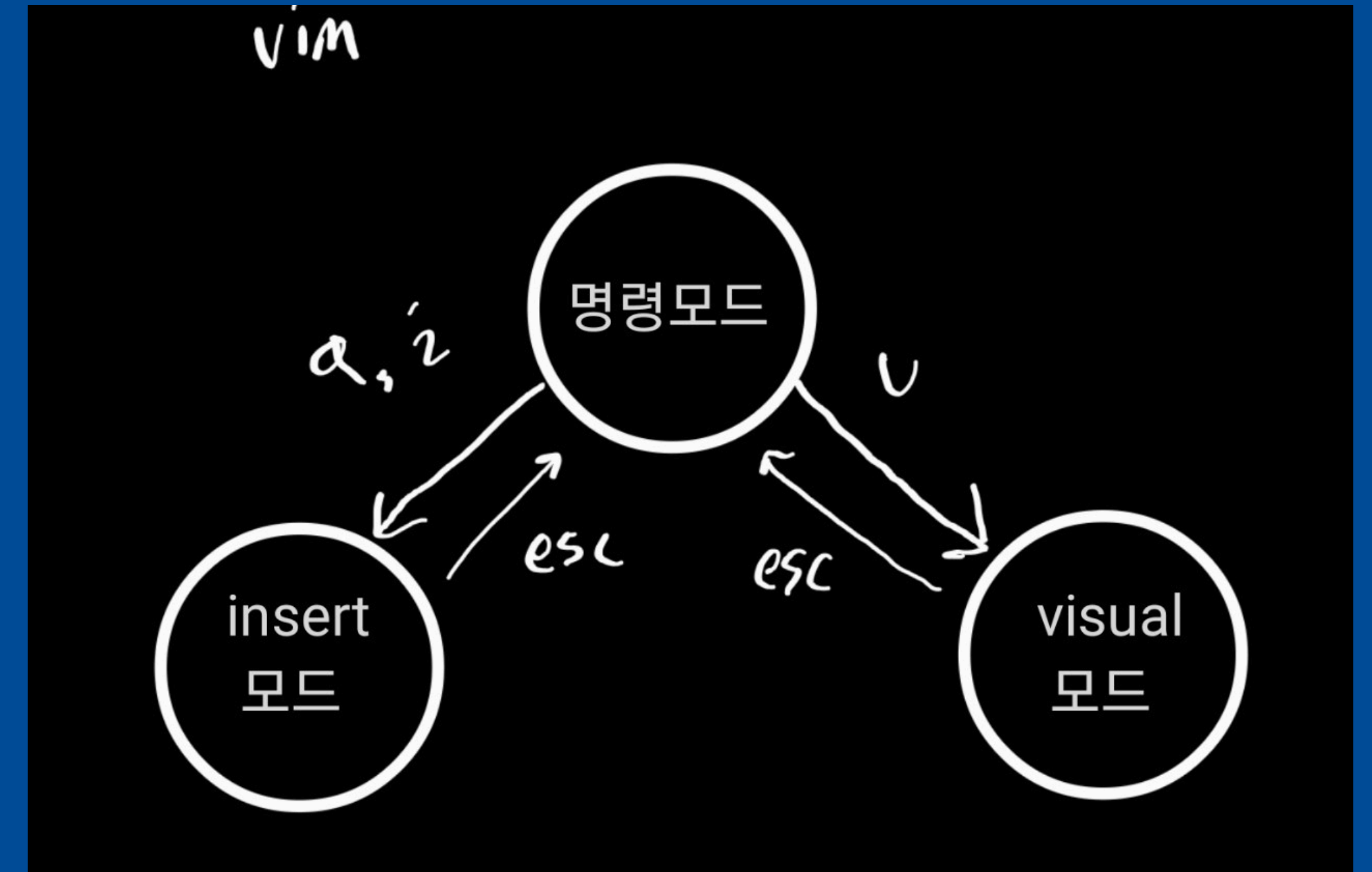
- ex00이라는 폴더를 만들고, 폴더 안에 위와 같이 파일을 만들어 보세요

패키지 매니저

- 프로그램의 설치, 업그레이드, 제거 등의 과정을 자동화 해주는 프로그램
- sudo: 슈퍼 유저, 최고 권한으로 실행
- apt

텍스트 편집기(vim)

- cli환경에 적합한 텍스트 편집기
- 비슷한 프로그램으로 nano, emacs 등이 있음
- 명령모드에서 명령 내리기 -> ':' 입력후 명령어 입력
 - 명령모드에서 저장: w, 종료: q
- 복사, 찾기, 등 다양한 기능 지원
- 각종 플러그인 사용 가능



ex01

- ex01 이라는 폴더를 만들고 안에 hello clug! 라는 내용의 hello.txt 파일을 작성해보자

```
→ ex01 cat hello.txt  
hello clug!  
→ ex01 █
```

스크립트 파일 쓰기

- .sh 파일은 스크립트 파일이다
- 스크립트 파일을 실행시키면 안에 써놓은 명령어를 실행 시킬 수 있다.
- 첫줄에 `#!/bin/bash` 쉘(zsh, bash, csh 등)
- 이후에 명령어 작성

ex02

- ex01에서 만든 hello.txt 파일을 ex02폴더 안에 복사해 넣고 내용을 출력하는 스크립트 파일 script.sh를 작성해보자.

ex03

- ex03에 다음과 같은 스크립트 파일을 작성하고 실행해보고 생성된 파일(폴더)를 다시 지워 보자

```
→ ex03 cat test.sh
#!/bin/bash

for num in 1 2 3 4
do
    mkdir $num
    cd $num
done

→ ex03
```

리눅스 심화

- 심화 명령어
- ssh
- 환경변수
- cron
- 유저, 그룹

리눅스 심화 명령어

- grep
- find
- tar
- diff
- 파이프 라인 (|)
- >> , >

grep

- 입력에서 일치하는 패턴의 라인을 찾아줌

```
→ seminar ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
anpi1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether b2:e5:0d:84:2c:05
    inet6 fe80::b0e5:dff:fe84:2c05%anpi1 prefixlen 64 scopeid 0x4
    nd6 options=201<PERFORMNUD,DAD>
    media: none
    status: inactive
anpi0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether b2:e5:0d:84:2c:04
    inet6 fe80::b0e5:dff:fe84:2c04%anpi0 prefixlen 64 scopeid 0x5
    nd6 options=201<PERFORMNUD,DAD>
    media: none
    status: inactive
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether b2:e5:0d:84:2c:e4
    nd6 options=201<PERFORMNUD,DAD>
    media: none
    status: inactive
en4: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether b2:e5:0d:84:2c:e5
    nd6 options=201<PERFORMNUD,DAD>
    media: none
    status: inactive
```

```
→ seminar ifconfig | grep "ether"
ether b2:e5:0d:84:2c:05
ether b2:e5:0d:84:2c:04
ether b2:e5:0d:84:2c:e4
ether b2:e5:0d:84:2c:e5
ether 36:0c:b6:4c:13:80
ether 36:0c:b6:4c:13:84
ether a2:65:7c:d1:7a:47
ether 80:65:7c:d1:7a:47
ether 36:0c:b6:4c:13:80
ether 5e:7b:5d:c0:5a:1d
ether 5e:7b:5d:c0:5a:1d
ether 00:e0:4c:cc:02:f6
```

find

- 파일 탐색
- find [옵션] [경로] [표현식]

```
→ Linux git:(master) ✕ find ~/code/compiler_midterm -name "*.c"  
/Users/mihnnhyuk/code/compiler_midterm/TestFile3.c  
/Users/mihnnhyuk/code/compiler_midterm/TestFile.c  
/Users/mihnnhyuk/code/compiler_midterm/TestFile2.c
```

파이프라인 (|)

- 이전 명령어의 출력을 다음 명령어의 표준입력으로 연결
- xargs : 표준 입력을 실행 명령어로 바꿔줌

```
→ example git:(master) ✕ cat a b  
hi  
hello
```

```
→ example git:(master) ✕ cat a b | grep "hi"  
hi
```

```
→ example git:(master) ✕ ls | cat  
a  
b
```

```
→ example git:(master) ✕ ls | xargs cat  
hi  
hello
```

>, >>

- 명령어의 출력을 파일로 써줌
- > 는 write, overwrite
- >>는 append

```
→ example git:(master) ✕ echo "Hello world" >> a
→ example git:(master) ✕ cat a
Hello world
→ example git:(master) ✕ echo "Hello world" >> a
→ example git:(master) ✕ cat a
Hello world
Hello world
```

ex04

- 현재 디렉토리의 c 파일 중에서 clug가 들어간 파일을 찾아주는 clug.sh 작성해보자

```
→ ex04 git:(master) ✕ sudo bash clug.sh  
./iloveclug.c  
./clug.c
```


tar

- 파일을 하나로 묶거나 풀때 쓰는 명령어
- man tar

```
tar [OPTION...] [FILE]...
  -f      : 대상 tar 아카이브 지정. (기본 옵션)
  -c      : tar 아카이브 생성. 기존 아카이브 덮어 쓰기. (파일 묶을 때 사용)
  -x      : tar 아카이브에서 파일 추출. (파일 풀 때 사용)
  -v      : 처리되는 과정(파일 정보)을 자세하게 나열.
  -z      : gzip 압축 적용 옵션.
  -j      : bzip2 압축 적용 옵션.
  -t      : tar 아카이브에 포함된 내용 확인.
  -C      : 대상 디렉토리 경로 지정.
  -A      : 지정된 파일을 tar 아카이브에 추가.
  -d      : tar 아카이브와 파일 시스템 간 차이점 검색.
  -r      : tar 아카이브의 마지막에 파일들 추가.
  -u      : tar 아카이브의 마지막에 파일들 추가.
  -k      : tar 아카이브 추출 시, 기존 파일 유지.
  -U      : tar 아카이브 추출 전, 기존 파일 삭제.
  -w      : 모든 진행 과정에 대해 확인 요청. (interactive)
  -e      : 첫 번째 에러 발생 시 중지.
```


diff

- 두 파일 차이 비교
- git 사용시 충돌 해결에 유용

patch

- man patch !

```
→ seminar cat a
hello world!
I love CLUG!
→ seminar cat b
Hello CLUG!
I love CLUG
→ seminar diff a b
1,2c1,2
< hello world!
< I love CLUG!
---
> Hello CLUG!
> I love CLUG
.
```

ex05 -



```
→ source git:(master) x cat a
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.
It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.
It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages,
and more recently with desktop publishing software like Aldus PageMaker including jersions of Lorem Ipsum.
```

```
→ source git:(master) x diff a b >> sw.diff
```

- b를 만들어보자

ssh

- secure shell
- 원격 호스트에 접속하기 위해 사용되는 보안 프로토콜
- 암호화 기술 (공개키 - 비공개키)

설치되지 않은 경우 apt install openssh-server로 설치(클라이언트는 openssh-client)

ssh <계정이름>@<서버주소> -p 포트번호

ssh서비스 실행과 상태 확인 : systemctl [start/status/restart ...] sshd