

1장 리눅스 소개

내용

- 컴퓨터와 운영체제
- 유닉스와 리눅스
- 리눅스의 특징
- 리눅스 사용자

1.1 컴퓨터와 운영체제

- 컴퓨터 시스템의 구성요소
 - 하드웨어:
 - 프로세서(CPU)
 - 메모리(주기억장치)
 - 디스크(보조기억장치)
 - 다양한 입출력장치 - 네트워크, 그래픽, 키보드, 마우스, USB ...
 - 소프트웨어:
 - 운영체제(OS)
- 컴퓨터 사용자들은 운영체제를 통하여 컴퓨터 시스템을 사용함
 - 컴퓨터 사용하기 → 운영체제 사용하기
 - 사용자 인터페이스 활용 → shell
 - 다양한 유틸리티 프로그램들 활용

운영체제 (Operating System)

■ 운영체제

- 하드웨어의 각 장치들을 효율적으로 사용하도록 통합하고 제어함
- 응용프로그램이 이러한 장치들을 사용할 수 있도록 서비스를 제공함
- 사용자/응용프로그램과 하드웨어 간의 인터페이스 역할

■ 운영체제의 예

- MS-DOS, OS/2
- Windows 95/98, Windows NT/2000/XP/Vista /7 /8 /10
- **UNIX(Solaris, FreeBSD, AIX, HPUX, ...), Linux**
- classical Mac OS, **OS-X (NeXT OS/BSD UNIX 기반), mac OS Sierra**
- MVS (IBM System/370 mainframe 용), OS/390, z/OS
- VMS, **OpenVMS** – VAX (DEC사, 현재 HP에 인수됨) 등

운영체제와 컴퓨터 시스템

■ 운영체제와 컴퓨터 시스템

- Linux/UNIX는 많은 종류의 컴퓨터 시스템에서 사용 가능
- 특히 리눅스는 거의 전 범위의 시스템에서 사용 가능
 - 소형 임베디드 시스템 ~ 고성능 슈퍼 컴퓨터
- 대부분의 다른 운영체제는 특정 계열의 시스템에서만 사용 가능

■ 리눅스와 유닉스의 관계

- 유닉스와 거의 동일한 사용자 환경을 제공하도록 설계됨
- 유닉스와 유사한 사용자 인터페이스 제공
 - ➔ 리눅스 사용자 \approx 유닉스 사용자

1.2 유닉스와 리눅스

■ 유닉스 (UNIX) 개발

- 미국 Bell 연구소의 Ken Thompson이 1969년에 개발
- 빠르고 작고 효과적인 간단한 운영체제를 목표로 함
 - MULTICS – 저자가 사용했던 복잡하지만 느린 성능의 운영체제
 - UNIX – 저자가 개발한 운영체제 (UNI : 한 가지 일을 잘 함)
- PDP-7 시스템 용으로 개발
- 어셈블리 언어로 작성

■ UNIX를 고급언어인 **C언어**로 다시 작성

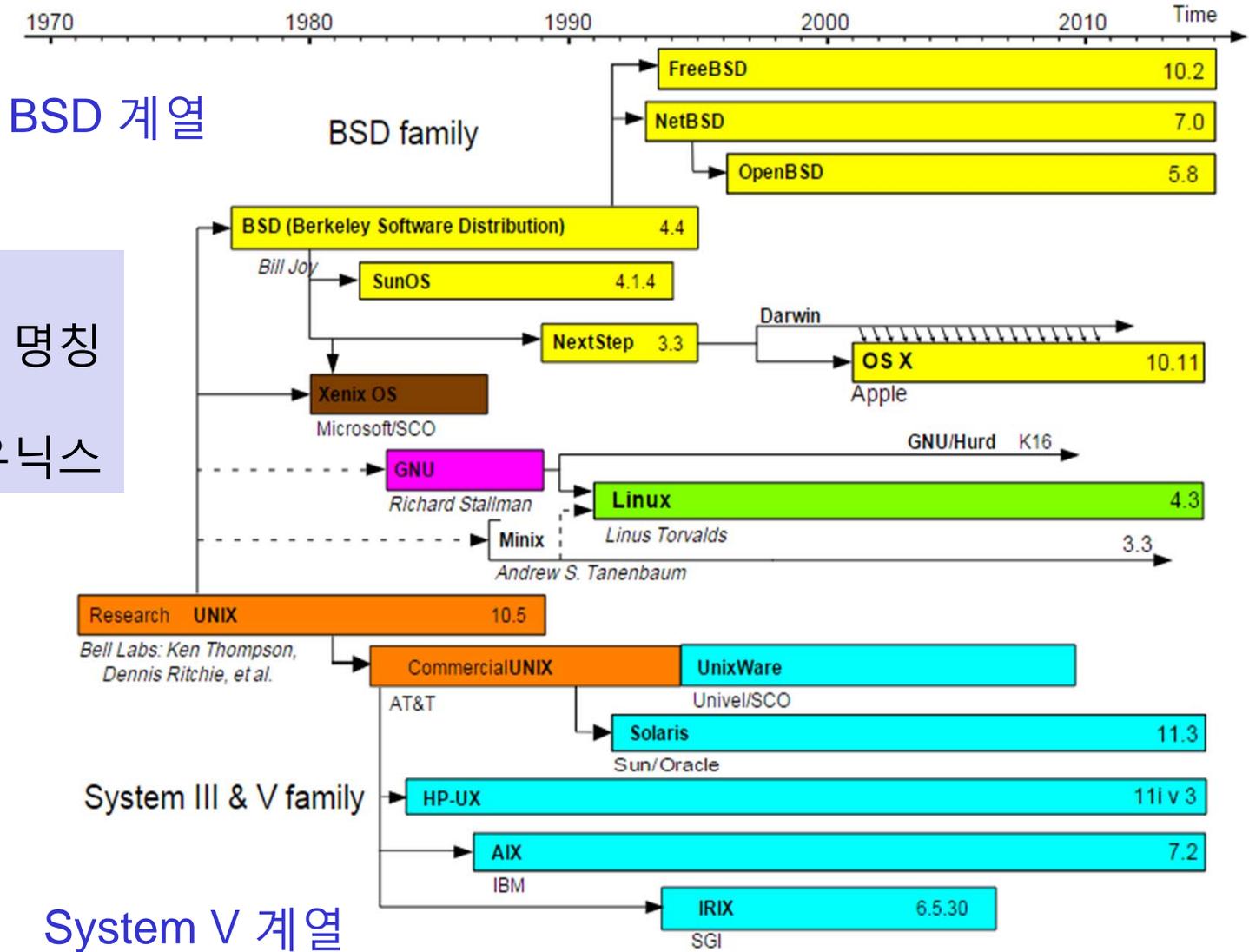
- Dennis Ritchie가 C언어를 개발하고, UNIX 코드 재작성 제안
- 대부분 C언어로 작성되어 이식성이 좋고, 기능 확장 및 개선이 용이함

■ UNIX source code를 연구용으로 무상 제공

- 특히 Berkeley대학교에서 UNIX에 대한 많은 연구가 이루어져서 많은 개선이 이루어짐 : 네트워크, 메모리 관리 등 ...
 - ➔ BSD 버전 (현재 FreeBSD로 이어짐)

Unix의 발전

UNIX
 상표권 등록 명칭
 Unix
 일반적인 유닉스



운영체제 인터페이스 표준

■ 운영체제 인터페이스 표준의 필요성

- 유닉스 버전이 다양해짐에 따라서 응용 프로그램 호환성에 문제가 발생
- 운영체제 버전에 관계없이 이식성 높은 응용 프로그램 개발이 필요

■ POSIX(Portable Operating System Interface)

- UNIX System V의 응용 프로그램 인터페이스와 사용자 인터페이스를 기반으로 운영체제 interface 표준안이 제정됨
- 그렇지만 BSD UNIX 기능들의 대부분을 포함
 - POSIX.1 (IEEE standard 1003.1)
- 대부분의 UNIX와 Linux가 표준을 지원함

■ POSIX와 응용프로그램 호환성

- POSIX를 사용하여 작성된 프로그램은 POSIX를 지원하는 운영체제에서 실행시킬 수 있다.
- UNIX/Linux가 아닌 많은 운영체제들도 POSIX를 지원하고 있음

리눅스

■ 리눅스 개발

- 헬싱키 대학교 학생이던 Linus Torvalds가 1991년에 개발한 유닉스 호환 운영체제
- UNIX 소스코드와는 독립적으로, 교육용 운영체제인 **Minix** 를 학습한 경험을 바탕으로 개발 : Linus' Minix = Linux

■ 리눅스의 발전

- Linux 1.0 : 1994년 소스코드를 GPL 라이선스로 공개
- Linux 2.0 : 1996년 → Linux 2.6.39까지 발전
- Linux 3.0 : 2011년 → Linux 3.18.48까지 발전
- Linux 4.0 : 2015년 → 2017년 2월말 현재 Linux 4.10까지 발전

■ GNU GPL (General Public License)

- 누구나 소스 코드를 변경, 개발하여 재배포할 수 있다.
- GPL로 공개된 프로그램에서 파생된 프로그램도 저작권을 인정하지 않고 GPL로 공개해야 함.
- 리눅스가 안정적이고 강력한 운영체제로 발전한 원동력이 됨.

리눅스의 활용 정도

- 슈퍼 컴퓨터용 운영체제의 약 95%
- 서버용 운영체제의 약 60%
- 데스크톱은 윈도우에 비해서 적음
 - 일반인들이 사용할 수 있는 응용 프로그램들이 부족
 - 최근에 Vmware, VirtualBox와 같은 가상 머신을 사용하여 기존 운영체제와 함께 Linux를 설치하여 사용하는 사용자가 증가 추세임
- Android, Tizen과 같은 임베디드 시스템용 운영체제도 리눅스 커널 기반으로 개발됨
- 많은 소형 임베디드 시스템들이 임베디드 리눅스를 사용

리눅스 배포판(distribution)

■ 배포판

- “커널 + 유틸리티 프로그램 및 라이브러리, 사용 환경”을 함께 제공
- 많은 무료 배포판이 있음

배포판	홈페이지 주소
우분투	http://www.ubuntu.com
데비안	https://www.debian.org/
페도라	https://getfedora.org/
CentOS	https://www.centos.org/
오픈수세	https://www.opensuse.org/
리눅스민트	http://www.linuxmint.com/

- 안정적인 사용과 지원을 위해서 유료 배포판(예: Red Hat) 사용 가능



페이지 방문 순위		
데이터 스펠:		
Year 2016		
가기		
순위	배포본	HPD*
1	Mint	2905
2	Debian	1819
3	Ubuntu	1576
4	openSUSE	1277
5	Manjaro	1145
6	Fedora	1081
7	Zorin	944
8	elementary	927
9	CentOS	828
10	Arch	780
11	deepin	749
12	Mageia	682
13	PCLinuxOS	676
14	Ubuntu MATE	650
15	Android-x86	605

유닉스 배포판

- 유닉스 계열 운영체제도 무료로 배포되는 버전이 존재함

이름	홈페이지 주소
솔라리스	http://www.oracle.com/solaris
FreeBSD	https://www.freebsd.org/
OpenBSD	http://www.openbsd.org/
NetBSD	https://www.netbsd.org/

1.3 리눅스의 특징

■ 유닉스의 특징

- 이식성
- 다중 사용자/다중 프로세스 지원
- 단순성 – 커널에 필요한 최소 기능 제공/ 나머지는 OS 밖에서 지원
- 개방성 – 소스코드 공개, 표준 인터페이스(POSIX)
- 셸 (shell), GUI – OS 밖에서 제공하는 사용자 인터페이스
 - 셸 스크립트 – 유틸리티들을 결합하여 셸이 실행하는 프로그램
- 네트워킹 – 현재의 많은 인터넷 기능들이 유닉스 기반으로 개발됨

셸	GUI	응용 프로그램
라이브러리		
운영체제 API		
운영체제 커널		
하드웨어		

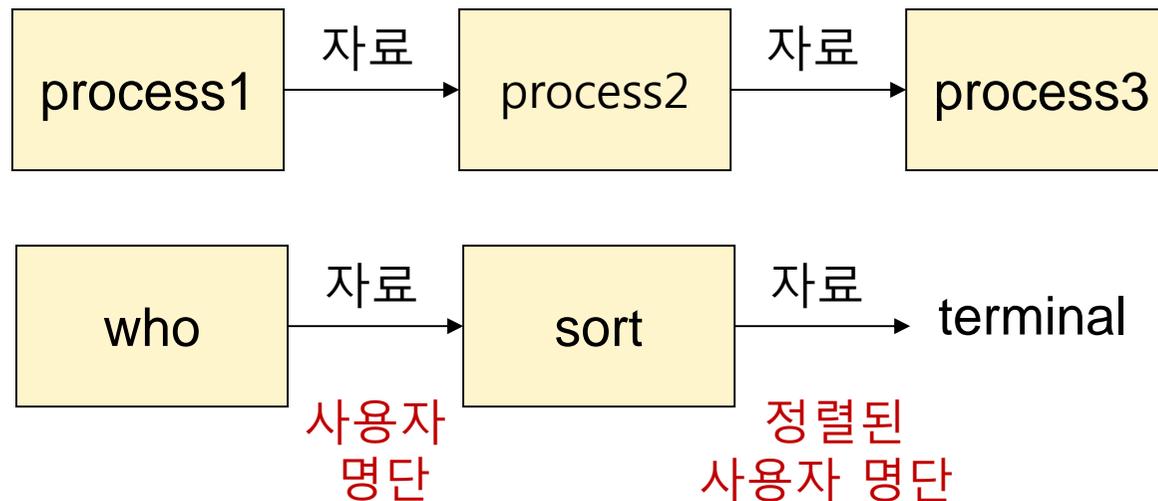
■ 리눅스의 특징

- 유닉스 호환 운영체제
- 공개 소스
- 빠른 업그레이드
- 저비용 및 빠른 성장
- 다양한 플랫폼 지원 – 30종 이상의 CPU, 다양한 파일시스템 지원

1.4 리눅스 사용자

■ UNIX 설계 철학

- 각 utility program은 **한 가지** 작업을 잘 수행해야 함.
- 간단한 작은 utility들을 결합하여 복잡한 작업을 수행함
 - 간단한 결합 : pipe 기능 사용



- 복잡한 결합 : 셸 스크립트(셸 프로그램) 사용
- 다양한 작업의 효율적 수행을 위해서 결합 방식을 알아야 함

UNIX/Linux 사용자 분류

■ UNIX/Linux 사용자의 분류

- **일반 사용자(non-programer)** : 필요한 유틸리티 사용
- **셸 사용자**: 셸의 기본 기능과 간단한 shell script 작성
- **고급 셸 사용자**: 셸의 고급 기능과 다양한 목적을 위한 shell script 작성
- **프로그램 개발자**: C/C++ 언어나 Perl, Python 등의 프로그램 작성
- **시스템 관리자** : 시스템 관리 방법 이해, 수행
- **시스템 개발자** : 사용자를 위한 커널, 라이브러리, 유틸리티들을 개발

■ 이 과목의 목표

- 고급 셸 사용자 및 프로그램 개발자