오픈소스 SW 입문

2w Linux, Shell Script

2023-0504, 강의실(과학관213)

강사 : 박노헌

https://github.com/nparkcourage/2023-kau-0504

한국항공대학교

잠깐 쉬어가기

GNU Project Logo

GNU [그누]

gnu [누]

목차

7교시 15:00	◆ **Linux 개요** • OS 개요 • Linux의 역사 • Linux 배포판과 현황
8교시 16:00	◆ **Linux 사용법** • Linux shell • Linux 사용자와 권한 • Linux 기본 명령
9교시 17:00	◆ **shell script** • 원격 접속 • 명령행 편집기 • shell script

첫째시간

7교시 15:00	◆ **Linux 개요** • OS 개요 • Linux의 역사 • Linux 배포판과 현황
8교시 16:00	◆ **Linux 사용법** • Linux shell • Linux 사용자와 권한 • Linux 기본 명령
9교시 17:00	◆ **shell script** • 원격 접속 • 명령행 편집기 • shell script

Linux 개요

Linux란?

Linux Kernel 기반의 오픈 소스 Unix계열(Unix-like) 운영체제들을 의미함(Wikipedia)

- Linux? Linux kernel?
- Unix? Unix-like?
- 운영체제들(family)

유닉스 계열(Unix-like) : '유닉스와 비슷하면서 **유닉스가 아니다**'는 뜻으로 유닉스(UNIX)와는 별개의 용어이다.

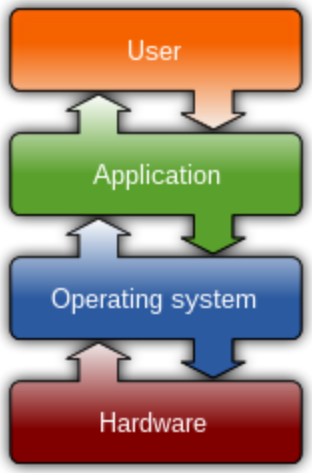
family of open-source Unix-like operating systems :
다양한 배포판, 다양한 HW 지원

OS 구성

운영체제(OS; Operating System) 의 구성

- kernel
- Library
- Utilities
- User Interface

Operating systems



https://en.wikipedia.org/wiki/Operating_system

Common features

- Process management
- Interrupts
- Memory management
- File system
- Device drivers
- Networking
- Security
- I/O

OS Kernel

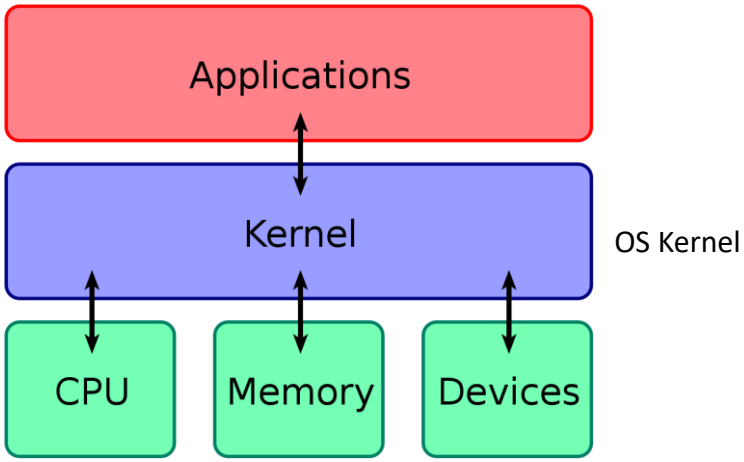
OS Kernel

- 운영체제의 핵심, 하드웨어 자원을 관리하고 프로그램을 실행하게 함

커널의 기능

커널은 다음과 같은 4가지 기능을 수행

- **메모리 관리:** 메모리가 어디에서 무엇을 저장하는 데 얼마나 사용되는지를 추적
- **프로세스 관리:** 어느 프로세스가 중앙 처리 장치(CPU)를 언제 얼마나 오랫동안 사용할지를 결정
- **장치 드라이버:** 하드웨어와 프로세스 사이에서 중재자/인터프리터의 역할을 수행
- **시스템 호출 및 보안:** 프로세스의 서비스 요청을 수신



https://www.redhat.com/ko/topics/linux/what-is-the-linux-kernel

Kernel 종류

Single-tasking and multi-tasking

- 싱글태스킹 : 한 번에 하나의 프로그램만 실행 가능
- 멀티태스킹 : 동시에 2개 이상의 프로그램을 실행 가능(time-sharing)

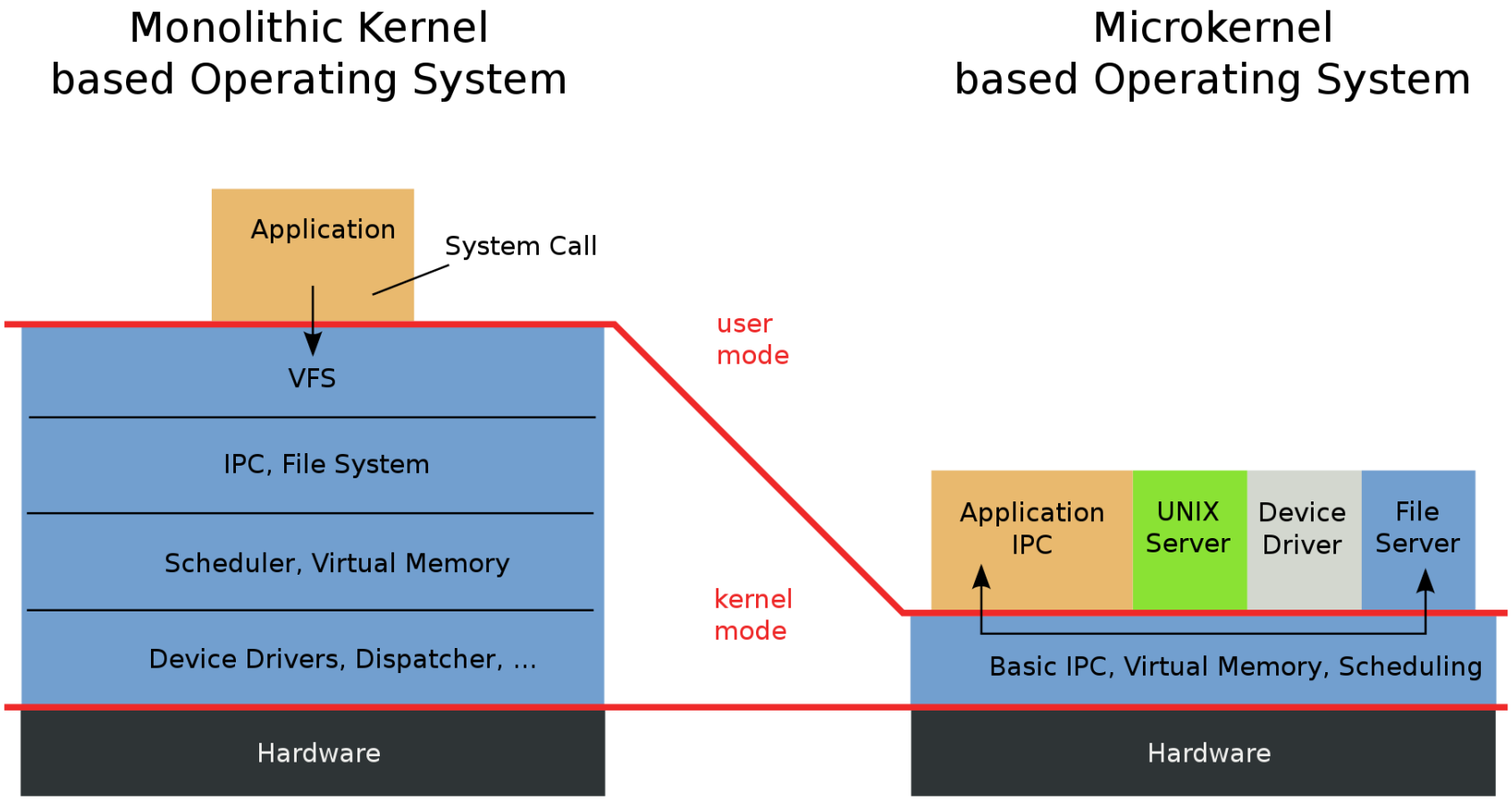
Single- and multi-user

- 단일사용자용(single-user) : 사용자 구별이 없음
- 다중사용자용(multi-user) : 사용자별 자원과 프로세스를 식별하여 여러 사용자들이 독립적으로 사용할 수 있도록 지원

단일형커널과 마이크로커널

- Monolithic Kernel : 모든 운영체제가 하나의 커널 공간에서 실행됨
  - 통합되어 한 번에 실행되므로 더 빠르게 실행
  - 모듈간 상호 작용이 직접 실행됨
  - 구조가 단순함
  - 전체 커널을 잘 만들어야 함
- Microkernel: 최소한의 운영체제 기능만을 커널에 구현하고 다른 기능들은 서버 형태로 제공
  - 작고 독립적이어서 모듈간 영향을 많이 주지 않음
  - 부분적으로 재 로드할 수 있고, 커널 전체를 재 컴파일 하지 않고 기능 추가 가능
  - 실행에 부가 비용이 발생 -> 성능 문제 발생 가능

단일형커널과 마이크로커널



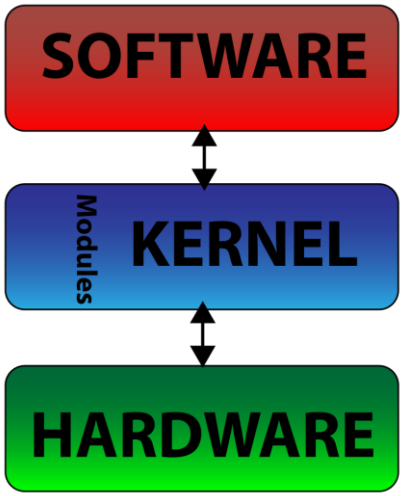
대부분의 Unix, Linux 등

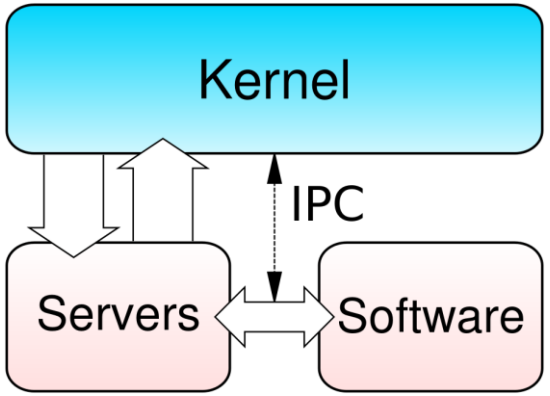
Symbian, Mac OS X, HURD

단일형커널과 마이크로커널

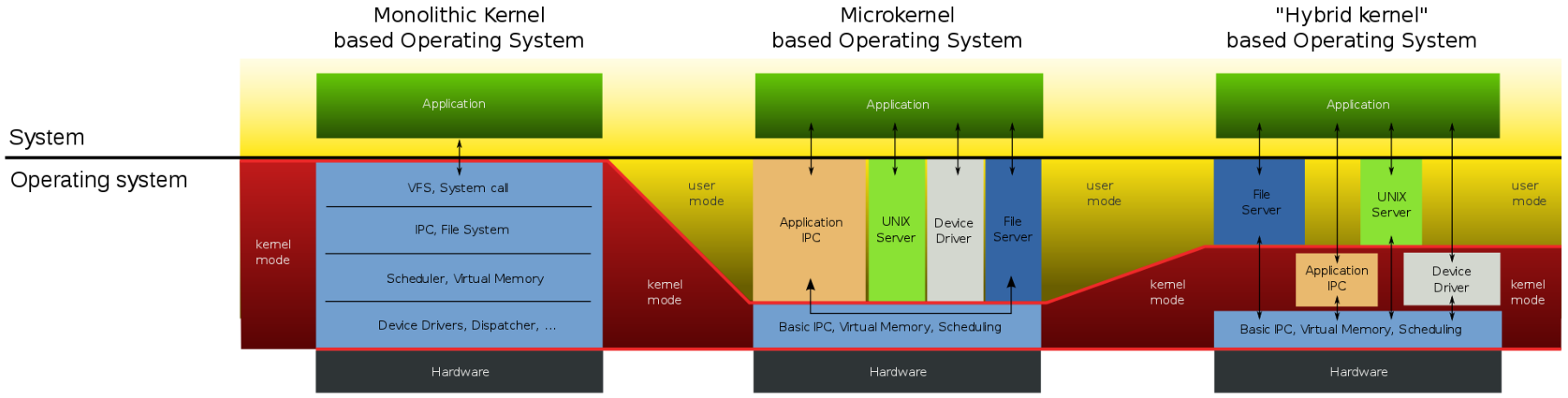
단일형커널과 마이크로커널, 하이브리드커널

단일형(Monolithic) Kernel





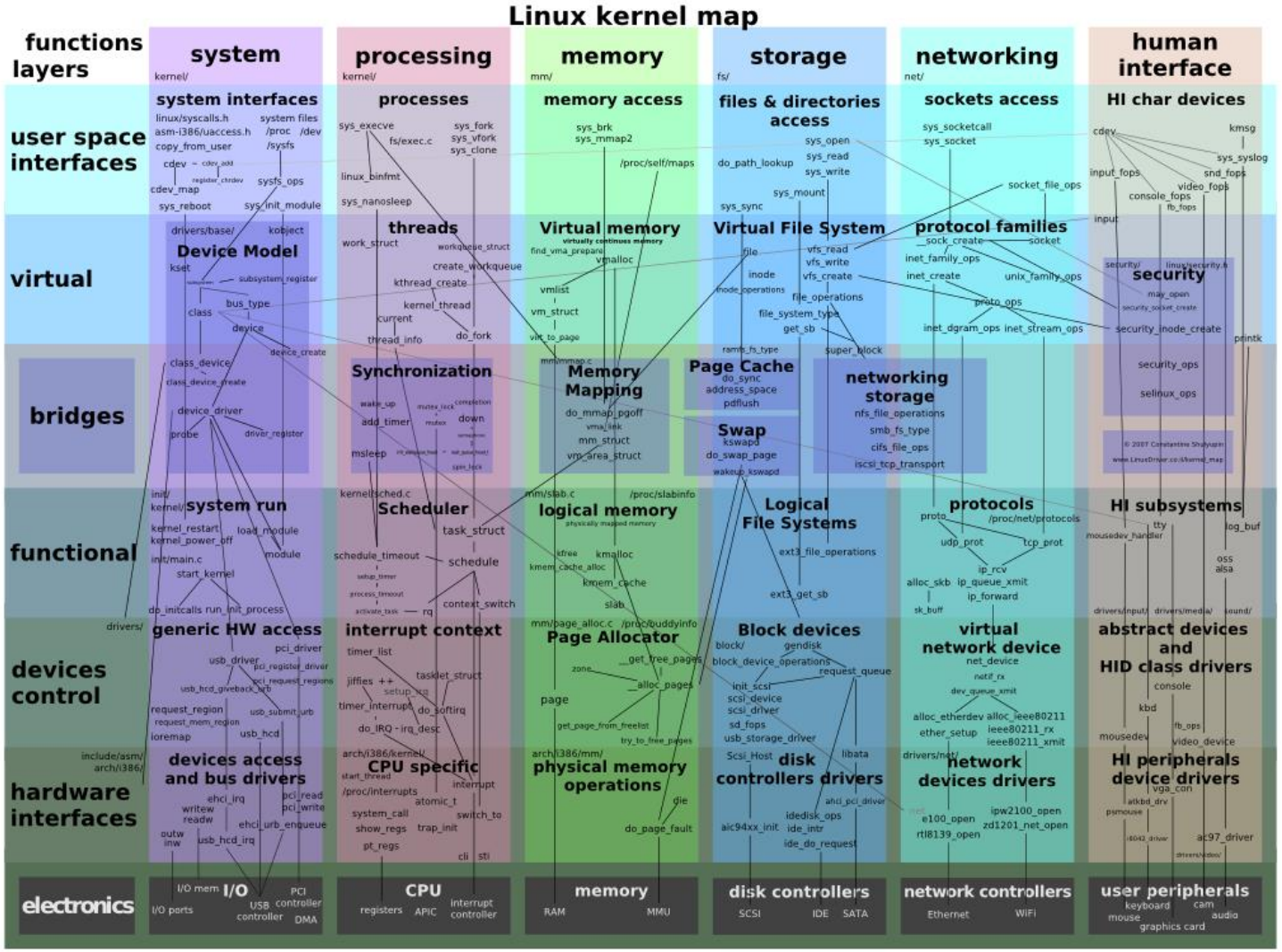
마이크로커널(Michrokernel)



Windows

2W Linux Shell Script : Linux 개요

리눅스 커널맵



https://en.wikipedia.org/wiki/Linux_kernel

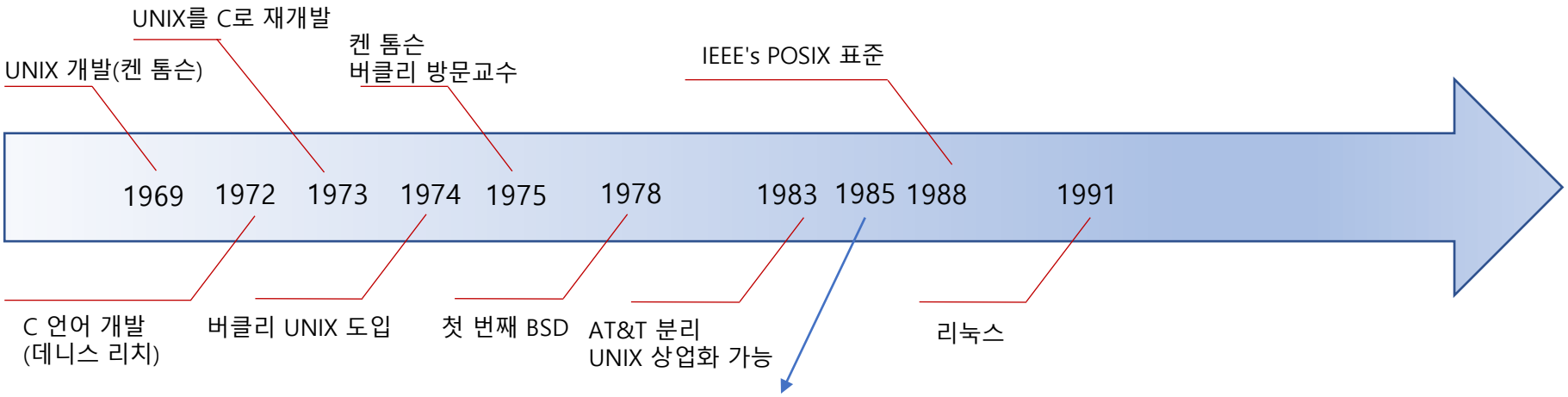
Linux Kernel 분류

Multi-tasking, Multi-user, 단일형커널

- Linux kernel은 기본적으로 Multi-tasking, Multi-user, 단일형커널
- 임베디드 OS의 경우 커널을 수정하여 Single-task나 Single-user용으로 사용하기도 함

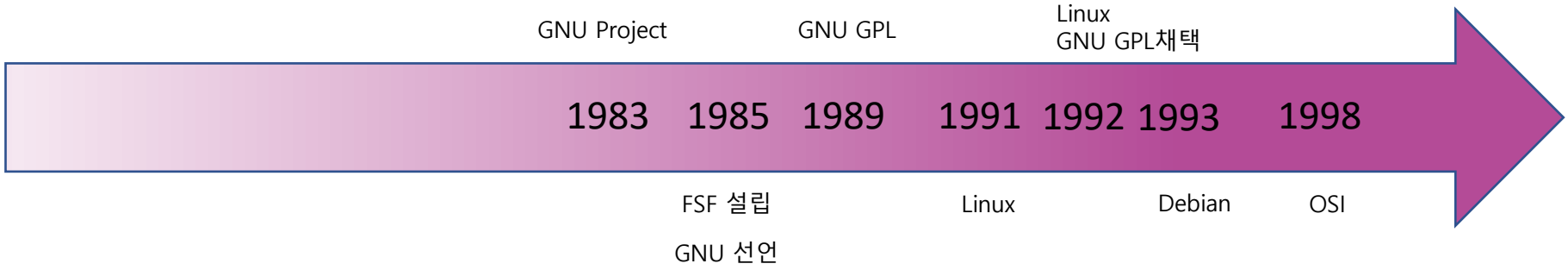
Unix와 Linux의 역사

유닉스(Unix)의 역사



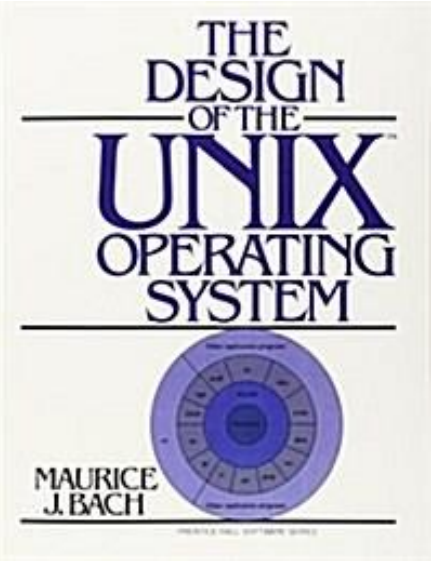
In 1985, Intel released the 80386, the first x86 microprocessor with a 32-bit instruction set and a memory management unit with paging.

Linux 역사



중요한 책들의 출판

The Design of the UNIX Operating System



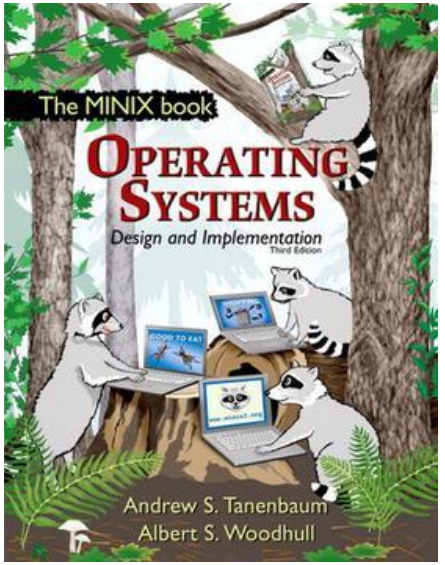
This definitive description principally covered the System V Release 2 kernel, with some new features from Release 3 and BSD

- In 1986
- Maurice J. Bach, of AT&T Bell Labs

Operating Systems: Design and Implementation

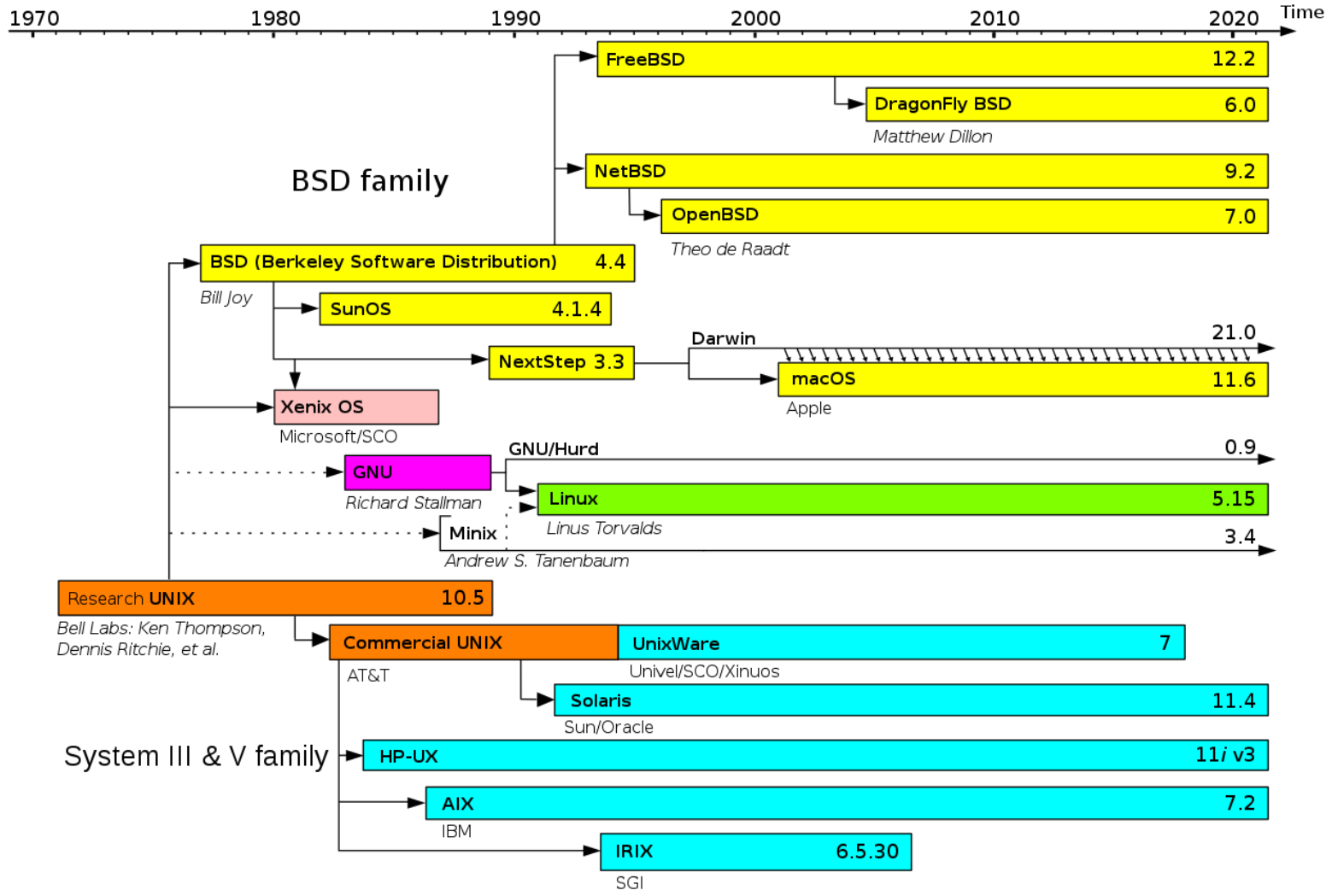
운영체제 설계에 대한 내용을 자세히 다루면 Minix 소스를 포함

- In 1987
- Andrew S. Tanenbaum
- MINIX, a Unix-like system intended for academic use
- 719 쪽의 Minix source code 포함



2W Linux Shell Script : Linux의 역사

Unix 계열



https://ko.wikipedia.org/wiki/%EB%A6%AC%EB%88%85%EC%8A%A4

Linux kernel의 개발

Linux Kernel의 개발과 소스 공개

1991/08/25, 리누스 토발즈는 리눅스 커널을 만들고 usenet(메일리스트)에 이 사실을 알리고 원하는 기능을 알려 달라고 요구함

제목 : 미닉스에 있으면 하고 가장 원하는 기능?

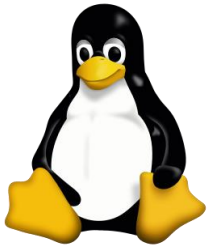
내용 요약

나는 취미로 386(486)용 무료 OS를 만들고 있음. 거의 다 만들어가고 있는데 미닉스를 사용하는 사람들이 미닉스에 대해 좋은 점과 나쁜점을 알려주면 도움이 될 것 같음.

현재 bash와 gcc을 포팅했는데 잘 작동하는 것 같음.

이것은 몇 달 후면 쓸만한 OS가 될 것 같은데 사람들이 원하는 기능을 알고 싶음.

어떤 제안도 좋은데 내가 구현한다고 약속할 수는 없음.



From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Hello everybody out there using minix –
I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).
I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them ☺
Linus (torvalds@kruuna.helsinki.fi)
PS. Yes – it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

Linux kernel의 발전

공동체 참여에 의한 발전

리누스 토발즈가 커널과 일부 툴들을 포팅하여 공개한 후
많은 개발자들의 자발적 참여로 완전한 OS로 발전하게 됨

- 많은 오픈소스 SW들이 포팅됨
- 많은 배포판의 등장
- 많은 기업들의 지원
- 저장소

  https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/
- 커널 최신 버전 : 6.3, 6.2(stable)
  - Ubuntu 배포판(23.04)에서 커널 6.2 사용 예정
- 현재 리더 : 그레그 크로하트맨(Greg Kroah-Hartman)

  리눅스 커널의 주도적인 유지보수자로서 개발을 가이드
- 윌리엄 존 설리반(William John Sullivan)

  자유 소프트웨어 재단의 이사로서[56] GNU 구성
  요소들을 지원

index : kernel/git/torvalds/linux.git

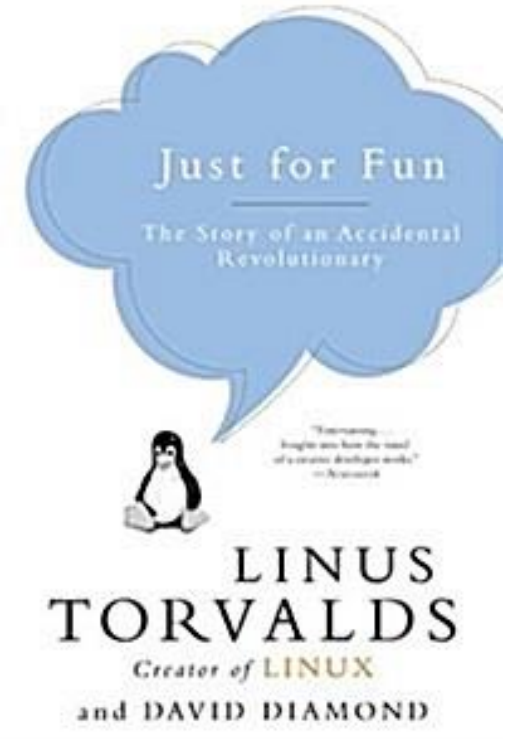
Linux kernel source tree

about summary refs log tree commit diff stats

Commits per author per week

Author	W06 2023	W07 2023	W08 2023	W09 2023	Total
Chuck Lever	0	0	101	2	103
Krzysztof Kozlowski	73	27	1	1	102
Christoph Hellwig	4	57	0	1	62
Greg Kroah-Hartman	53	6	1	0	60
Hans de Goede	58	1	0	0	59
Herbert Xu	3	55	0	0	58
Liam R. Howlett	50	0	1	0	51
Thomas Weißschuh	27	19	1	1	48
Vladimir Oltean	36	0	7	0	43
Dave Chinner	42	0	0	0	42
Others (933)	1425	1269	575	143	3412
Total	**1771**	**1434**	**687**	**148**	**4040**

Linux 그냥 참고로

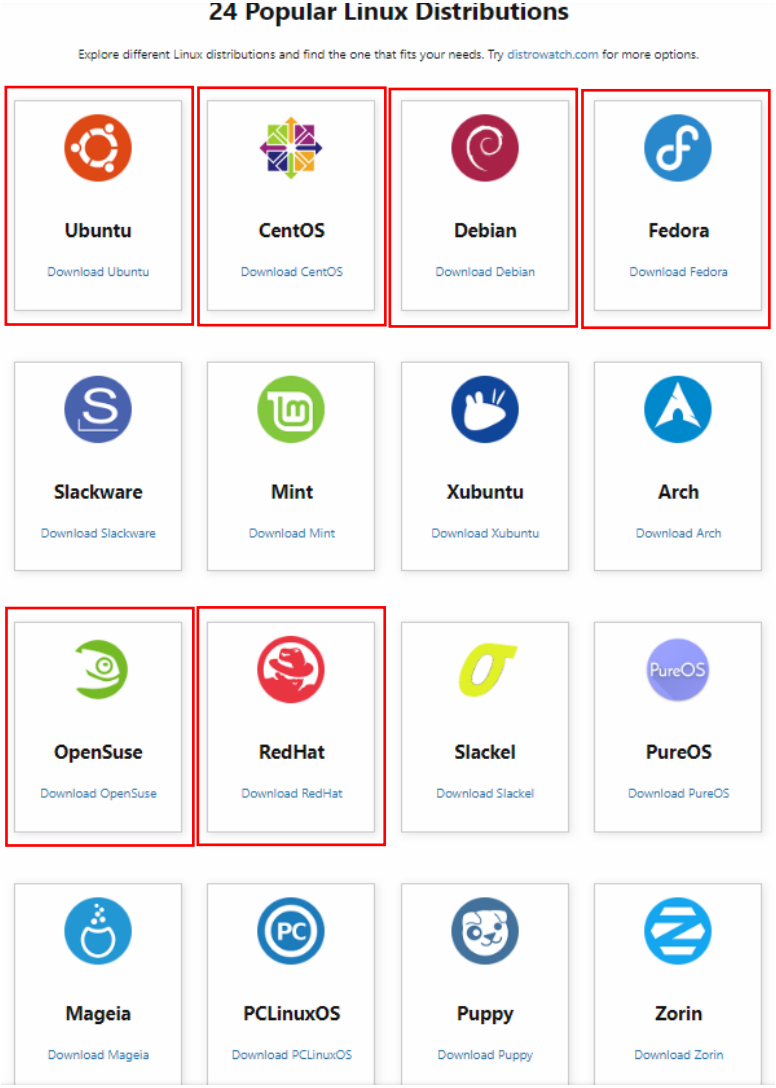


https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=1317258

한국어 번역본은 절판되었음, 도서관에 있음.

Linux 배포판(Distro, Distribution)

- 리눅스 배포판(-配布版, Linux distribution, 간단히 distro)은 리눅스 커널, GNU 소프트웨어 및 여러 가지 자유 소프트웨어로 구성된 운영 체제
- 설치하고 사용하기 쉽게 배포용 패키지를 만들어서 제공
- GNU GPL을 따르는 리눅스 라이선스에 따라 소스 공개만 하면 누구나 리눅스 배포판을 만들어 무료나 유료로 배포할 수 있음
- 리눅스 홈페이지(linux.org)에는 인기있는 24개의 배포판 목록이 있음
- 그 중 자주 보게 되는 배포판들은 다음과 같음
  - Ubuntu
  - Debian
  - Fedora
  - RedHat
  - CentOS
  - Rocky
  - SUSE
  - Arch



Linux 배포판(Distro, Distribution)

Debian

- Debian Project
- stable, testing, unstable
- 패키지 관리 : dpkg, apt, *.deb
- 안정적, 버전 업그레이드 느림

Ubuntu

- Canonical(영국 기업)
- LTS(Long Term Support), non-LTS
- 패키지 관리 : dpkg, apt, *.deb
- Debian을 기반으로 만듦
- 버전 업그레이드 빠름(최신 커널 지원)

RedHat

- IBM 인수
- RHEL(RedHat Enterprise Linux) : 기업용 운영체제(유료)
- 패키지 관리 : rpm, up2date, *.rpm

Fedora

- Fedora Project(RadHat 지원)
- RedHat의 커뮤니티 버전, 실험적, 테스트
- 버전 업그레이드 빠름

CentOS

- CentOS Project(RedHat 제휴)
- RedHat 완벽 호환 기업용 리눅스
- 2021 지원중단

Rocky

- The Rocky Enterprise Software Foundation
- CentOS 계승
- 패키지 관리 : rpm, yum, *.rpm

Linux 배포판(Distro, Distribution)

SUSE

- SUSE Software Solutions(독일 기업, EQT Partners)
- SLES(SUSE Linux Enterprise Server)
- 기업용으로 많이 사용됨
- 유럽에서 많이 사용됨

openSUSE

- openSUSE Project(SUSE 후원 받음)
- 패키지 관리 : ZYpp

Arch Linux

- Levente Polyak(Leader)
- latest stable versions of most software(major release없음)
- 최소한의 설치
- 패키지 관리 : pacman

임베디드

- Adroid
- ChromeOS
- 라즈베리파이용 OS : 라즈비안, SUSE, Ubuntu

기업용 서버에 만이 사용되는 배포판

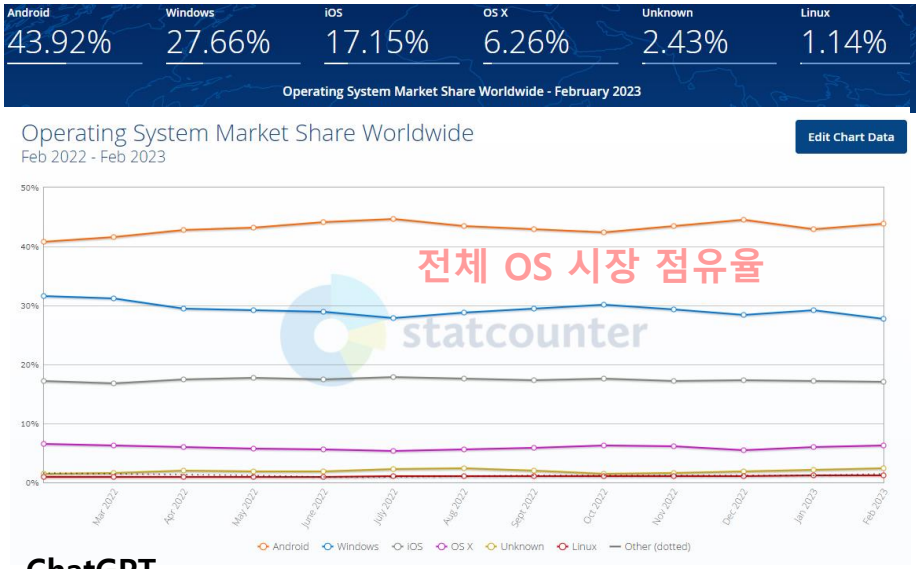
- RedHat(RHEL), CentOS, Rocky Linux, SUSE(SOES), openSUSE

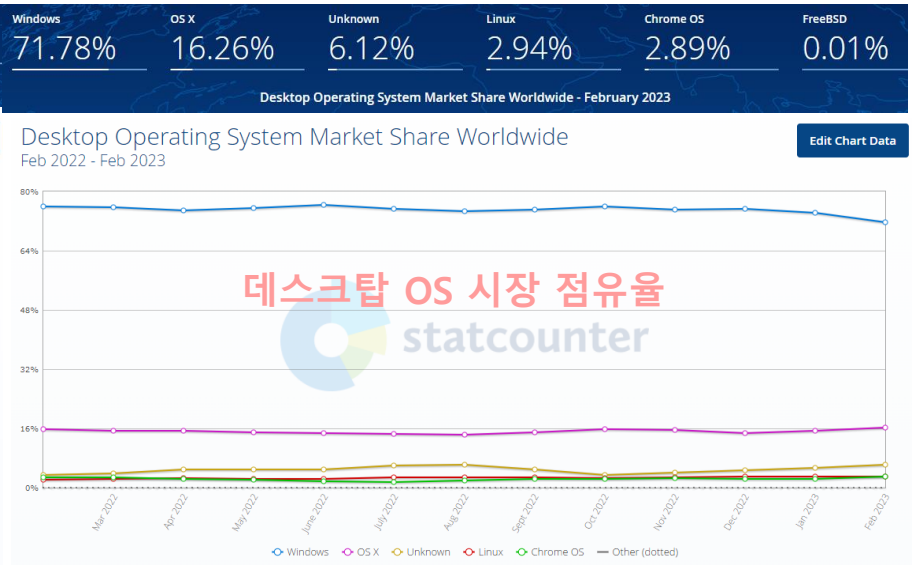
데스크탑, 개인용, 딥러닝용 시스템에 만이 사용되는 배포판

- Ubuntu, CentOS, Rocky Linux

Linux 시장 점유율

시장 점유율과 활용





ChatGPT

As of September 2021, according to StatCounter, the global market share for server operating systems is:

1.Linux - 94.25%

2.Windows - 3.51%

서버 OS 시장 점유율

3.Unix - 0.49%

4.FreeBSD - 0.10%

Note that these figures may vary depending on the source and the specific time period being measured. Additionally, these numbers only reflect server operating systems and do not include desktop or mobile operating systems. It's also important to note that Linux dominates the server market due to its open source nature, flexibility, and stability.

둘째시간

7교시 15:00	◆ Linux 개요 • OS 개요 • Linux의 역사 • Linux 배포판과 현황
8교시 16:00	◆ **Linux 사용법** • Linux shell • Linux 사용자와 권한 • Linux 기본 명령
9교시 17:00	◆ shell script • 원격 접속 • 명령행 편집기 • shell script

Linux shell

OS shell

- 운영 체제 상에서 다양한 운영 체제 기능과 서비스를 구현하는 인터페이스를 제공하는 프로그램
- 셸은 일반적으로 명령 줄과 그래픽 형의 두 종류로 분류
  - CLI(명령줄 인터페이스)
  - GUI(그래픽 사용자 인터페이스)

Unix 계열 CLI

- 본셸 : sh
- 배시 : bash
- z셸 : zsh
  - CLI(명령줄 인터페이스)
  - GUI(그래픽 사용자 인터페이스)

Windows CLI

- 도스 프롬프트 :command.com(과거 윈도우)
- 명령 프롬프트 : cmd.exe
- 파워셸 : powershell.exe

Linux용 GUI

Linux는 GUI는 독립적으로 설치하여 사용

- GNOME
- KDE

https://ko.wikipedia.org/wiki/%EC%85%B8

Linux 사용자와 권한

Linux 사용자 계정

- Linux 사용자는 계정을 가지고 로그인을 해서 사용
- 사용자 계정으로 로그인하기 위해서는 비밀번호(password)로 인증을 받아야 함
- 모든 계정은 숫자로 된 UID 값을 가짐
- root user : 모든 권한을 가짐(super user), root user는 UID 0번을 가짐
- 일반 user : root user가 아닌 모든 사용자 계정
- 모든 사용자는 자신의 home directory를 가짐
- 사용자 계정은 /etc/passwd 파일에서 관리됨
- 사용자 계정에 대하여 로그인을 허용할 수도 있고 금지할 수 있음
- 보안 문제로 root 계정은 로그인을 금지시키는 추세임

Linux 사용자 그룹

- 모든 사용자 계정은 하나 이상의 사용자 그룹에 속함
- 사용자 그룹은 숫자로 된 GID 값을 가짐
- root user는 root group에 속함
- root group GID 0번을 가짐

Linux 사용자와 권한

Linux 권한 관리

- Linux에서 모든 자원은 파일(디렉토리 포함)로 관리됨
- 모든 파일에는 소유권(owner)이 지정됨
- 모든 파일에는 소유자와 소유자 그룹 그리고 다른 사용자에 대하여 권한을 설정할 수 있음
- 모든 파일에는 읽기/쓰기/실행 권한을 설정할 수 있음

루트 권한의 사용

- 보안 문제로 root사용자 계정을 사용할 수 없게 설정하는 경우가 많음(Ubuntu default)
- sudo 명령어 : 일반 사용자 계정이 root 권한으로 명령을 실행할 수 있음
- /etc/sudoers 파일에서 sudo 명령을 사용할 수 있는 사용자 계정 지정

password 바꾸기 권한 필요

- passwd [사용자 계정]

Linux 기본 명령

셸

- 셸 확인 : ps
- 셸 변경 : 셸 프로그램 실행(sh, bash, zsh ; 설치되어 있어야 함)
- 셸 탈출 : exit

파일 보기

- 파일 목록 보기 : ls, ls -a, ls -l, ll(alias)
  - . : 현재 디렉토리
  - .. : 상위(부모) 디렉토리
  - .으로 시작되는 파일 : 숨김 파일
  - 파일 종류 : 파일, 디렉토리, 링크, 파이프

```
npark@npark-dell:~$ ls -al
total 28
drwxr-xr-x 2 npark npark 4096 Mar 13 02:05 .
drwxr-xr-x 3 root  root  4096 Mar  8 13:06 ..
-rw-r--r-- 1 npark npark    5 Mar 13 02:05 a.txt
-rw------- 1 npark npark   31 Mar  8 13:23 .bash_history
-rw-r--r-- 1 npark npark  220 Mar  8 13:06 .bash_logout
-rw-r--r-- 1 npark npark 3526 Mar  8 13:06 .bashrc
-rw-r--r-- 1 npark npark  807 Mar  8 13:06 .profile
```

종류 : -, d, l, p

소유자

소유그룹

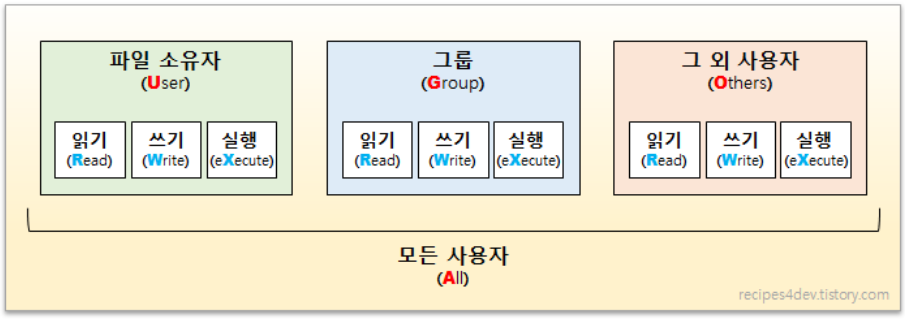
파일크기

최종 수정 시간

파일 권한(파일 모드)

- 9글자 : owser/group/other 3자씩
- 읽기/쓰기/실행 : rwx
- 2진수 값 : 허용(1), 금지(0)
  - rwx : 7
  - - - - : 0

권한 설정



Linux 기본 명령

자동 완성과 과거 명령 실행, 도움 보기

- 명령 입력의 일부만 입력한 수 tab을 입력하여 자동 완성 기능 사용 가능(미리 설정)
- 위/아래 화살표를 이용하여 과거 실행한 명령을 실행
- 명령어 --help 또는 man 명령어

디렉토리

- 루트 디렉토리 : 모든 디렉토리의 출발점(/)
- 디렉토리 위치 : 절대 경로(루트 디렉토리 부터), 상대 경로(현재 디렉토리 부터)
    - / : 루트 디렉토리
    - ~ : 홈 디렉토리
    - .. : 상위 디렉토리
    - . : 현재 디렉토리
- 이동 : cd (change directory), 읽기 권한이 없으면 이동할 수 없음
    - cd <경로> : <경로>로 이동
    - cd .. : 상위로 이동
    - cd 또는 cd ~ : 홈 디렉도리로 이동
    - cd - : 이전에 있던 디렉토리로 이동
- 확인 : pwd(print working directory)

Linux 기본 명령

파일 속성 수정

- chown : 파일(디렉토리)의 소유자 수정(chown uid:gid file_name)
- chmod : 파일(디렉토리)의 권한 수정(chmod 755 file_name, chmod o+x file_name)

파일(디렉토리) 다루기

- mkdir : 디렉토리 만들기
- rmdir : 디렉토리 지우기(빈 디렉토리만 지울 수 있음)
- cp : 복사 (-R 하위 디렉토리까지 복사)
- touch : 빈 파일을 만들기
- mv : 파일이나 디렉토리를 옮기거나 이름을 변경
- rm : 지우기 파일이나 디렉토리를 지울 수 있음(강력한 옵션 ; rm -rf <파일 또는 디렉토리 이름>)
- find : 파일 찾기 (find / -name 문자열)
- which : 경로에 있는 파일 찾기
- tree : 디렉토리를 트리모양으로 보기(별도 설치 필요, tree -L 3)

Linux 기본 명령

텍스트 파일 보기

- 텍스트 파일은 편집 프로그램을 이용하여 읽고 수정할 수 있음
- 텍스트 파일을 보는 명령어
  - more
  - cat
  - head [-n 줄수]: 맨 앞의 10줄만 보기[-n에 설정한 줄 수 만큼 보기]
  - tail [-n 줄수]: 맨 뒤의 10줄만 보기 [-n에 설정한 줄 수 만큼 보기]
  - grep : 지정된 문자열이 포함된 행만 표시

주요 디렉토리 및 파일

- /etc : 시스템 환경 설정에 대한 파일들이 저장된 디렉토리
- /etc/passwd : 사용자 계정
- /etc/group : 사용자 그룹
- /etc/os-release : 배포판 버전 정보
- ~/.bashrc : bash 설정 파일
- /tmp : 임지 파일들을 저장하는 디렉토리
- /etc/apt/soruces.list : 패키지 저장소 목록을 저장

Linux 기본 명령

파이프

- |를 써서 앞의 명령 실행 결과를 뒤 명령의 입력으로 사용할 수 있다
- 예) ps -ef | grep abc

프로세스 보기

- ps : 프로세스를 보는 명령, 현재 셸의 실행 프로세스 표시
  - ps -ef : 현재 시스템에서 실행 중인 모든 프로세스 목록 표시
  - ps -ef | grep python : python 이란 문자열이 있는 프로세스 행 표시

주요 시스템 명령

- uname : os 정보 보기 (uname -a)
- cat /etc/os-release : 배포판 정보 보기
- id : 사용중인 계정에 대한 정보
- uptime : 실행된 시간
- last reboot : 부팅된 시간
- ip addr : ip 주소 보기

Linux 기본 명령

환경 변수 보기

- env : 환경 변수 보기

패키지 관리(Ubuntu, Debian)

- apt : 패키지 관리 명령(관리자 권한 필요)
  - sudo apt update : 패키지 목록 업데이트
  - sudo apt upgrade : 설치된 패키지(소프트웨어)들을 최신 버전으로 업그레이드
  - sudo apt install <설치할 패키지 이름> : 패키지 설치

시스템 자원 보기

- df : 파일 시스템 보기(df -h)
- du : 디렉토리의 파일 크기 보기(du -sh *)
- top : 자원 사용 상위 프로세스 보기
- htop : 별도 설치 필요

Linux 기본 명령

웹 관련 명령

- curl : cli browser
- wget : 웹 다운로드

세째시간

7교시 15:00	◆ Linux 개요 • OS 개요 • Linux의 역사 • Linux 배포판과 현황
8교시 16:00	◆ Linux 사용법 • Linux shell • Linux 사용자와 권한 • Linux 기본 명령
9교시 17:00	◆ **shell script** • 원격 접속 • 명령행 편집기 • shell script

원격 접속

네트워크 통신

- IP 주소 : 네트워크 상의 노드(컴퓨터, 네트워크 인터페이스)주소, 집 주소와 같은 역할(집의 출입문)
  - ipv4 예 : 192.168.0.11
  - 도메인 주소 : www.google.com(사람이 기억하기 좋게 매핑, 실제 통신은 IP주소로 통신함)
- Port : 노드 내에서 프로그램이 통신을 위해 점유하는 번호, 집에 사는 사람의 이름과 같은 역할
  - 하나의 포트 번호는 하나의 프로그램만이 독점할 수 있음
  - 하나의 프로그램이 하나의 포트 번호를 점유하며 실행할 때 리슨(Listen)한다고 함
  - 서비스 프로그램은 항상 응답할 수 있도록 포트로 리슨하고 있어야 함
  - 예) http 서버는 80포트로 리슨하고 있음
  - 주요 서비스 프로그램들이 사용하는 포트는 관례적으로 정해져 있음(변경 가능)
    - http : 80
    - https : 443
    - ssh : 22
  - IP 주소와 포트의 표시 : IP주소:포트번호
    - 예) 192.168.0.11:8080
  - Client/Server : 포트 리슨을 하는 서비스 프로그램과 서비스에 요청을 하는 클라이언트 프로그램으로 이루어진 컴퓨팅 시스템

원격 접속

ssh(Secure Shell Protocol)

- 통신을 암호화 하여 보안을 강화하였음, 현재 가장 많이 사용되는 원격 접속 방법
  - Client/Server 방식
    - 서비스 프로그램 : sshd(ubuntu) (기본 사용 포트는 22)
    - 클라이언트 프로그램 : ssh, scp, sftp
  - ssh 클라이언트를 사용하여 주로 원격 로그인에 사용
  - scp나 sftp를 이용하여 파일 원격 복사나 전송에도 사용됨.
- 사용 방법(클라이언트 사용방법)
  - ssh 사용자id@IP -p 포트번호 : IP의 지정된 포트 번호에 지정된 사용자id로 접속 요청
  - ssh 사용자id@IP주소 : 포트를 특정하지 않으면 22번 포트로 ssh 접속을 요청
  - ssh IP주소 -p 포트번호 : 사용자 id를 특정하지 않으면 현재 시스템 사용자id로 접속을 요청
  - 접속 시 비밀번호 입력을 요구함
    - ssh : 22

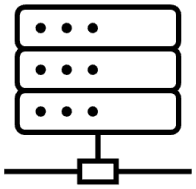
원격 접속

ssh key 사용

- ssh key를 사용하면 비밀번호를 매번 입력하지 않아도 되므로 편리하며 보안성을 더 높일 수 있음
  - 개인키와 공개키를 만들어서 클라이언트와 서버에 저장
- ssh key 만들기(보통 클라이언트에서 만든다)
  - ssh-keygen -t rsa : RSA 알고리즘으로 키 생성

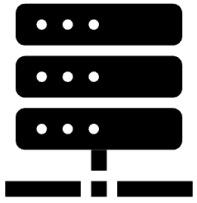
    ~/.ssh/id_rsa : 개인키

    ~/.ssh/id_rsa_pub : 공개키
  - 공개키를 서버로 복사
  - ssh-copy-id -i ~/.ssh/id_rsa.pub [user]@[Remote_Host]



SSH Client

개인키(Private Key)



SSH Server

공개키(Public Key)

명령행 편집기

대표적인 편집기 : nano, vim/vi, emacs

nano

- Ubuntu기본 편집기
- 사용법이 간단함
- 사용법
  - nano file_name : file_name 을 편집, 없으면 새로 생성
  - 아래쪽에 주요 명령어 표시 : 캐럿(^)은 Ctrl 키를 의미함
  - 문자 M은 Alt 키를 의미함
  - Ctrl+g : 모든 항목 보기
  - Ctrl+o : 저장
  - Ctrl+x : 종료
  - Ctrl+w :검색

명령행 편집기

vi/vim

- 대표적인 Unix용 명령행 편집기
- 사용법이 어려우며 기능이 많음
- 사용법
  - vi file_name : file_name 을 편집, 없으면 새로 생성
  - 편집모드와 명령모드가 있음
  - esc키로 명령모드로 변경
  - a, i, r, o, O : 편집모드로 변경
  - 명령모드의 명령
    - x, dd : 삭제
    - yy, p : 복사
    - w, w! : 저장
    - q, q! : 종료
    - wq,wq! : 저장 후 종료
    - h,j,k,l : 이동 키

명령행 편집기

vi/vim 단축키

shell script

shell은 커널과 사용자 사이의 인터페이스이면서 프로그래밍 언어

- shell script == shell program
- Linux 명령어를 모아서 실행할 때 유용 : 배치(batch) 프로그램, 반복적으로 실행할 작업들을 모음
- Unix/Linux 시스템 관리자들에게는 필수
- Hello World
  - hello.sh 파일을 만들고 아래 코드를 입력
  - `echo "Hello World"`
- shell script 실행
  - bash hello.sh
  - sh hello.sh
  - ./hello.sh -> 실행 권한 필요
- 변수
  - a=b : 빈칸 없음
  - echo $a : 변수 사용(echo ${a}, echo $aa, echo ${a}a
  - echo $(which bash) : 명령 결과 사용

https://tldp.org/LDP/abs/html/

shell script

제어문

- 조건문

```
if [condition ]
then
      Statements
else
      Statements
fi
```

- for loop

```
for var in word1 word2 ... wordN
do
    Statements
done
```

- for while

```
while [ condition ]
do
  Statements
done
```

- until

```
until [ condition ]
do
  Statements
done
```

예)

- 명령행에서

```
while :
    > do
    > echo "----- refresh -----"
    > sleep 3
    > done
```

  - 실행을 끝내기 위해서는 Ctrl c

https://tldp.org/LDP/abs/html/

shell script

script file

- sha-bang (#!) :
  - 스크립트의 맨앞에서 다음 명령들을 처리할 인터프리터를 설정
- exit :
  - 스크립트 종료 시 exit 처리, 리턴 값 줄 수 있음

예)

- 다음과 같이 스크립트 파일 시작

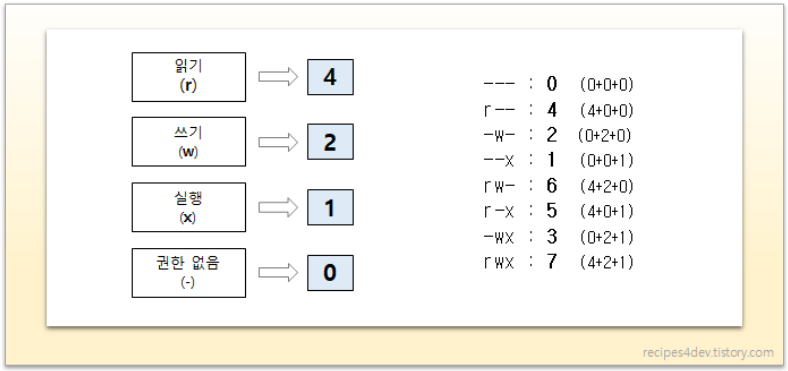
```
#!/usr/bin/bash

echo "name : "
echo "박노헌"

exit 0
```

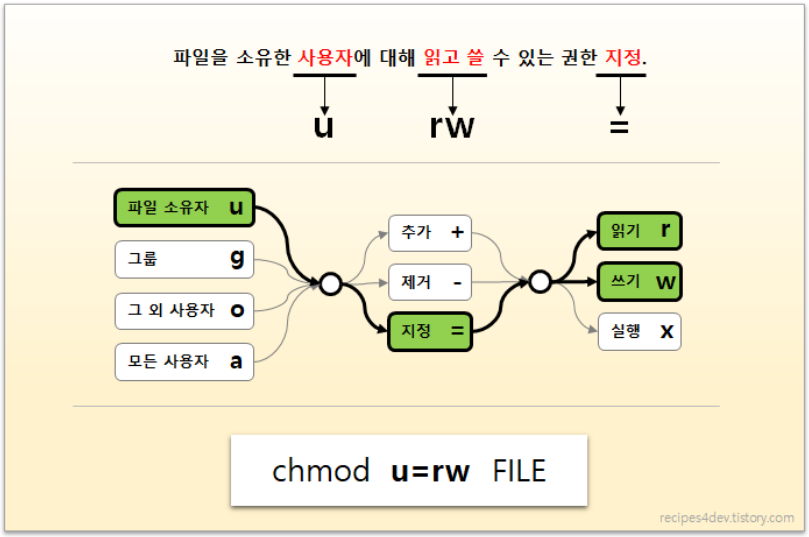
script file 실행 권한

- chmod 명령으로 실행 권한 주기
  - chmod a+x <스크립트파일>
  - chmod 755 <스크립트파일>



https://recipes4dev.tistory.com/175

```
-rwxr-xr-x 1 root root 133792  1월 18  2018 /bin/ls*
```



shell script

script file 백그라운드 실행

- 보통의 실행은 foreground 실행 : 모 프로세스가 종료하면 함께 종료
- 백그라운드에서 계속 실행하도록 하기
  - nohup <스크립트파일> &

백그라운드 실행 프로세스의 종료

- 프로세스 찾기
  - ps -ef | grep <스크립파일> : 프로세스의 pid 찾기
- 프로세스 강제 종료
  - kill pid

https://recipes4dev.tistory.com/175

shell script

Unix/Linux와 Windows의 텍스트 파일 줄바꿈 차이

- Unix/Linux : line feed, LF (₩n)
- Windows : carriage return line feed, CRLF (₩r₩n)
- Linux에서 확인
  - cat -e file_name : ^M으로 표시됨
  - cat -v file_name : ^M으로 표시됨
- 제거 방법
  - `tr -d '\015' < file.log > file.tr.log`
  - 개발 툴 별로 처리 방법들이 있음

https://en.wikipedia.org/wiki/ASCII

shell script

ASCII(American Standard Code for Information Interchange)

- 가장 기본적인 문자 인코딩, 직접 사용되는 경우는 적지만 현재 사용하는 대부분의 인코딩의 일부로 포함되어 있음
- 7bit (128자)로 제어문자와 영문알파벳, 숫자와 특수문자를 모두 표현
  - 제어문자(control character) : 0~31, 127
  - 표현문자(printable character) : 32~126
  - bit 시스템에서 남는 하나의 bit는 패러티로 사용되었으나 필요 없는 경우 사용하지 않고 0으로 처리

Binary	Oct	Dec	Hex	Abbreviation			Unicode Control Pictures[b]	Caret notation[c]	C escape sequence[d]	Name (1967)
				1963	1965	1967				
000 1001	011	9	09	HT/SK	HT		HT	^I	₩t	Horizontal Tab[g]
000 1010	012	10	0A	LF			LF	^J	₩n	Line Feed
000 1011	013	11	0B	VTAB	VT		VT	^K	₩v	Vertical Tab
000 1100	014	12	0C	FF			FF	^L	₩f	Form Feed
000 1101	015	13	0D	CR			CR	^M	₩r	Carriage Return[h]

https://en.wikipedia.org/wiki/ASCII

shell script

Redirection

- 출력을 다른 파일로 전달
  - echo "test" > test.txt
  - echo "test" >> test.txt

STDIN, STDOUT, STDERR

- file descriptor
  - stdin : 0
  - stdout : 1
  - stderr : 2
- redirection
  - ls | cat 1> /dev/null
  - ls | abcd 1> /dev/null
  - ls | cat 2> /dev/null
  - ls | abcd 2> /dev/null

2&>1

- error인 출력도 표준출력으로
  - `ls foo > /dev/null 2>&1` :

예)

```
$ cat foo.txt
foo
bar
baz

$ cat foo.txt > output.txt 2>&1

$ cat no.txt > output.txt 2>&1
```

•There are two places programs send output to: Standard output (stdout) and Standard Error (stderr);
•You can redirect these outputs to a different place (like a file);
•File descriptors are used to identify stdout (1) and stderr (2);
•command > output is just a shortcut for command 1> output;
•You can use &[FILE_DESCRIPTOR] to reference a file descriptor value;
•Using 2>&1 will redirect stderr to whatever value is set to stdout (and 1>&2 will do the opposite).

https://www.brianstorti.com/understanding-shell-script-idiom-redirect/

https://hwan-shell.tistory.com/356

과제

원격접속 그리고 셸 스크립트

- 원격 시스템에 ssh 로그인
- 자신의 공개 키를 만들어서 원격 시스템에 복사
- 원격 시스템에 자신의 닉네임으로 디렉토리 만들기
- 원격 시스템에 자신의 닉네임으로 만들 디렉토리에 셸스크립트 만들기(만들어서 복사해도 됨)
  - file name : w2_homework.sh
  - ./w2_homework.sh로 실행되어야 함(실행 안되거나 오류 나면 무효)
  - 학번, 이름 출력
  - w2_homework.txt 파일을 찾아서 경로와 행의 수, 그리고 마지막 라인을 출력(find 명령 사용)
    - w2_homework.txt 홈디렉토리의 하위 어딘가에 있음.

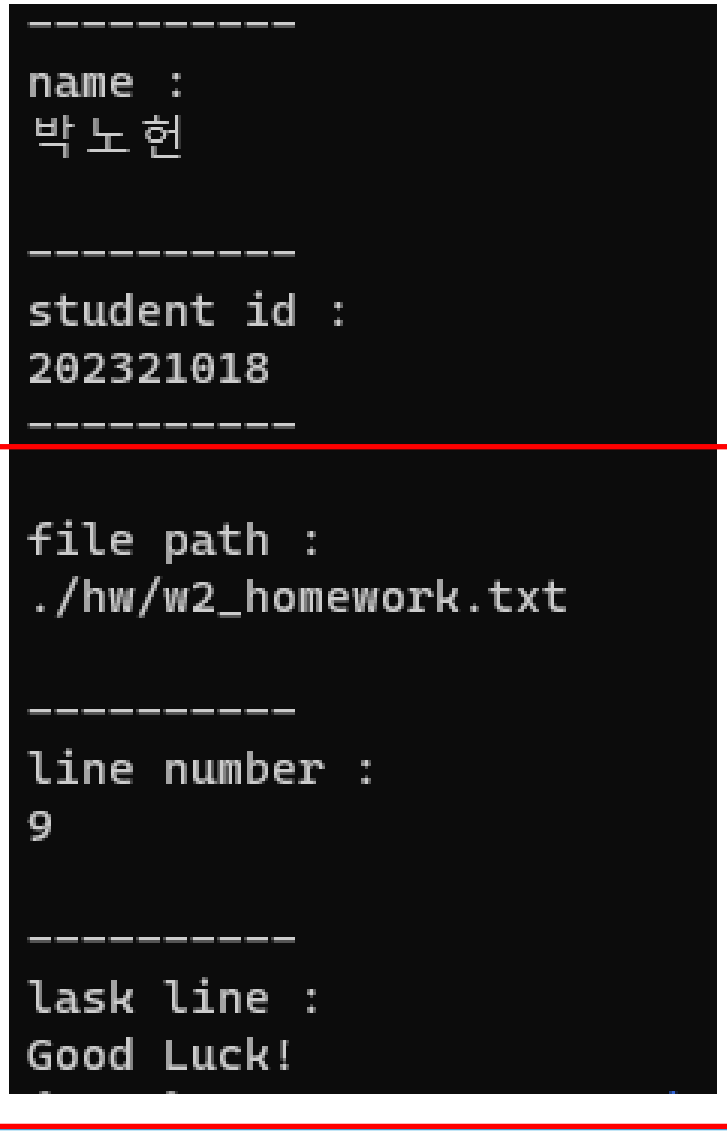
제출

- 다음 주 월요일(20) 자정(24시)까지(KST;한국 표준시)
- 원격 시스템의 자기 디렉토리와 LMS에 모두 제출
- github의 Discussions에서 질의 응답을 하는 것은 허용
- 직접 답을 물어보거나 알려주는 것은 부정행위로 간주

과제

결과 형식 : 옆의 예처럼 출력

```
name :
<이름>
student id :
<학번>
file path :
<w2_homework.txt의 절대 경로>
line number :
<w2_homework.txt의 행 수, 숫자만 표시, 예) 7>
last line :
< w2_homework.txt의 마지막 줄 전체 출력>
```



파일의 위치와 내용은
변경이 될 예정 어떻게
변경이 되어도 찾아서
내용을 가지고 오면 됨