

Introduction

2015. 6. 30

배영현

과목 소개

■ 수업 목표

- 소프트웨어의 동작 원리를 이해한다.
- 운영체제의 주요 기능과 설계 원리를 이해한다.
 - 응용 프로그램의 설계와 구현에 응용할 수 있다.
- 운영체제가 제공하는 시스템 프로그래밍 인터페이스를 이해한다.
 - 운영체제의 서비스를 이용하는 프로그램을 작성할 수 있다.

■ 필요 지식

- 컴퓨터 시스템과 프로그래밍에 대한 기본적인 이해
- 자료구조, C 프로그래밍 언어
- 리눅스 개발 환경

학습 목표

- 리눅스에서 다양한 개발 도구 프로그램을 사용하여 소프트웨어를 작성할 수 있다.
- 운영체제의 역할과 기본 동작을 이해하고, 응용 프로그래밍 인터페이스를 이용할 수 있다.
- 파일 시스템의 개념을 이해하고 파일 입출력 프로그래밍을 할 수 있다.
- 프로세스와 스레드의 개념을 이해하고 관련 API를 이용하여 프로그램을 작성할 수 있다.
- 동기화를 통한 병행 제어의 개념을 이해하고 동기화와 데드락 방지 프로그래밍을 할 수 있다.
- 가상 메모리의 개념과 메모리 관리 기술을 이해하고, 동적 메모리 할당과 메모리 조작 프로그래밍을 할 수 있다.

수업 교재

■ 주 교재

- Operating System Concepts (8th/9th Edition)
Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne
- 리눅스 시스템 프로그래밍 / Robert Love

■ 참고 교재

- Advanced Programming in the UNIX Environment
W. Richard Stevens, Stephen A. Rago
- 유닉스 리눅스 프로그래밍 필수 유틸리티 / 백창우

수업 활동

- 강의와 실습
 - 간단한 예제 프로그램의 작성과 이해
- 과제
 - 학습 내용 정리 (요약 노트, 마인드맵)
 - 실습을 보충하는 예제 프로그램 작성
- 기말 프로젝트
 - MyShell - 간단한 기능의 셸 구현
 - 수업 진행에 따라 단계별로 진행

성취도 평가

- 평가 방법
 - 개념의 이해
 - 필기 시험 (기말고사만 실시)
 - 학습 내용 정리 과제
 - 시스템 프로그래밍 역량
 - 프로그래밍 과제와 기말 프로젝트
- 성취도 산정 방법
 - 학업 과정과 최종 역량을 각각 고려하는 방식 (수업 계획서 참고)
- 기타
 - 결석 5회 초과 시, 평점은 F
 - 자가 학습 결과물 제출하면 출석 인정

오늘 수업의 목표

운영체제가 무엇인지 왜 필요한지 이해한다.

운영체제의 주요 역할/기능을 이해한다.

리눅스를 실습 환경으로 설치한다.

수업 내용

- 운영체제의 정의
- 운영체제의 역할
 - 프로세스 관리, 메모리 관리, 저장장치 관리, 보호와 보안
- 리눅스 운영체제 소개
- 가상 머신과 리눅스 설치
 - VirtualBox와 Xubuntu

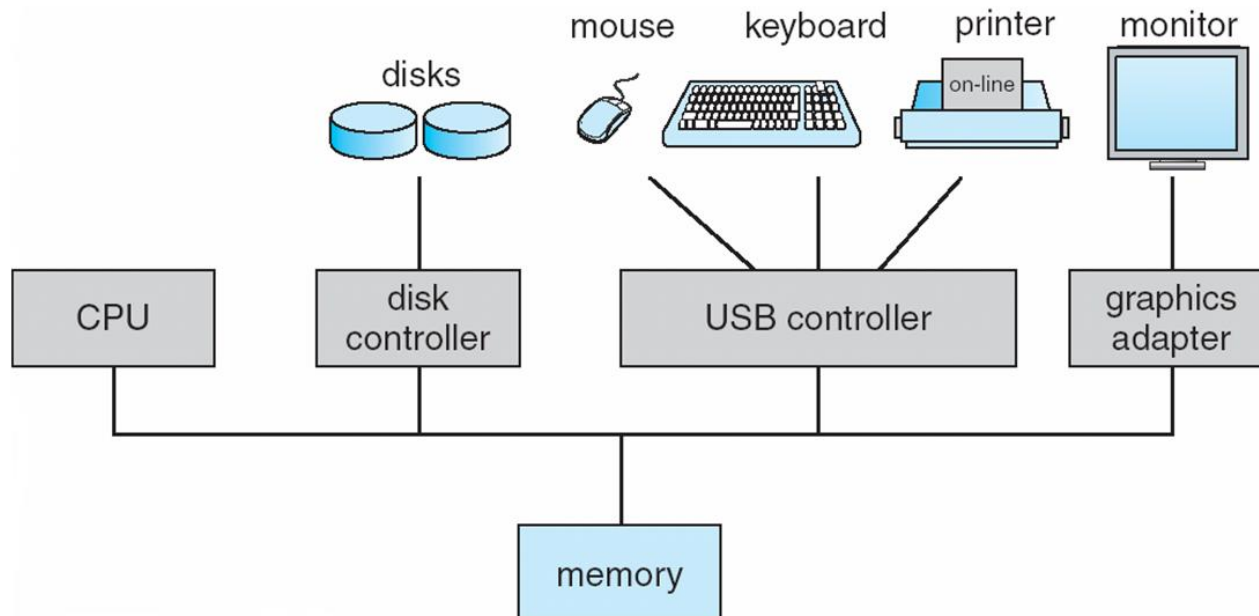
컴퓨터 시스템의 구성

- 하드웨어 (Hardware)
 - 기본적인 컴퓨팅 자원을 제공 (CPU, memory, I/O devices)
- 운영체제 (Operating system)
 - 다양한 응용 프로그램과 사용자들 사이에서 하드웨어의 사용을 제어하고 조정
- 응용 프로그램 (Application programs)
 - 사용자의 문제를 해결하기 위해 시스템 자원을 사용하는 방법을 정의
 - Word processors, compilers, web browsers, database systems, video games
- 사용자 (Users)
 - People, machines, other computers

컴퓨터 시스템의 하드웨어 구성

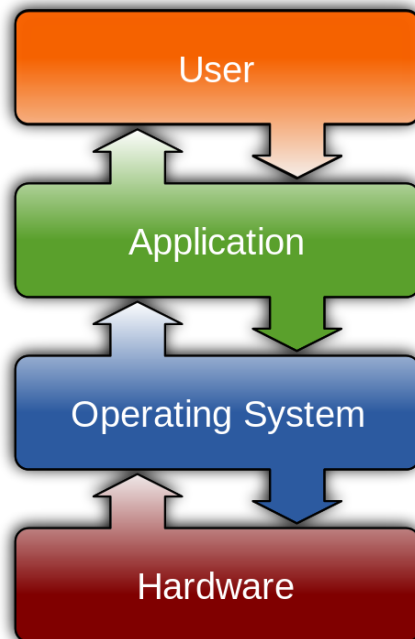
■ 컴퓨터 시스템

- 하나 이상의 CPU와 장치 제어기(device controller)가 공유 메모리를 접근할 수 있는 공통 버스(bus)에 연결
- CPU와 장치 제어기는 메모리에 접근하기 위해 경쟁하면서 병행적으로 실행 (concurrent execution)



운영체제(Operating System)란?

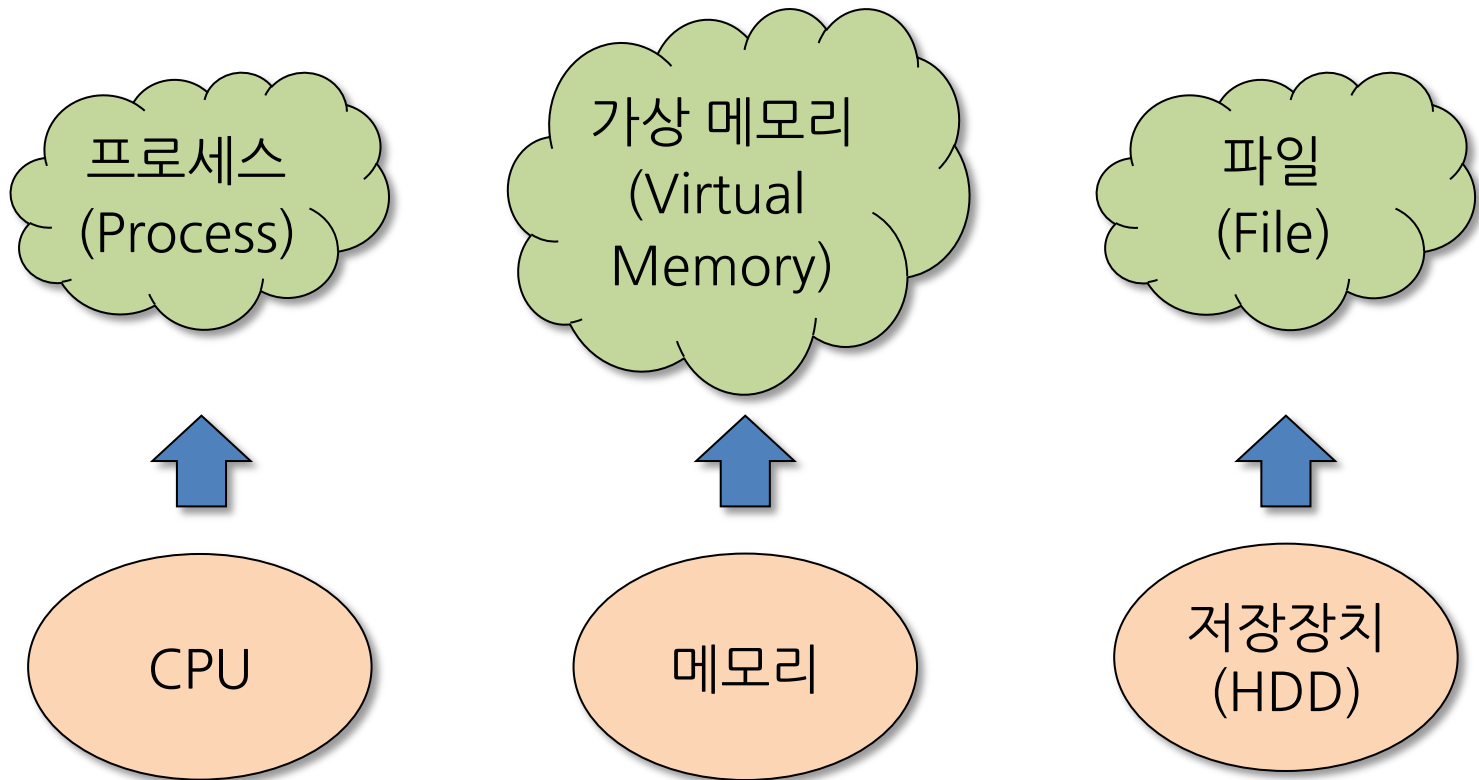
- 컴퓨터 하드웨어와 사용자 사이의 중재 역할을 수행하는 프로그램
 - 프로그램을 실행하고 문제를 쉽게 해결함
 - 컴퓨터 시스템을 편리하게 사용할 수 있도록 함
 - 컴퓨터 하드웨어를 효율적으로 이용하도록 함



source: wikipedia

운영체제(Operating System)란?

컴퓨터 시스템을 추상화하여 프로그램과 사용자에게 제공



운영체제(Operating System)의 정의

- 자원 할당자 (resource allocator)
 - 하드웨어 자원을 관리하여 효율적이고 공정하게 사용 요청을 처리
- 제어 프로그램 (control program)
 - 에러와 적합하지 않은 컴퓨터 사용을 방지하면서 프로그램을 실행
- 컴퓨터에서 항상 실행되는 하나의 프로그램 (커널, kernel)
 - cf.) 시스템 프로그램, 응용 프로그램

운영체제의 핵심 기능

- 다중 프로그래밍 (Multiprogramming)
 - CPU와 자원의 이용률 (utilization) 향상
 - CPU가 항상 동작하도록 작업(job, 코드와 데이터)을 배치
 - 모든 작업 중에서 일부가 메모리에 적재되며 작업 스케줄링(job scheduling)에 의해 실행될 작업을 선택
- 시분할 (Time-sharing) - 멀티태스킹(multitasking)
 - 사용자가 실행 중인 작업과 상호작용이 가능하도록 CPU가 작업을 매우 빈번하게 교체하여 실행 (interactive computing)
 - 응답 시간(response time)이 1초 미만
 - 각 사용자는 하나 이상의 프로그램을 메모리에서 실행 (process)
 - 실행 대기 중인 여러 개의 작업을 CPU 스케줄링으로 처리
 - 스와핑(swapping)과 가상 메모리(virtual memory) 기술

운영체제의 역할

- 프로세스 관리 (Process Management)
- 메모리 관리 (Memory Management)
- 저장장치 관리 (Storage Management)
 - 파일 시스템 (File system)
 - 대용량 저장장치 (Secondary Storage)
- 입출력 시스템 (I/O System)
- 보호와 보안 (Protection & Security)
- 네트워킹 (Networking)

프로세스 관리 (Process Management)

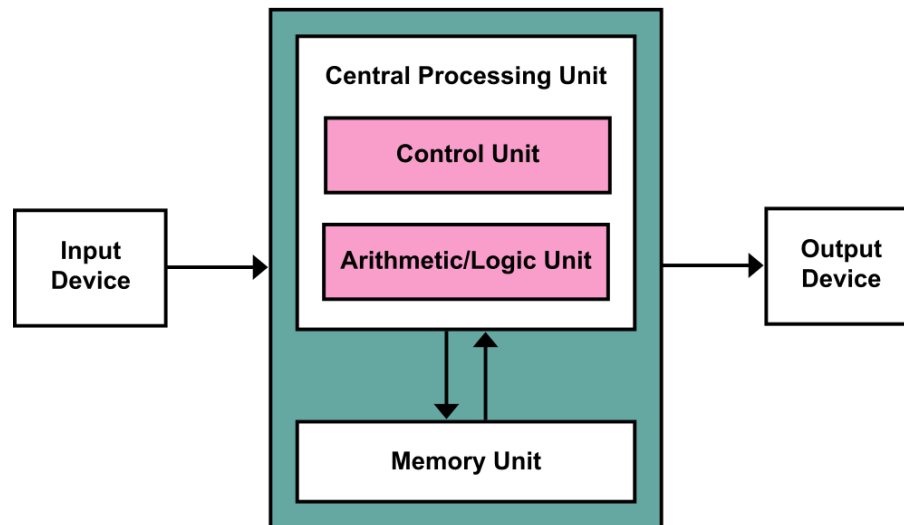
- 프로세스 (Process): 실행 중인 프로그램
 - 프로그램: 어떤 목적을 위한 명령어의 집합 (파일) ⇨ 수동적 개체
 - 프로세스: 순차적으로 실행되는 능동적 개체
- 프로세스가 작업을 완료하기 위해 필요한 자원
 - 프로세스가 시작될 때 혹은 실행 중에 할당
 - CPU, 메모리, 파일, 입출력 장치
 - 초기 데이터
- 병행 실행 (concurrency)
 - 다수 개의 프로세스를 하나의 CPU 상에서 멀티플렉싱 (multiplexing)하여 처리

프로세스 관리 (Process Management)

- 운영체제의 프로세스 관리 기능
 - CPU에 대해 여러 프로세스와 스레드(thread)의 스케줄링
 - 사용자 및 시스템 프로세스의 생성과 제거
 - 프로세스의 일시 중지(suspend)와 재실행(resume)
 - 프로세스 동기화를 위한 기법 제공
 - 프로세스 간 통신을 위한 기법 제공
 - 교착상태(deadlock)를 처리하기 위한 기법 제공

메모리 관리 (Memory Management)

- 폰 노이만 (Von Neumann) 방식 컴퓨터
 - 모든 명령과 데이터는 실행과 처리를 위해 메모리 상에 존재해야 한다.
 - 메모리는 CPU가 직접 접근할 수 있는 유일한 저장 장치
 - 메모리는 워드(word) 혹은 바이트(byte)의 큰 배열이며, 각각은 주소(address)에 의해 식별된다.



source: wikipedia

메모리 관리 (Memory Management)

- 메모리 관리
 - 어떤 프로세스와 데이터를 메모리에 적재할 지를 결정
 - 목적: CPU 사용률(utilization)과 사용자에게 대한 응답시간을 최적화
- 운영체제의 메모리 관리 기능
 - 메모리의 어느 부분이 현재 누구에 의해 사용되는 지 추적
 - 어떤 프로세스와 데이터가 메모리에 적재하고 제거될 지 결정
 - 메모리 공간을 할당(allocation)하고 회수
 - 프로세스에게 가상 메모리 공간 제공

저장장치 관리 (Storage Management)

- 운영체제의 역할
 - 저장장치의 물리적 특성을 추상화(abstraction)하여 논리적 저장 단위로 제공 ⇨ 파일 (file)

저장장치 관리 (Storage Management)

- 파일 시스템 (File System)
 - 파일(file): 파일 생성자에 의해 정의된 관련 정보의 집합체
 - 프로그램과 데이터를 표현, 비트/바이트/라인/레코드들의 연속
 - 디렉터리(directory): 파일 관리를 용이하게 하기 위한 정보
 - 파일 접근 제어: 누가 어떤 파일에 접근할 수 있는 지를 결정
- 운영체제의 파일 시스템 관리 기능
 - 파일 및 디렉터리의 생성과 제거
 - 파일 및 디렉터리를 조작하기 위한 프리미티브(primitive) 제공
 - 파일을 보조 저장장치(secondary storage)로 사상(mapping)
 - 안정적인 (비휘발성) 저장 매체에 파일을 백업

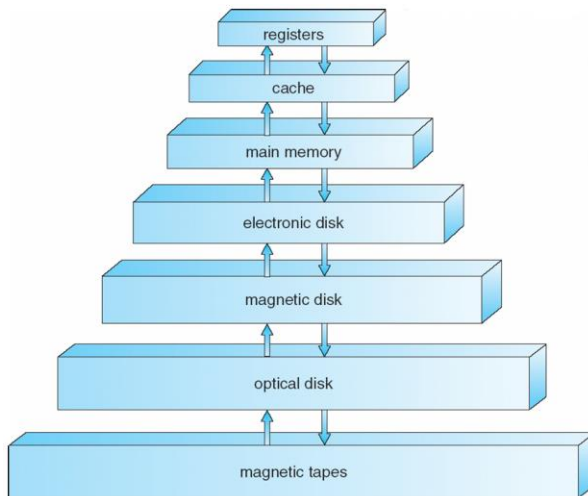
대용량 저장장치 (Mass-Storage) 관리

- 보조 저장장치 (Secondary Storage)
 - 다량의 프로그램과 데이터를 장기간 보관하기 위한 저장 장치
 - 주로 디스크 장치가 사용됨 (HDD, SSD)
 - CPU 및 메모리와의 성능 차이가 매우 크기 때문에 컴퓨터 시스템 전체의 성능에 중요한 부분을 차지 (병목 현상)
- 운영체제의 대용량 저장장치 관리 기능
 - 자유 공간 (free-space) 관리
 - 저장 공간 할당 (allocation)
 - 디스크 스케줄링

저장장치의 계층과 캐싱 (caching)

■ 캐싱 (caching)

- 낮은 단계 저장장치에 저장된 많은 정보를 높은 단계 저장장치가 제공하는 빠른 속도로 접근
- 낮은 단계 저장장치에서 자주 접근되는 일부의 정보를 높은 단계 저장장치에 저장



Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

입출력 시스템 (I/O System)

- 운영체제의 역할
 - 하드웨어 장치의 고유한 특성을 사용자에게 숨김
ex) keyboard, mouse, monitor, network adaptor, etc.
- 운영체제의 입출력 시스템 관리 기능
 - 다양한 입출력 장치의 설치 및 제거
 - 버퍼링, 캐싱, 스푼링을 포함한 입출력의 메모리 관리
 - 범용 및 특수 장치 드라이버 인터페이스

보호(Protection)와 보안 (Security)

- 보호 (Protection)
 - 운영체제가 정의한 자원에 대해 프로세스나 사용자의 접근을 제어
- 보안 (Security)
 - 내부 또는 외부의 공격으로부터 시스템을 방어하는 기능
 - 공격의 예: DOS (denial-of-service), worms, viruses, identity theft, theft of service
- 사용자 식별
 - 누가 무엇을 할 수 있는지 결정하기 위한 수단
 - 사용자 ID(보안 ID)를 모든 파일과 프로세스에 적용하여 해당 사용자의 접근을 제어
 - 그룹 ID를 이용하여 다수의 사용자 접근 제어를 한번에 처리
 - 특정 작업을 위해 높은 권한의 유효 사용자 ID를 획득

수업 내용

- 운영체제의 정의
- 운영체제의 역할
 - 프로세스 관리, 메모리 관리, 저장장치 관리, 보호와 보안
- 리눅스 운영체제 소개
- 가상 머신과 리눅스 설치
 - VirtualBox와 Xubuntu

리눅스 운영체제 소개

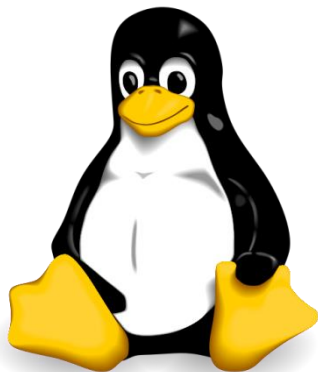
- 유닉스 (Unix)
 - 범용의 multiuser, multitasking 운영체제
 - 1969년 AT&T의 벨 연구소에서 Ken Thompson이 개발
 - 1973년에 전체를 C 언어로 구현
 - 현재는 교육 및 연구용으로 주로 사용



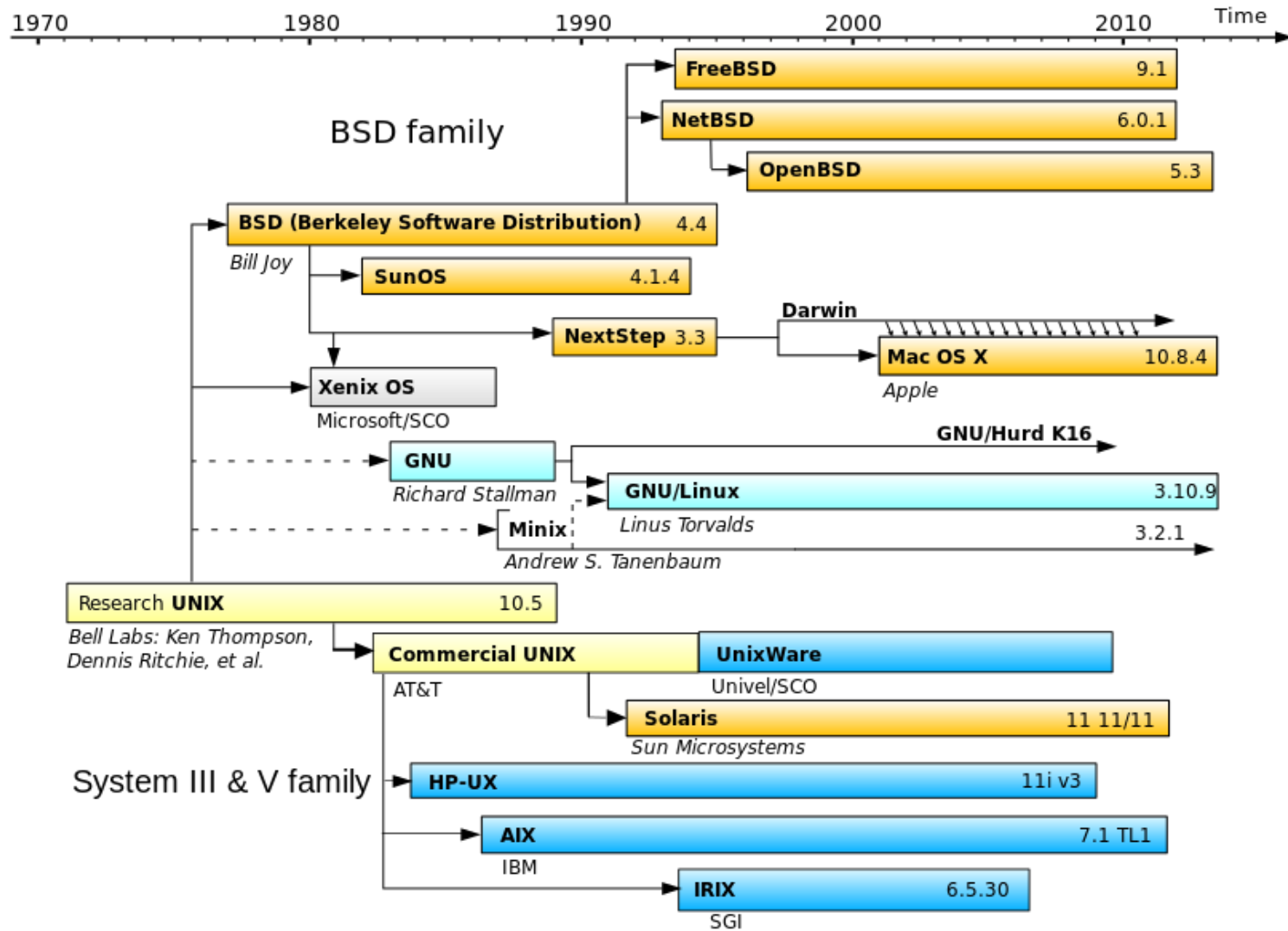
Ken Thompson & Dennis Ritchie [PDP-11]
source: wikipedia

리눅스 운영체제 소개

- 리눅스 (Linux)
 - 오픈 소스 기반의 강력한 운영체제 (GPL - GNU Public License)
 - GNU is Not UNIX (Richard Stallman, 1985)
 - 1991년 헬싱키 대학 학생 Linus Torvalds가 개발
 - Minix 운영체제의 한계로 부터 시작
 - 커널 공개 이후 많은 프로그래머들이 공동으로 개발
 - 데스크톱과 서버, 내장형 시스템의 운영체제로 널리 사용



유닉스와 리눅스의 역사



source: wikipedia

리눅스 운영체제 소개

- 리눅스(Linux)와 GNU/Linux
 - 리눅스는 원래 운영체제 커널(kernel)을 의미
 - GNU 프로젝트 기반의 전체 시스템 (GNU/Linux)
 - 커널, 라이브러리, 쉘, 그래픽 UI 등
- 리눅스 배포판
 - 리눅스 커널 기반의 전체 운영체제 패키지



xubuntu 

fedora 

 chrome

 CentOS

ubuntu 

 redhat.

 openSUSE™

리눅스 운영체제의 특징

- 대부분은 유닉스와 유사
 - 다중 사용자 (multi-user)
 - 선점형 시분할 멀티태스킹 (multitasking)
 - 프로세스와 스레드
- 멀티 프로세서 지원 (SMP, NUMA)
- 높은 하드웨어 이식성
- 가상 파일 시스템
- 다양한 배포판
- 오픈 소스 소프트웨어
- ...

리눅스와 표준 (Standard)

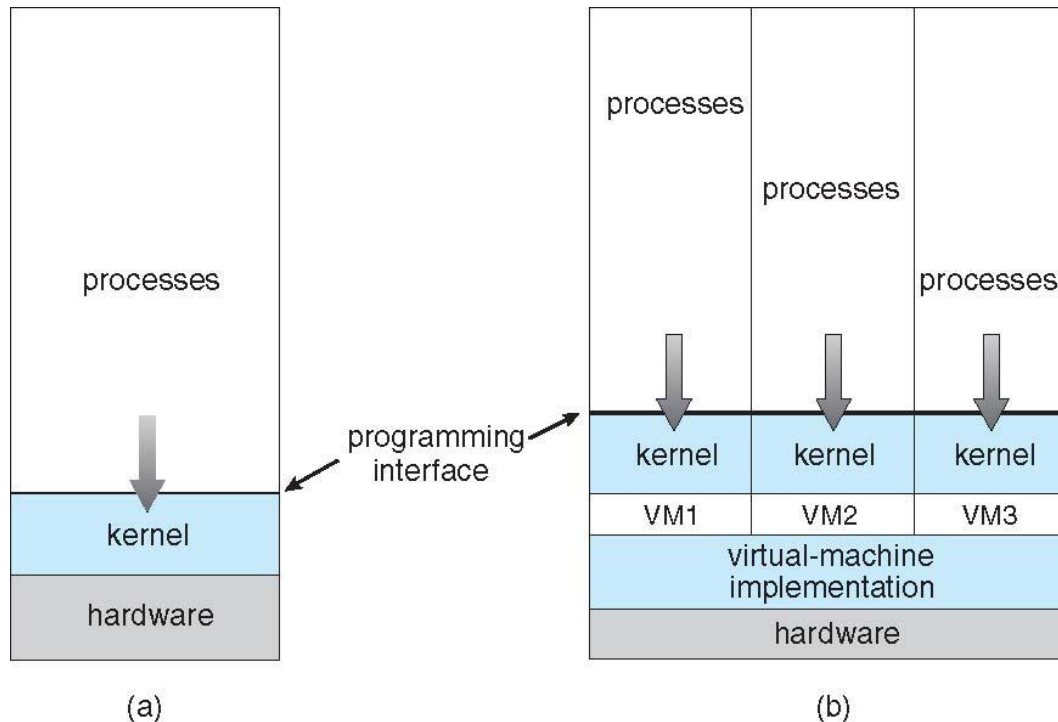
- 표준(Standard)과 인증(Certification)
 - 기술의 구현과 사용 방법에 대한 합의된 규칙
 - 다양한 기술 구현들 사이의 호환성을 위해 필수적인 요소
- 리눅스와 관련된 주요 표준
 - POSIX (Portable Operating System Interface)
 - IEEE 1003.1-2004 (POSIX.1)로 정의된 C 프로그래밍 API
 - SUS(Single UNIX Specification) 표준에 포함되어 있음.
 - LSB (Linux Standard Base)
 - 리눅스 커널 API 표준, 배포판들 사이의 표준화 기준
 - C 언어 표준
 - 리눅스의 gcc는 ISO C99를 지원

수업 내용

- 운영체제의 정의
- 운영체제의 역할
 - 프로세스 관리, 메모리 관리, 저장장치 관리, 보호와 보안
- 리눅스 운영체제 소개
- 가상 머신과 리눅스 설치
 - VirtualBox와 Xubuntu

가상 머신 (Virtual Machine)

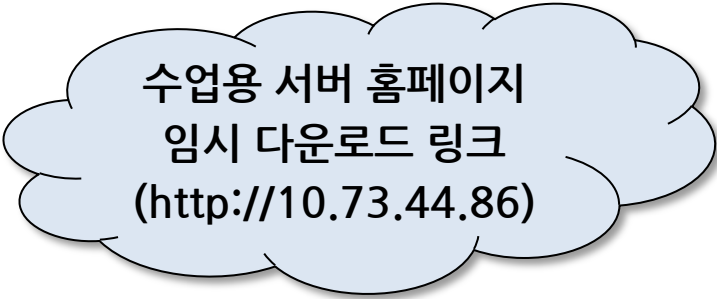
- 컴퓨터의 하드웨어를 추상화(abstraction)하여 여러 개의 독립적인 컴퓨터 시스템 실행 환경을 제공
 - 호스트(host)는 각각의 게스트(guest)에게 자신만의 개별적인 컴퓨터 시스템을 사용하고 있는 환상(illusion)을 만들어 준다.



가상 머신 설치 파일과 운영체제 이미지 파일

■ VirtualBox

- Oracle에서 제공하는 무료 가상 머신 소프트웨어
- <https://www.virtualbox.org/wiki/Downloads>
- VirtualBox-4.3.28



수업용 서버 홈페이지
임시 다운로드 링크
(<http://10.73.44.86>)

■ Xubuntu 14.04 LTS

- 리눅스 운영체제 배포판
- <http://xubuntu.org/getxubuntu/>
- xubuntu-14.04.2-desktop-i386.iso (32비트 운영체제)

주요 설치 과정

- VirtualBox 프로그램 설치
 - VirtualBox에서 리눅스를 위한 가상 머신 생성 (Linux/Ubuntu)
 - RAM: 1GB, 저장장치: 16GB
 - 네트워크: 호스트 전용 어댑터 설치 (어댑터 2 사용하기)
 - 호스트와 리눅스 사이의 네트워크

- Xubuntu 14.04 LTS 설치
 - 배포판 iso 파일을 이용
 - 실제 PC에서 설치하는 과정과 동일 (CD 부팅)
 - 빠른 설치를 위해 업데이트는 선택하지 않음
 - 기본 계정(사용자)은 편하게 사용할 수 있는 것을 권장

주요 설치 과정

- Xubuntu 설치 후 작업
 - 소프트웨어 업데이트 (최신 프로그램 패키지)
 - 설정 - 소프트웨어 업데이트 도구
 - 속도가 느리면 업데이트 서버 변경 (ftp.daum.net)
 - 설정 - 소프트웨어 & 업데이트 (다운로드 위치 - 기타)
 - 게스트 확장 설치
 - 가상 머신 사용의 편의성 제공 (화면 크기 조정, 클립보드 복사 등)
 - 부팅된 상태에서 VirtualBox 장치 - 게스트 확장 설치 메뉴 이용
 - CD 삽입되면, “autorun.sh” 실행
 - 재부팅 후 가상 머신 창 크기로 효과 확인

주요 설치 과정

- Xubuntu 설치 후 작업
 - 한글 사용 환경 설정
 - 키보드 입력기 설치
 - 터미널 에뮬레이터 실행하여 명령 입력
 - `sudo apt-get install ibus-hangul`
 - 설정-언어지원 (키보드 입력기 'IBus' 선택)
 - 설정-키보드 입력기 (한영 전환 키 설정)

주요 설치 과정

- 기타 설정
 - 터미널 글꼴 변경 (편집-기본 설정)
 - 공유 폴더 설정 (VB 메뉴 장치-공유 폴더 설정)
 - xubuntu 터미널에서 디스크 마운트 명령 사용
 - `sudo mount -t vboxsf 공유폴더이름 /mnt`
 - 클립보드 공유 설정 (VB 메뉴 장치-클립보드 공유)
- Xubuntu의 호스트 전용 어댑터 인터넷 주소 확인
 - 터미널 명령 “ifconfig” 입력
 - ‘eth1’의 ‘inet addr’ 확인
- 호스트 운영체제에 외부 접근 터미널 설치
 - PuTTY, WinSCP 등의 secure shell 클라이언트

기타 수업 및 실습 환경

- 리눅스 서버 (ubuntu 64bit)
 - 10.73.44.86 (secure shell 접속)
 - 계정: os1502, 암호: next1234
 - 예습 및 보충 강의 동영상 참고 자료
 - <http://10.73.44.86/os>
- 터미널 접속 및 FTP 클라이언트 프로그램 (Secure Shell)
 - PuTTY
 - <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
 - SSHSecureShellClient-3.2.9
 - Mac 터미널에서 “ssh” 명령 사용
 - **ssh** 사용자_id@10.73.44.86

리눅스 관련 온라인 참고 자료

- Open source software Learning Community (OLC)
 - <http://olc.oss.kr/main.jsp>
 - 소프트웨어 관련 다양한 온라인 강의 자료 제공
 - Concept of Linux and Basic Operations (이민석)
 - System - Others 카테고리
 - Linux System Management (구용민)
Linux System Programming (김두현)
Kernel of Linux (고건)
 - System - OS 카테고리
 - C 언어 프로그래밍 (이상엽)
 - Language&Tool - Language 카테고리

운영체제 과목의 주요 수업 내용

- 리눅스 운영체제 사용법
 - 셸과 주요 명령어
 - 리눅스의 프로그램 개발 도구 (vi, gcc, gdb, make 등)
- 운영체제 개관
 - 운영체제의 역할과 기본 동작
 - 운영체제의 프로그래밍 인터페이스 (시스템 호출)
- 파일 시스템
 - 파일 시스템의 역할과 저장 구조
 - 파일 입출력 프로그래밍 인터페이스

운영체제 과목의 주요 수업 내용

- 프로세스와 스레드
 - 프로세스와 스레드의 개념
 - 프로세스 및 스레드 제어 프로그래밍 인터페이스 (API)
 - 프로세스 간 통신
 - 프로세스 스케줄링과 스케줄링 알고리즘

- 프로세스 동기화와 데드락
 - 동기화의 필요성과 요구 조건
 - 여러 가지 동기화 방법
 - 데드락의 필요 조건과 방지 방법

운영체제 과목의 주요 수업 내용

- 메모리 관리와 가상 메모리
 - 메모리 주소 공간의 개념과 주소 변환
 - 페이징과 페이지 교체 알고리즘
 - 동적 메모리 할당과 메모리 관리 프로그래밍