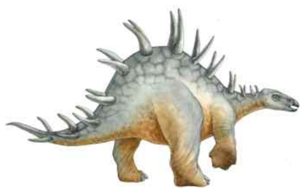
Chapter 1: Introduction

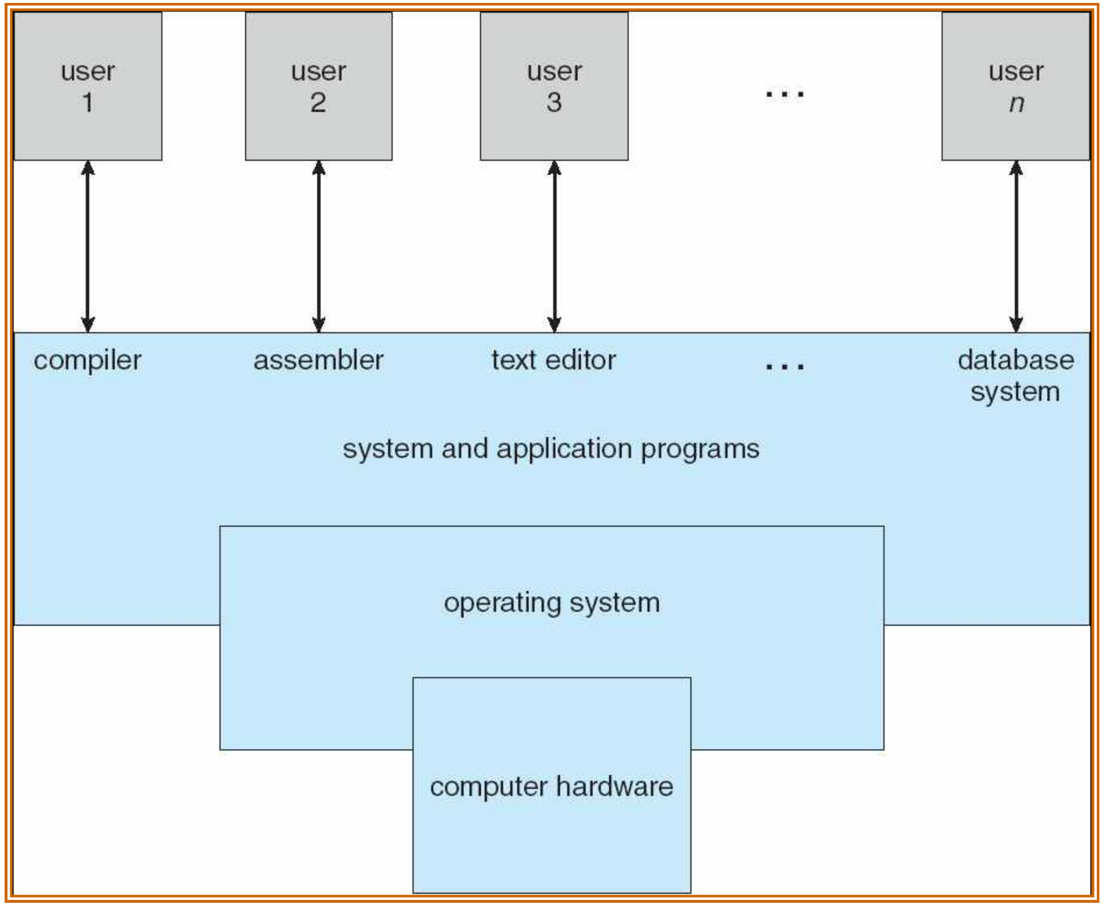
Operating System이란?

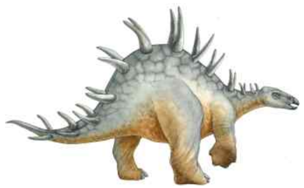
- computer hardware와 computer user 사이의 중재자 역할을 수행하는 프로그램
  - The job of the OS is to adapt to hardware. Examples: MS-DOS/Windows, MacOS, Unix, and many more

- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.



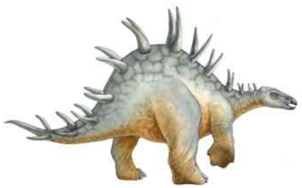

Abstract View of System Components





Computer System Components

- Computer system의 네가지 구성요소
  - Hardware –basic computing resources를 제공
    - CPU, memory, I/O devices
  - Operating system
    - 다양한 응용과 사용자들간의 하드웨어 사용을 Controls하고 coordinates 함
  - Application programs
    - 사용자의 계산 문제를 풀기 위해 시스템 자원이 어떻게 사용될지를 정의
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers



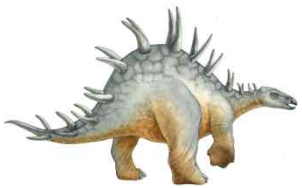
Operating System Definitions

- 운영체제의 정의

- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer

최초의 컴퓨터는?

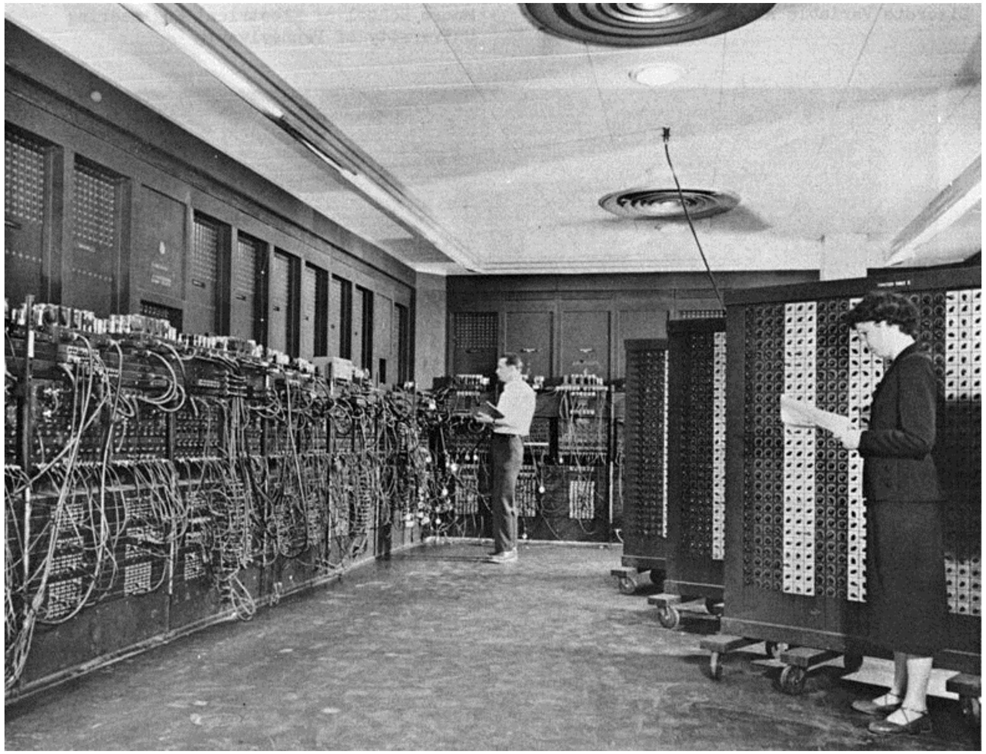


OS History

- Phase 0 : No OS
- Phase 1 : Batch System(일괄처리 시스템)
- Phase 2: Multi-programmed System(다중 프로그램 시스템)
- Phase 3 : Time Sharing System(시분할 시스템)
- Phase 4 : Personal Computing
- tc
  - Real-time System
  - Multi-Processing System
  - Distributed System

OS History – ENIAC의 프로그래밍과 OS?

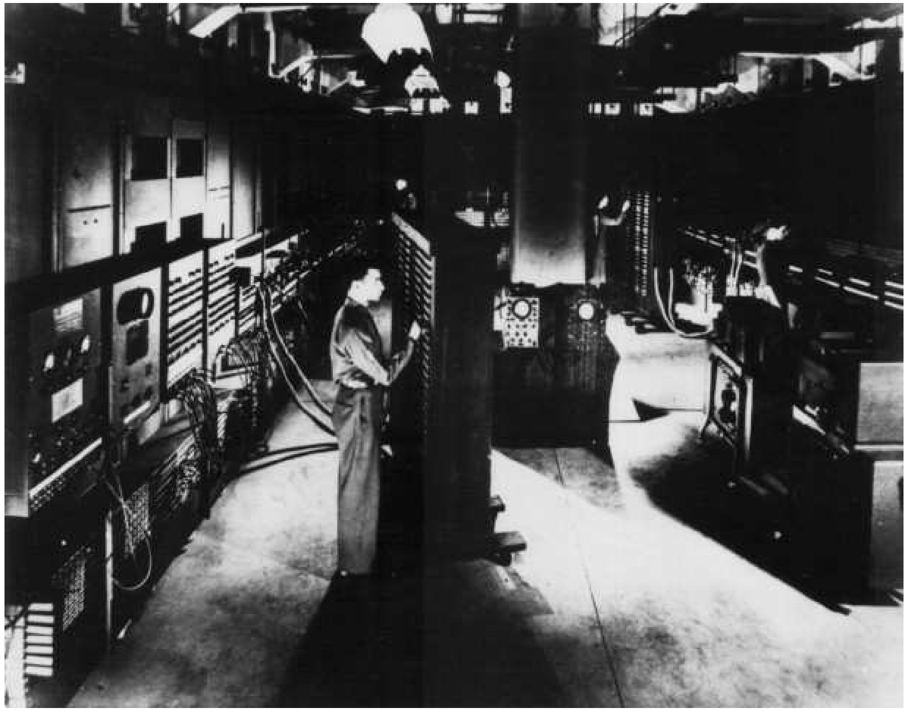
- 최초의 다목적 컴퓨터(ENIAC)에서 프로그래밍은?
  - 1946-1955
  - 10진수 내장 프로그래밍



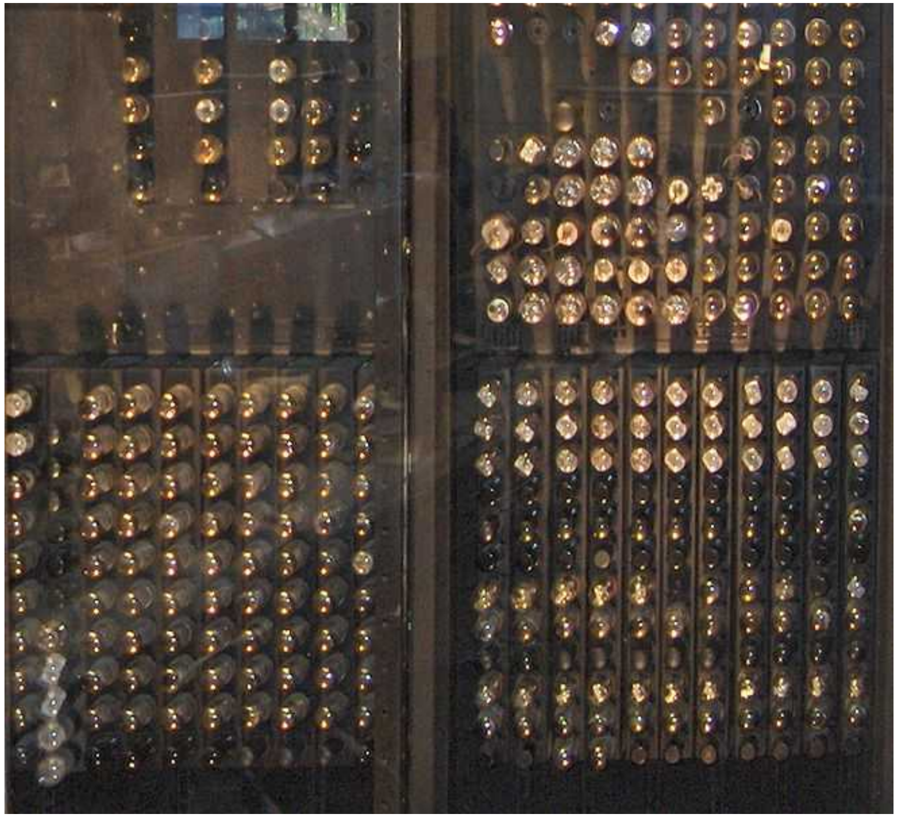
Glen Beck (background) and Betty Snyder (foreground) program the ENIAC in BRL building 328

http://en.wikipedia.org/wiki/ENIAC

OS History – ENIAC의 프로그래밍과 OS?



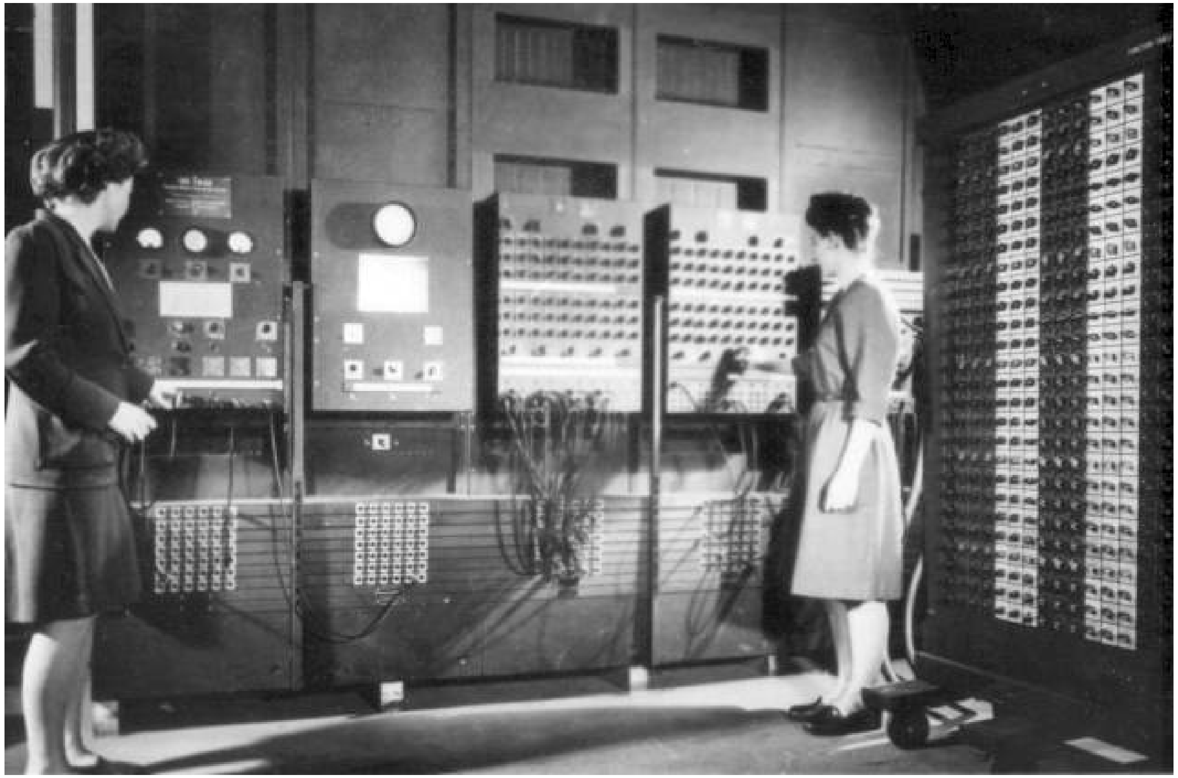
Irwin Goldstein (foreground) sets the switches on one of the ENIAC's function tables at the Moore School of Electrical Engineering



Detail of the back of a panel of ENIAC, showing vacuum tubes.

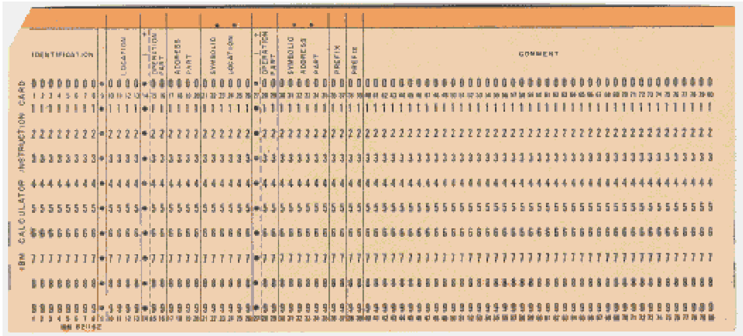
OS History – ENIAC의 프로그래밍과 OS?

- 최초의 다목적 컴퓨터에서 Operating 은?

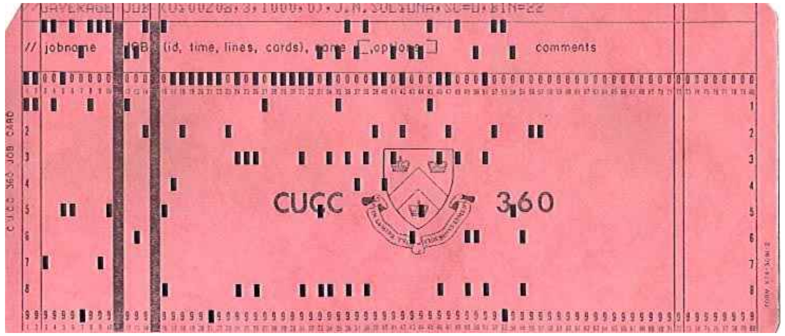


Two women operating the ENIAC's main control panel while the machine was still located at the Moore School

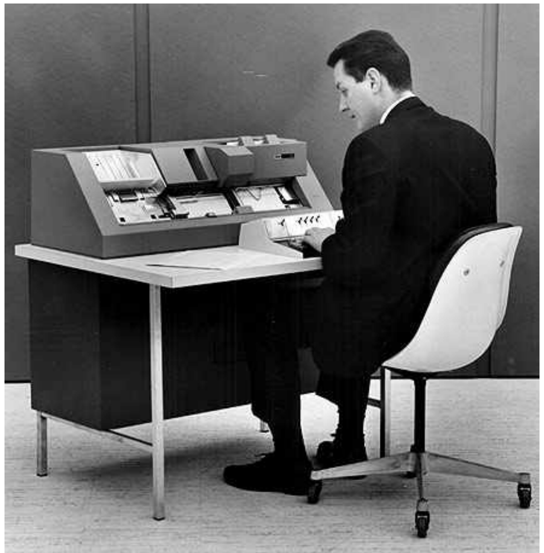
OS History-Batch System



Punch Card의 예



Punch Card의 예

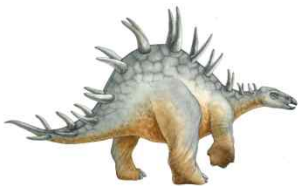


Punch Card 작업을 하는 사람(IBM)

Punch Card 입력 컴퓨터(IBM)

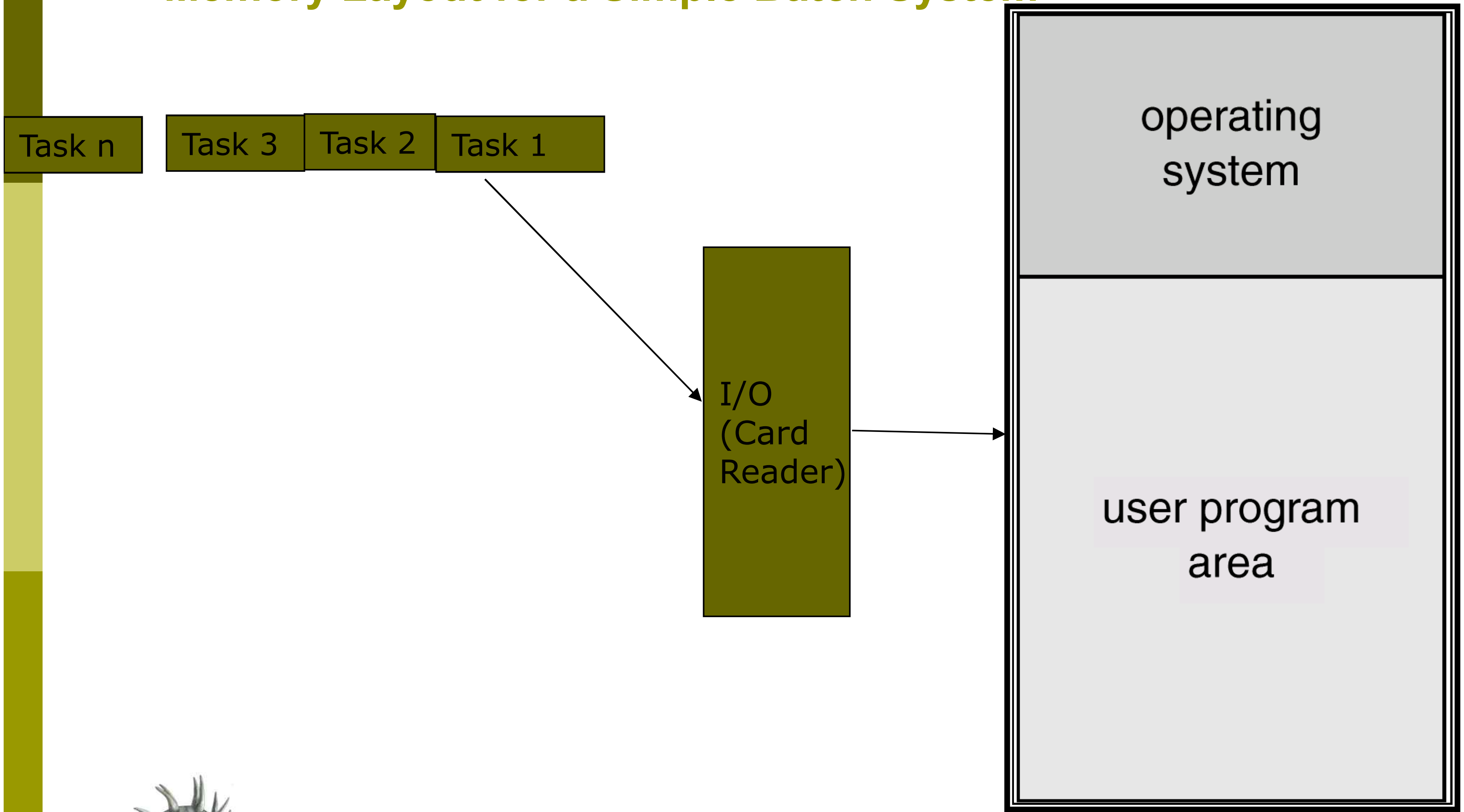
OS History-Batch System

- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
  - 예)CPU와 I/O간(Card Reader)의 처리속도 차이 존재
    - CPU : 초당 수천개
    - Card Reader : 초당 20장(분당 1200장)
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor



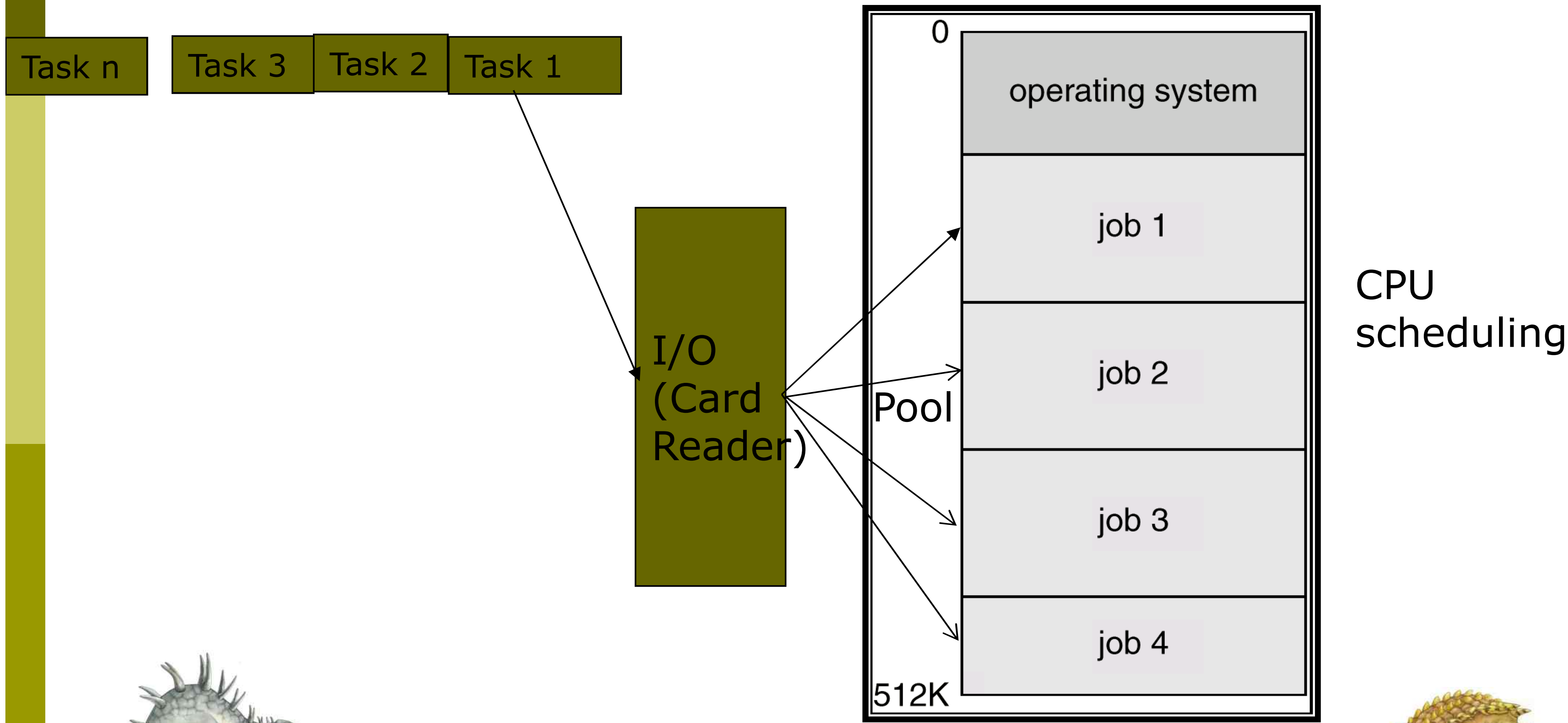
OS History - Batch System

Memory Layout for a Simple Batch System



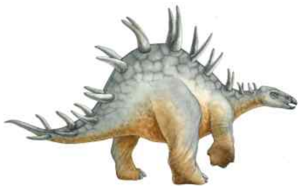
OS History – Multiprogrammed Systems

Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.



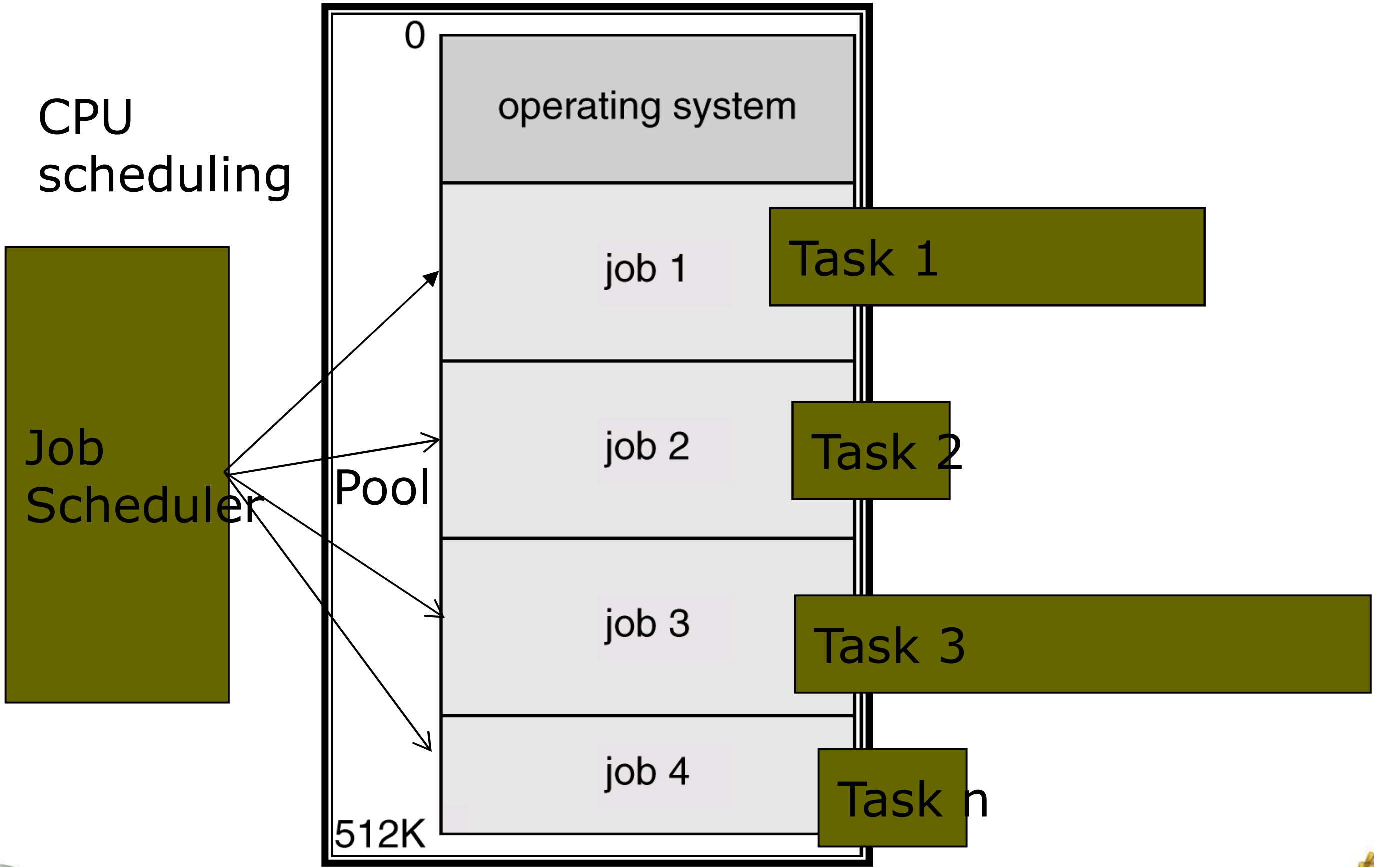
OS History – Multiprogrammed Systems

- OS Features Needed for Multiprogramming
  - I/O routine supplied by the system.
  - Memory management
    - the system must allocate the memory to several jobs.
  - CPU scheduling
    - the system must choose among several jobs ready to run.
  - Allocation of devices.



OS History - Time-Sharing Systems

Minimize Response Time!

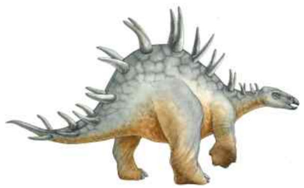


Time-Sharing Systems–Interactive Computing

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
- A job swapped in and out of memory to the disk.
  - Swap-In
  - Swap-Out
- On-line communication between the user and the system is provided;
  - when the operating system finishes the execution of one command, it seeks the next “control statement” from the user’s keyboard.
- On-line system must be available for users to access data and code.

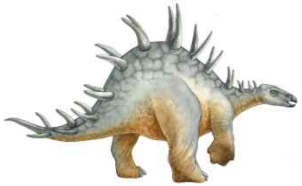
Desktop Systems

- *Personal computers* – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system’ often individuals have sole use of computer and do not need advanced CPU utilization of protection features.
- May run several different types of operating systems (Windows, MacOS, UNIX, Linux)



Parallel Systems

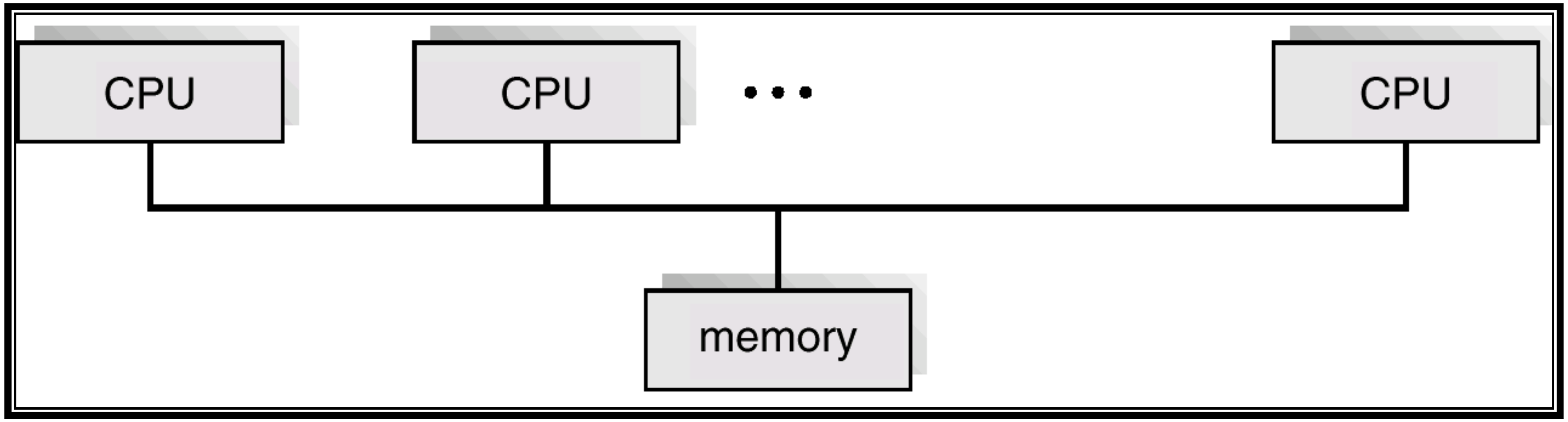
- Multiprocessor systems with more than on CPU in close communication.
- *Tightly coupled system* – processors share memory and a clock; communication usually takes place through the shared memory.
- Advantages of parallel system:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - graceful degradation
    - fail-soft systems



Parallel Systems (Cont.)

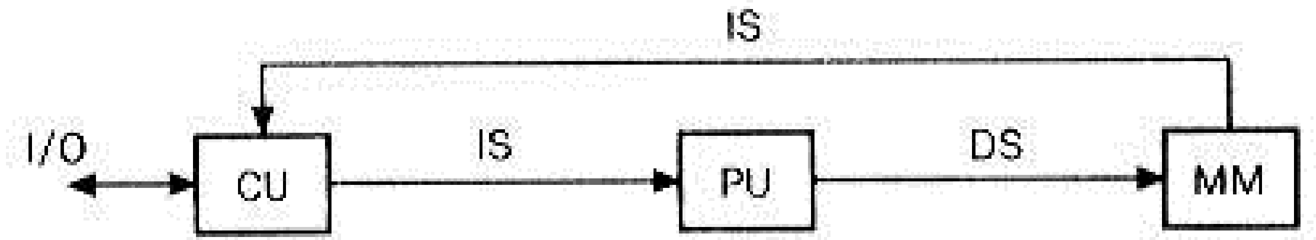
- *Symmetric multiprocessing (SMP)*
  - Each processor runs and identical copy of the operating system.
  - Many processes can run at once without performance deterioration.
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
  - More common in extremely large systems

Symmetric Multiprocessing Architecture

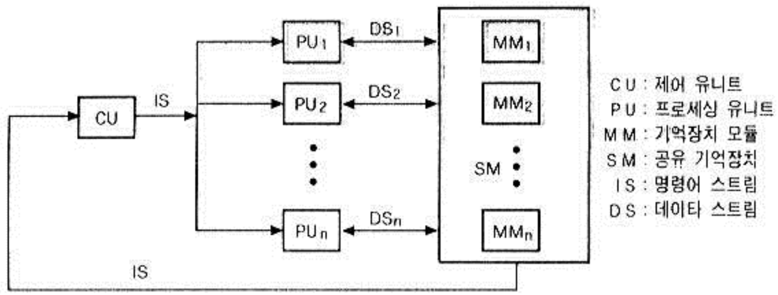


대칭적 다중처리(SMP : Symmetric Multiprocessing)

- 컴퓨터 시스템의 분류 (by Flynn)
  - Single Instruction Single Data (SISD) stream
    - 단일 처리기가 한 메모리에 저장된 데이터를 처리하기 위해 단일 명령 스트림을 수행

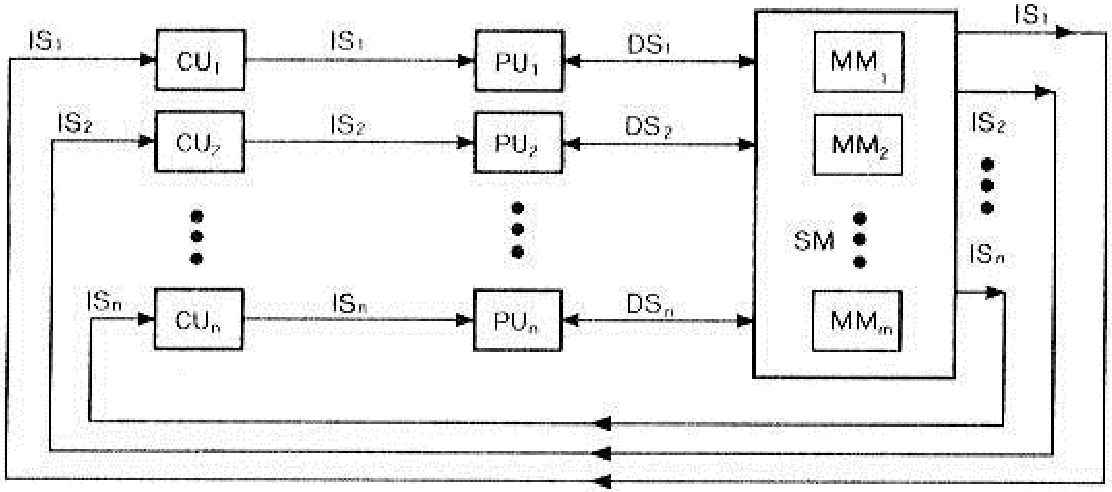


  - Single Instruction Multiple Data (SIMD) stream
    - 각 (동일) 명령이 서로 다른 데이터 집합에 대하여 서로 다른 처리기에 의해 수행
    - 벡터 및 배열 처리기 (vector and array processors)
    - Pentium 처리기의 superscalar 구조



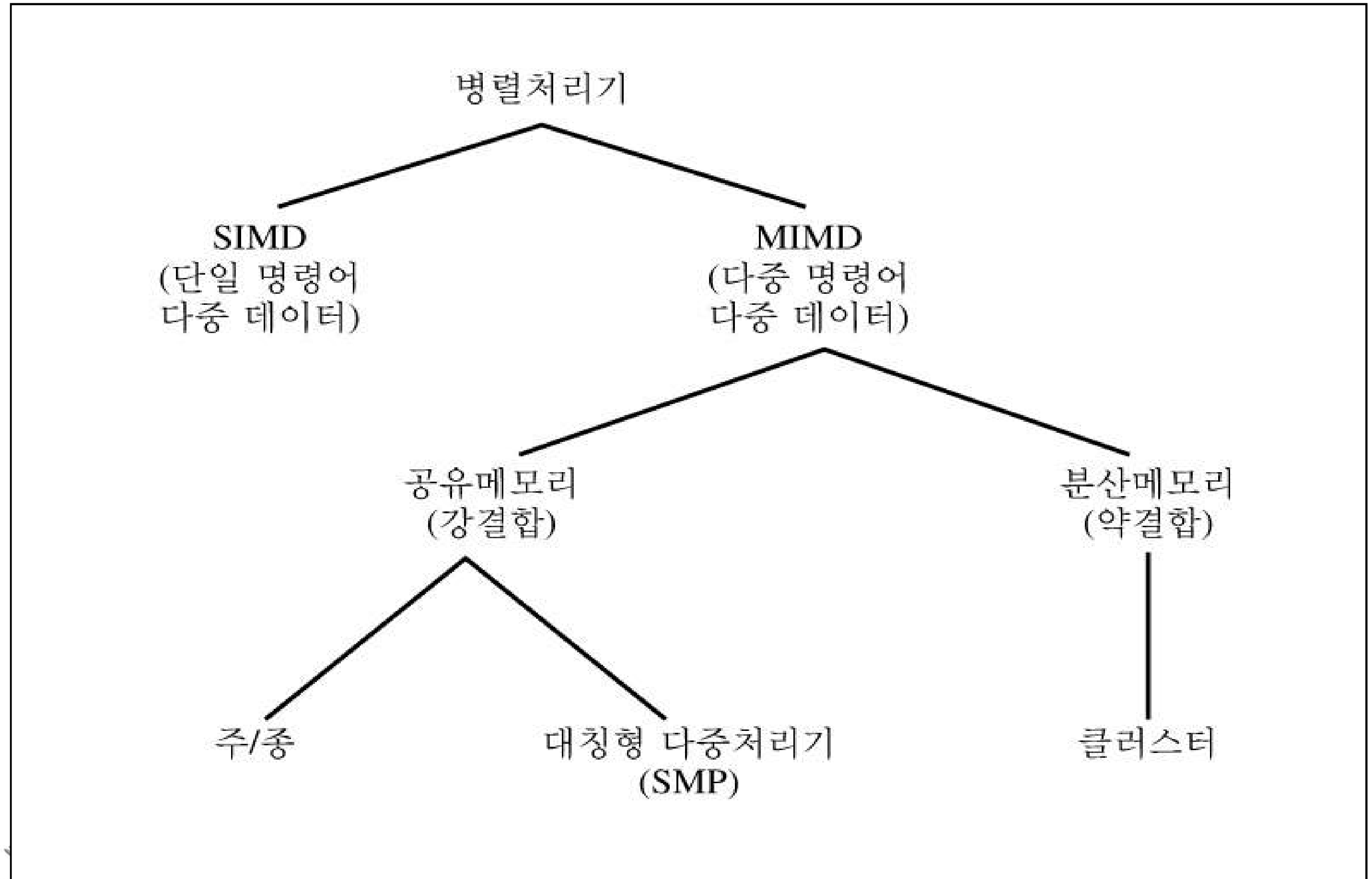
대칭적 다중처리 (계속)

- 컴퓨터 시스템의 분류 (by Flynn) (계속)
  - Multiple Instruction Single Data (MISD) stream
    - (같은) 일련의 데이터가 처리기들의 집합에 전송되고, 각 처리기는 서로 다른 명령을 수행
    - 지금까지 구현된 적이 없음
  - Multiple Instruction Multiple Data (MIMD) stream
    - 다수의 처리기가 서로 다른 데이터 집합에 대하여 서로 다른 일련의 명령어들을 동시에 수행



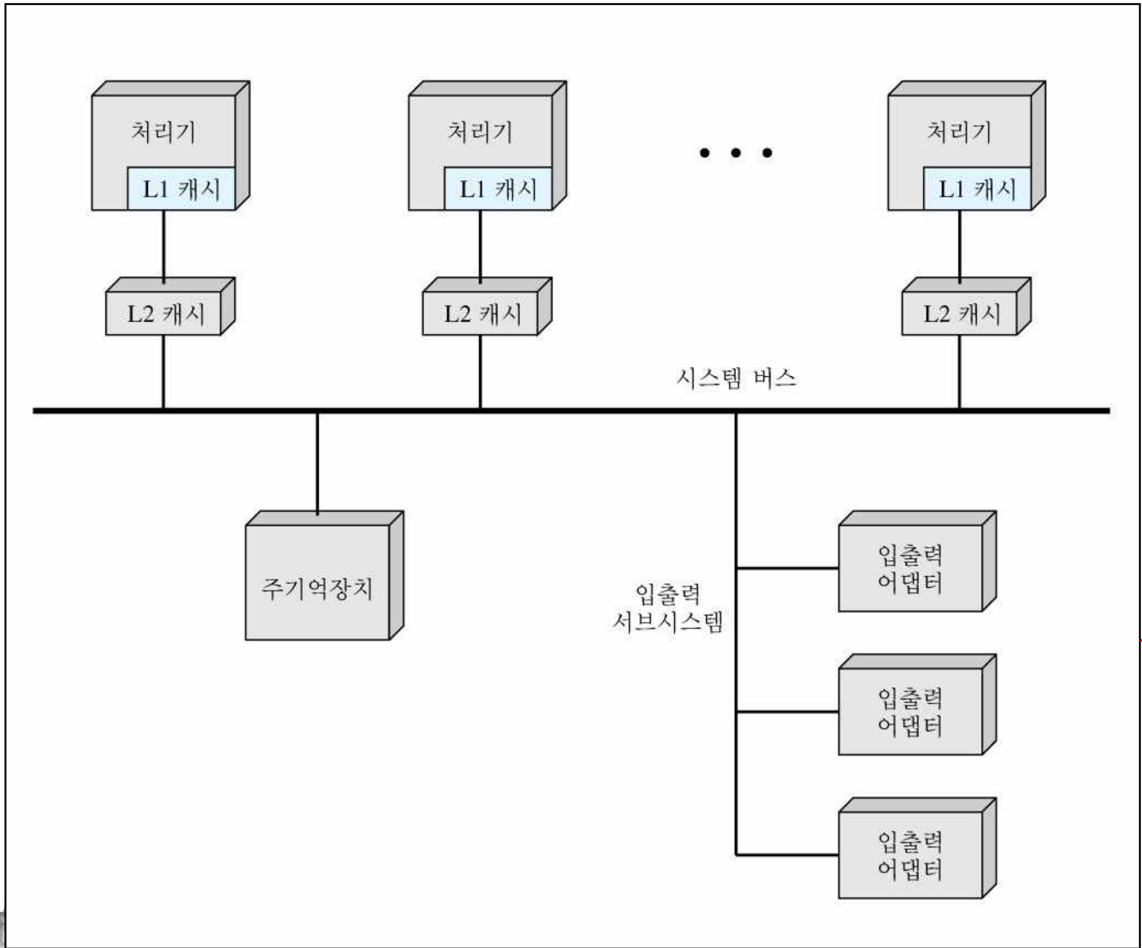
대칭적 다중처리 (계속)

- 병렬 처리기들의 분류 (Categories of Parallel Processors)



대칭적 다중처리 (계속)

- SMP 구성

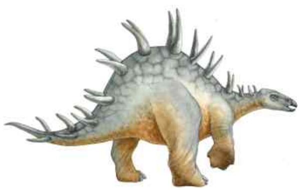


캐시 일관성 문제

24

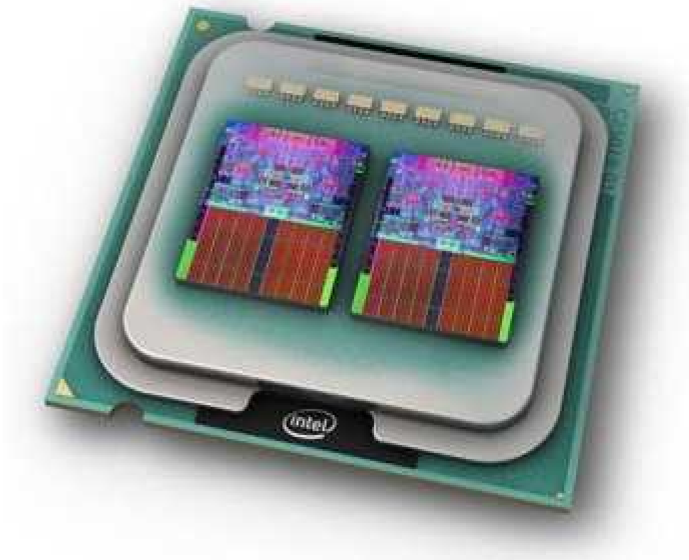
대칭적 다중처리 (계속)

- SMP를 위한 OS 설계 시 고려 사항
  - 목적: 사용자가 다중프로그래밍 단일처리기 시스템(multiprogramming uniprocessor system)과 동일한 관점을 가질 수 있게 자원을 관리
  - 동시적 병행(simultaneous concurrent) 프로세스 또는 스레드
    - 재진입(reentrant) 커널 코드
    - 분리된(separated) 커널 자료 구조
  - SMP를 위한 프로세스(스레드) 스케줄링
    - 전역 큐 / 지역 큐
    - 집단 스케줄링(gang scheduling)
    - 캐쉬 친화성(cache affinity)

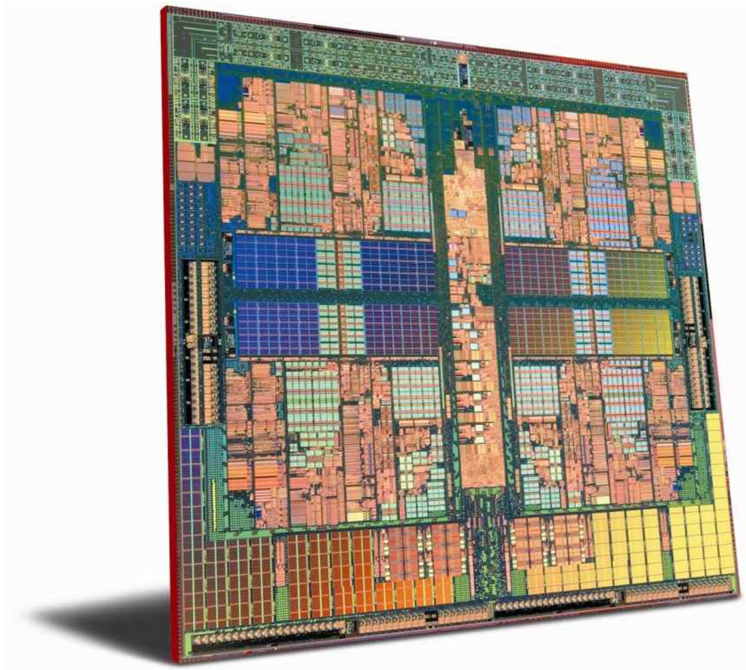


대칭적 다중처리 (계속)

- SMP를 위한 OS 설계 시 고려 사항(계속)
  - 동기화
    - 상호배제(mutual exclusion)
    - 사건 순서화(event ordering)
  - 메모리 관리
    - 다중포트 메모리(multiport memory): dualport memory
  - 신뢰성 및 결함 허용 (fault tolerance)
    - 이주 (migration)



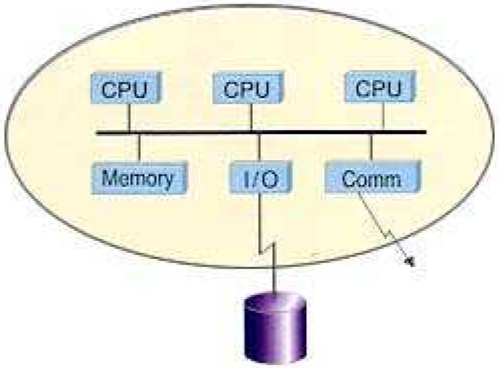
Intel Core 2 Quad(Q6600)



AMD Quad Core CPU(Opteron)

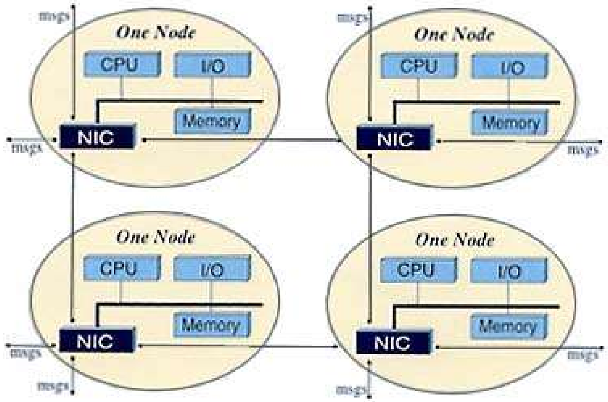
병렬 컴퓨터

- SMP



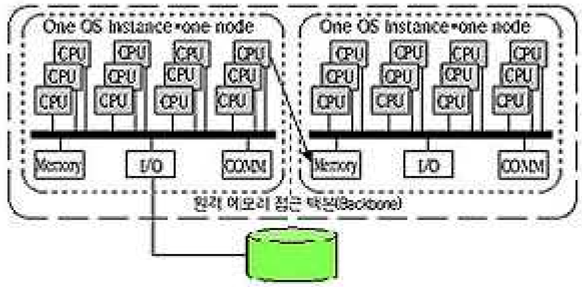
- MPP

(Massive Parallel



- NUMA

(**Non-Uniform Memory Access**)

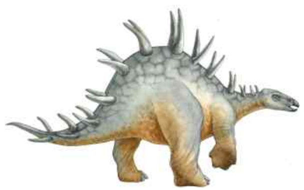


Distributed Systems

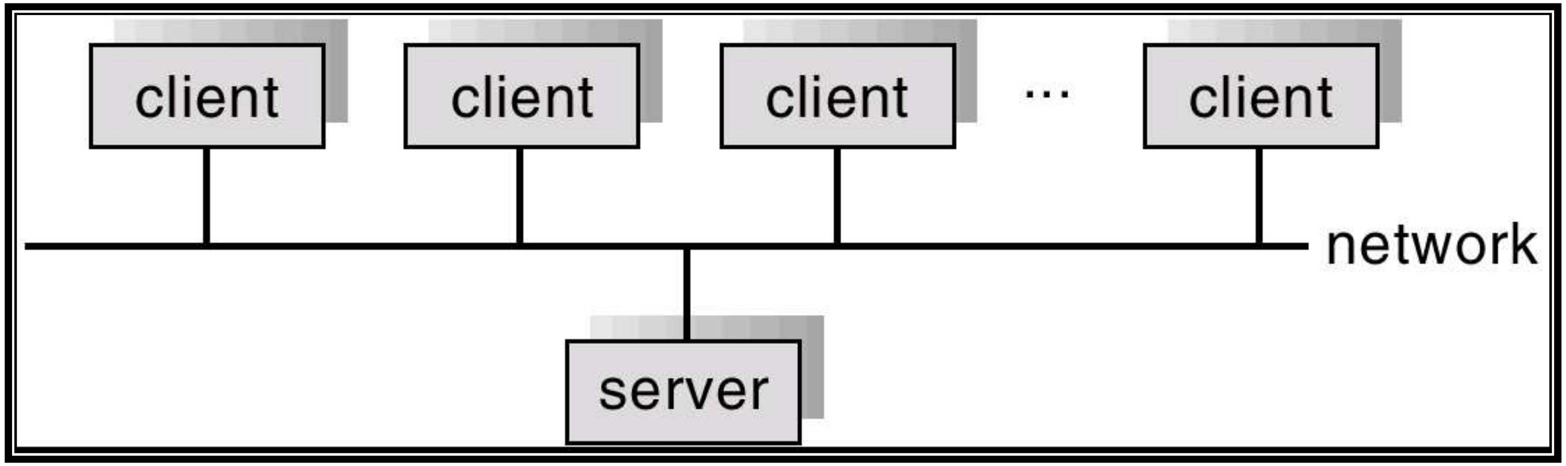
- Distribute the computation among several physical processors.
- *Loosely coupled system* – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications

Distributed Systems (cont)

- Requires networking infrastructure.
- Local area networks (LAN) or Wide area networks (WAN)
- May be either client-server or peer-to-peer systems.



General Structure of Client-Server

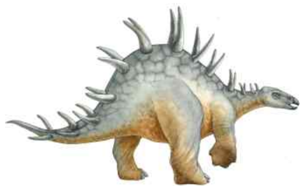


Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- Real-Time systems may be either *hard* or *soft* real-time.

Real-Time Systems (Cont.)

- Hard real-time:
  - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- Soft real-time
  - Limited utility in industrial control of robotics
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.



Handheld Systems

- Personal Digital Assistants (PDAs)
- Cellular telephones
- Issues:
  - Limited memory
  - Slow processors
  - Small display screens.




Clustered Systems

- Clustering allows two or more systems to share storage.
- Provides high reliability.
- *Asymmetric clustering*: one server runs the application while other servers standby.
- *Symmetric clustering*: all N hosts are running the application.

http://www.top500.org/

Desing of Cluster Computer : http://www.scl.ameslab.gov/Projects/parallel_computing/cluster_design.html
Examples of Cluster Computer : http://www.scl.ameslab.gov/Projects/parallel_computing/cluster_examples.html

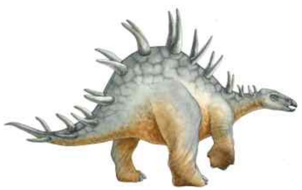
TOP 500

- Top 500
  - www.top500.org

FLOPS (or **flops** or **flop/s**) is an acronym meaning **FL**oating point **OP**erations per **S**econd

Computer Performance	
Name	**FLOPS**
yottaFLOPS	10^{24}
zettaFLOPS	10^{21}
exaFLOPS	10^{18}
petaFLOPS	10^{15}
teraFLOPS	10^{12}
gigaFLOPS	10^{9}
megaFLOPS	10^{6}
kiloFLOPS	10^{3}

107.55 GFLOPS (Intel Core i7 980 XE)

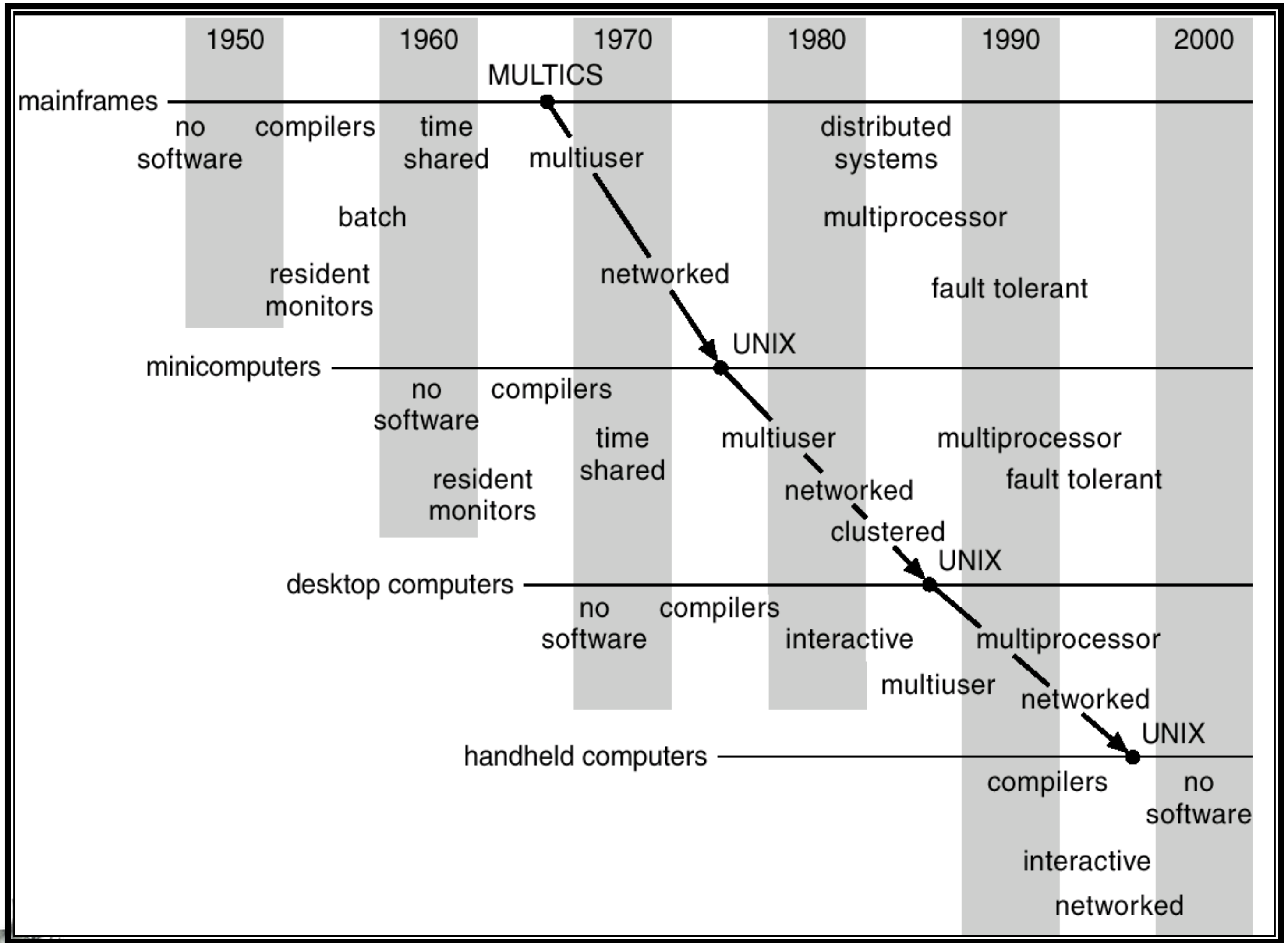


November 2011

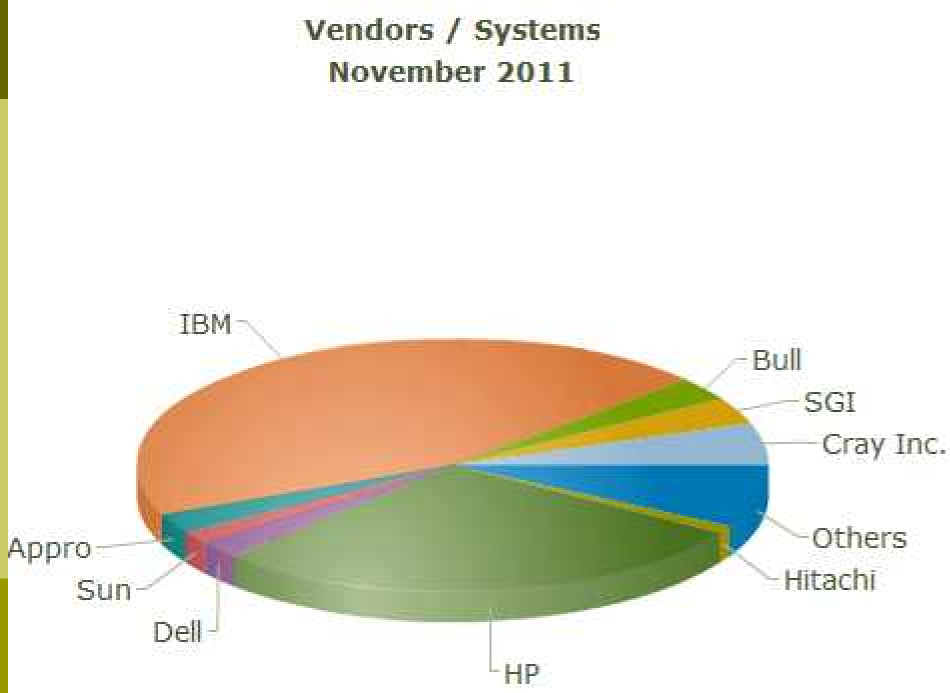
Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect / 2011 Fujitsu	705024	10510.00	11280.38	12659.9
2	National Supercomputing Center in Tianjin China	NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050 / 2010 NUDT	186368	2566.00	4701.00	4040.0
3	DOE/SC/Oak Ridge National Laboratory United States	Cray XT5-HE Opteron 6-core 2.6 GHz / 2009 Cray Inc.	224162	1759.00	2331.00	6950.0
4	National Supercomputing Centre in Shenzhen (NSCS) China	Dawning TC3600 Blade System, Xeon X5650 6C 2.66GHz, Infiniband QDR, NVIDIA 2050 / 2010 Dawning	120640	1271.00	2984.30	2580.0
5	GSIC Center, Tokyo Institute of Technology Japan	HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows / 2010 NEC/HP	73278	1192.00	2287.63	1398.6
6	DOE/NNSA/LANL/SNL United States	Cray XE6, Opteron 6136 8C 2.40GHz, Custom / 2011 Cray Inc.	142272	1110.00	1365.81	3980.0
7	NASA/Ames Research Center/NAS United States	SGI Altix ICE 8200EX/8400EX, Xeon HT QC 3.0/Xeon 5570/5670 2.93 Ghz, Infiniband / 2011 SGI	111104	1088.00	1315.33	4102.0
8	DOE/SC/LBNL/NERSC United States	Cray XE6, Opteron 6172 12C 2.10GHz, Custom / 2010 Cray Inc.	153408	1054.00	1288.63	2910.0
9	Commissariat a l'Energie Atomique (CEA) France	Bull bullx super-node S6010/S6030 / 2010 Bull	138368	1050.00	1254.55	4590.0
10	DOE/NNSA/LANL United States	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband / 2009 IBM	122400	1042.00	1375.78	2345.0
11	National Institute for Computational Sciences/University of Tennessee United States	Cray XT5-HE Opteron Six Core 2.6 GHz / 2011 Cray Inc.	112800	919.10	1173.00	3090.0
12	HWW/Universitaet Stuttgart Germany	Cray XE6, Opteron 6276 16C 2.30 GHz, Cray Gemini interconnect / 2011 Cray Inc.	113472	831.40	1043.94	
13	Forschungszentrum Juelich (FZJ) Germany	Blue Gene/P Solution / 2009 IBM	294912	825.50	1002.70	2268.0
14	National Supercomputing Center in Jinan China	Sunway BlueLight MPP, ShenWei processor SW1600 975.00 MHz, Infiniband QDR / 2011 NRCPCET	137200	795.90	1070.16	1074.0
15	Lawrence Livermore National Laboratory United States	Xtreme-X GreenBlade GB512X, Xeon E5 (Sandy Bridge - EP) 8C 2.60GHz, Infiniband QDR / 2011 Appro	46208	773.70	961.13	924.2

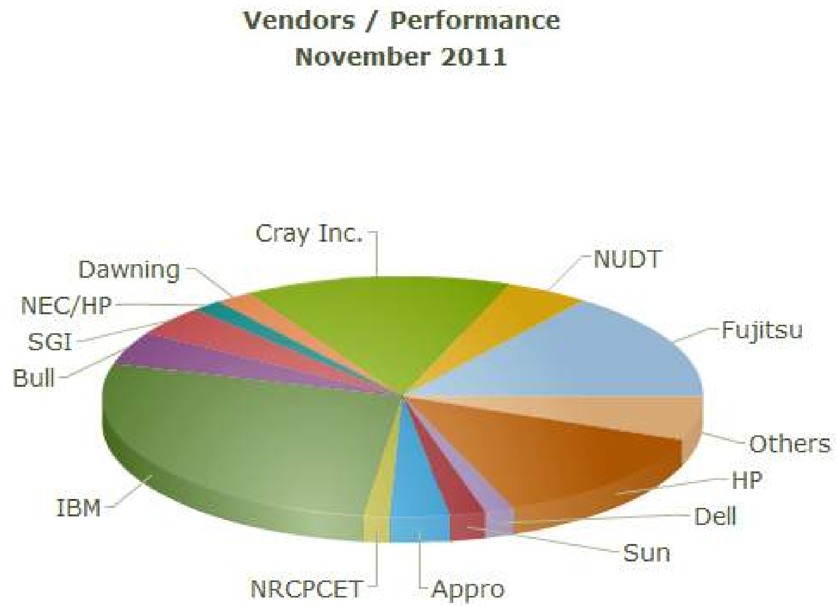
Top 1 : K Computer in Japan

Migration of Operating-System Concepts and Features

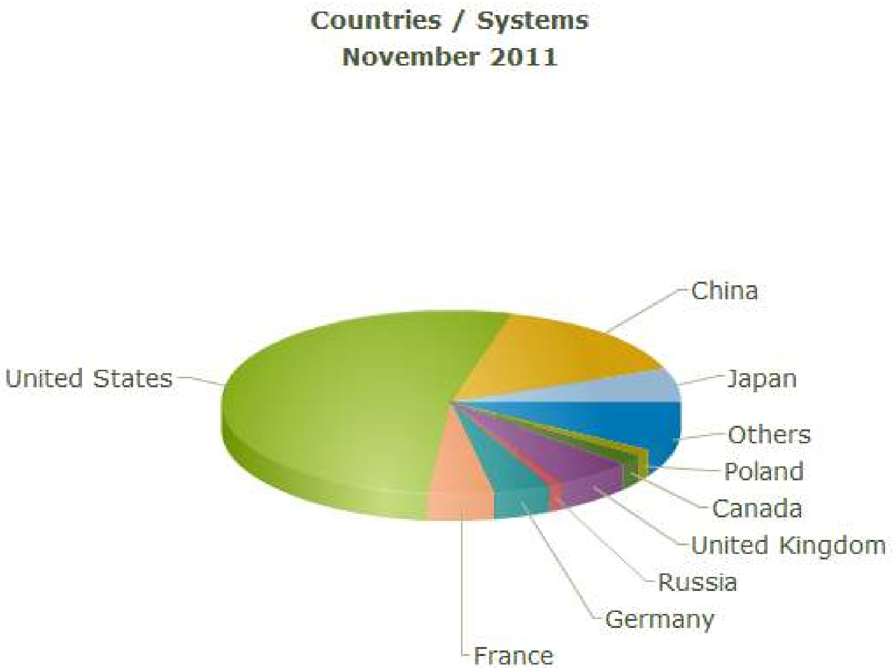


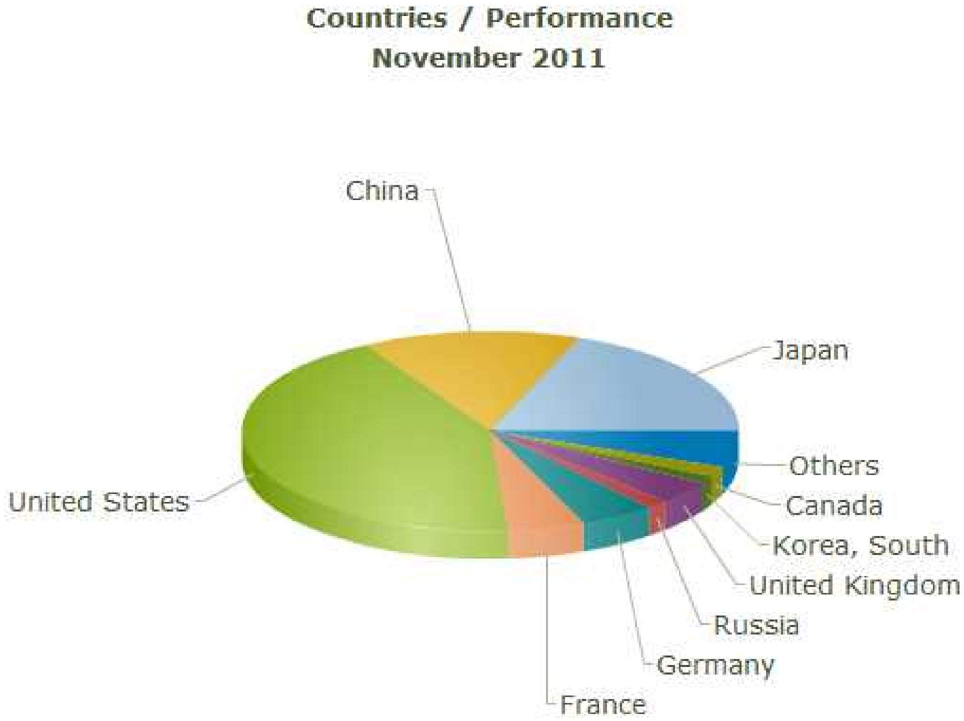
Systems and Vendors in TOP 500



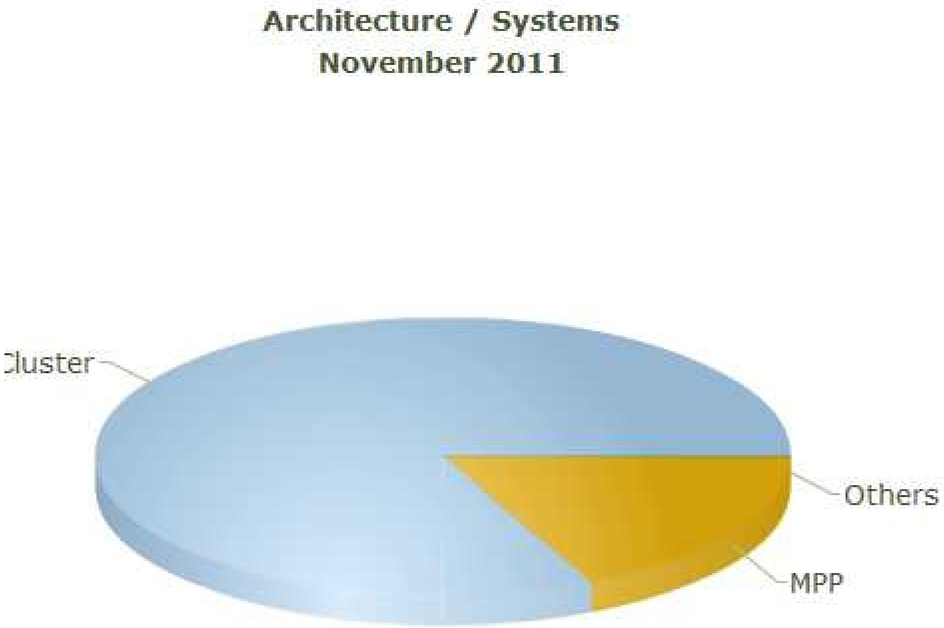


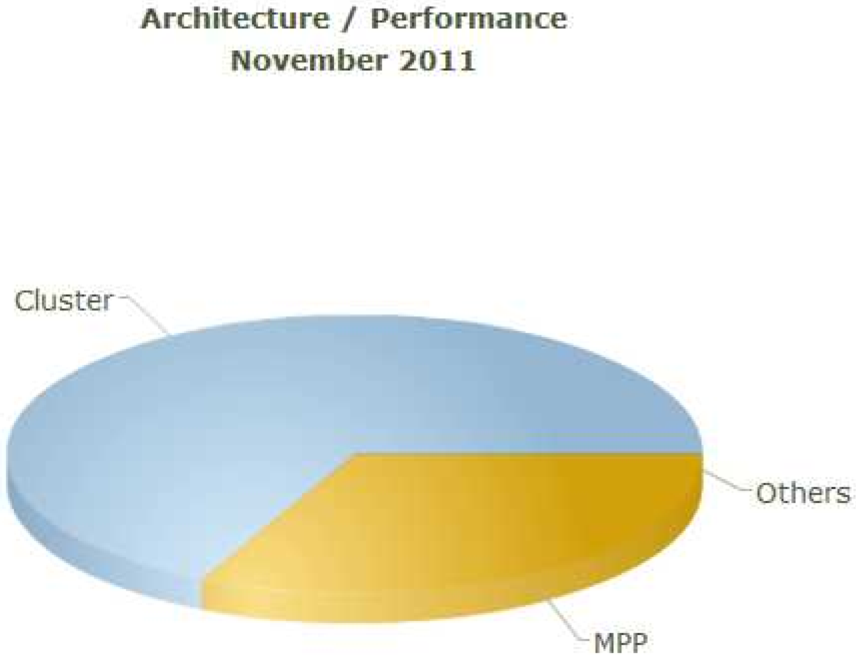
Systems and Vendors in TOP 500



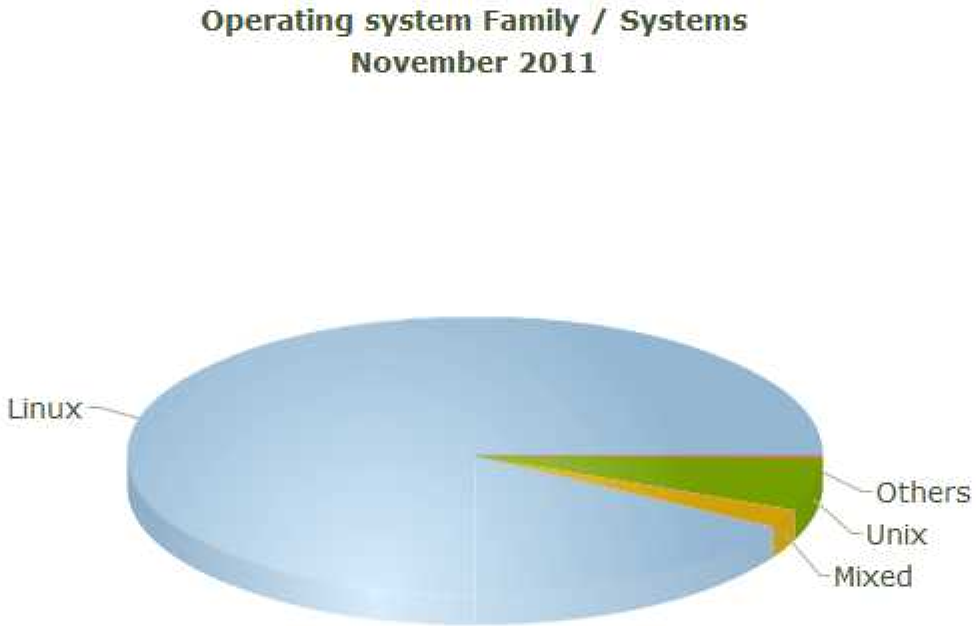


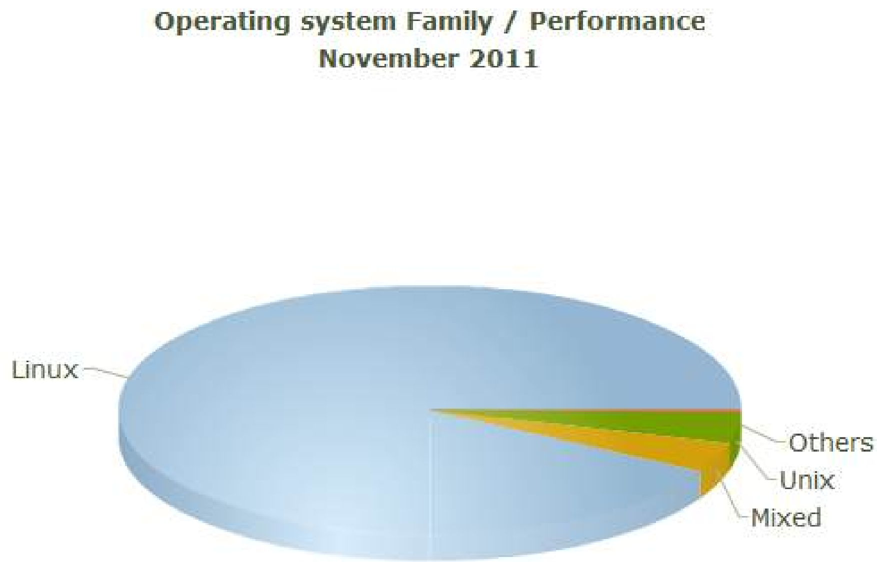
Architecture in TOP 500





Current OS in TOP 500

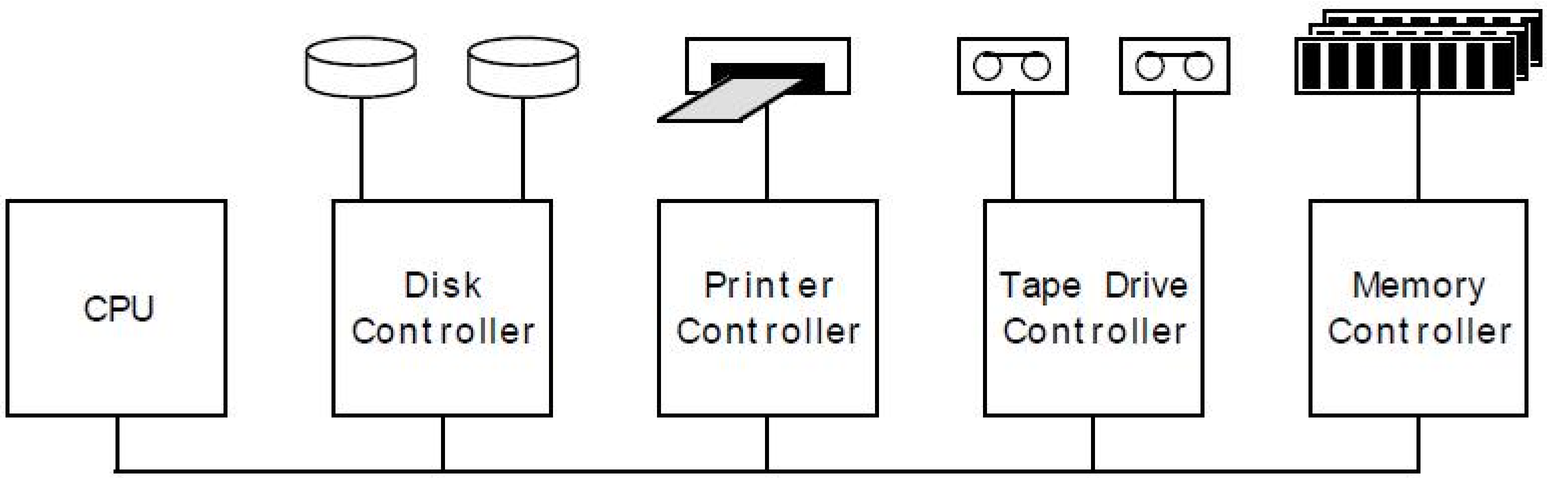


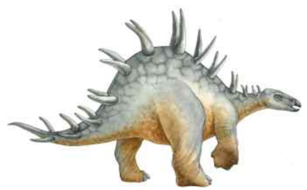


Computer System Architecture

Computer-System Architecture

- **Computer-system operation**
  - 하나 또는 그 이상의 **CPUs, device controllers**이 공유 메모리에 대한 접근을 제공한 공통 **bus**에 접속됨
  - **CPUs and devices**들이 **system bus**와 **memory cycle**을 위해 경쟁

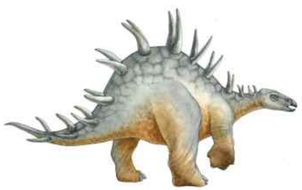




Computer-System Operation

- I/O devices와 CPU는 concurrently 실행할 수 있다
- 각 device controller 는 특정 device type을 갖는다
- Each device controller는 local buffer를 갖는다
  - CPU/IO간 동작
    - CPU는 data를 from/to main memory to/from local buffers 이동시킨다
    - I/O는 device들로부터 controller의 local buffer로 이동시킨다

Why not connect all I/O devices directly to CPU? Why not… memory…?



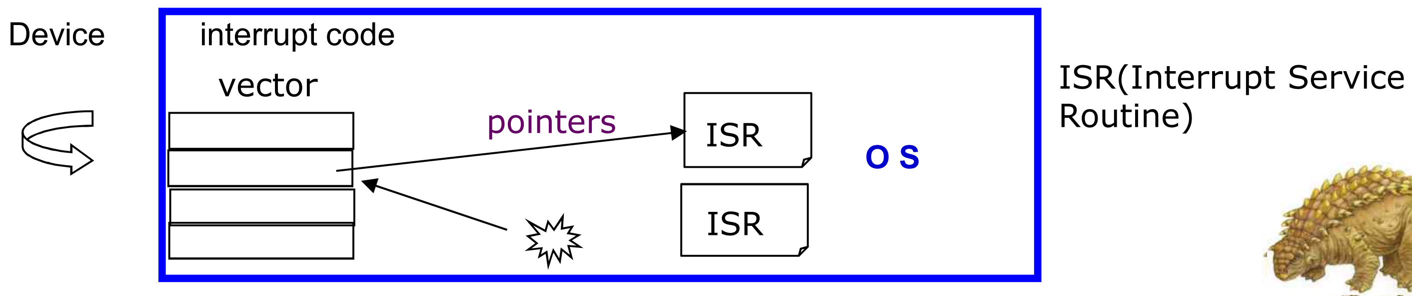
Common Functions of Interrupts

현대의 운영체제는 interrupt driven 방식을 사용한다!

- Interrupt는 일반적으로 interrupt vector를 통해 제어를 interrupt service routine으로 transfer한다. interrupt vector는 모든 service routine의 address를 포함하고 있다

- 'lost interrupt'를 방지하기 위해서 하나의 interrupt가 process 중일 때는 다른 interrupt는 사용할 수 없다
  - *error또는 사용자 request에 의해 발생하는 software-generated interrupt를 trap이라고 한다.*

- An operating system is *interrupt* driven.

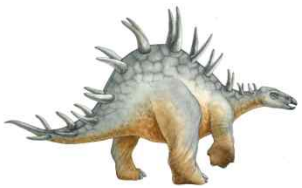


Interrupt Handling

- 1. The operating system preserves the state of the CPU by storing registers and the program counter.

- 2. 어떤 type의 intterupt가 발생했는지를 결정한다
  - *polling*
  - *vectored* interrupt system

- 3. 분리된 code segment를 통해 각 interrupt의 type별로 어떤 action이 수행되어야 하는지를 결정한다



I/O Structure

I/O Intterrupt의 수행순서

Device interrupts

↓

CPU stops current operation, switches to kernel mode, and saves current PC and other state on kernel stack

↓

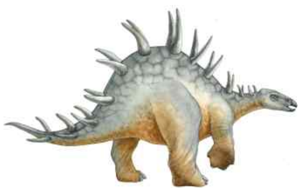
CPU fetches proper vector from vector table and branches to that address (to routine to handle interrupt)

↓

Interrupt routine examines device database and performs action required by interrupt

↓

Handler completes operation, restores saved (interrupt state) and returns to user mode (or calls scheduler to switch to another program)



Computer-System Operation

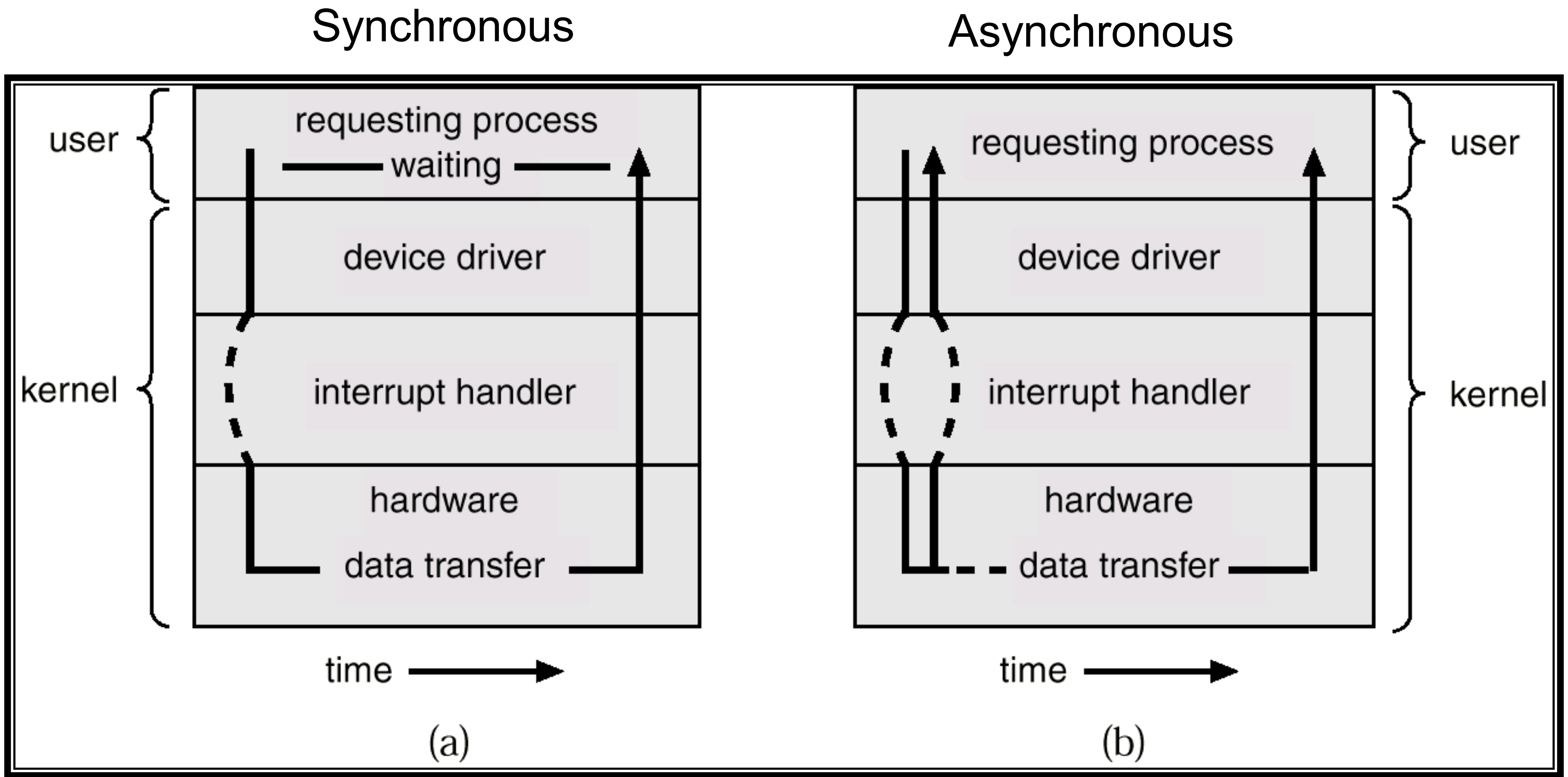
- Terminology
  - Synchronous I/O
    - CPU execution waits while I/O proceeds
  - Asynchronous I/O
    - I/O proceeds concurrently with CPU execution

I/O Structure

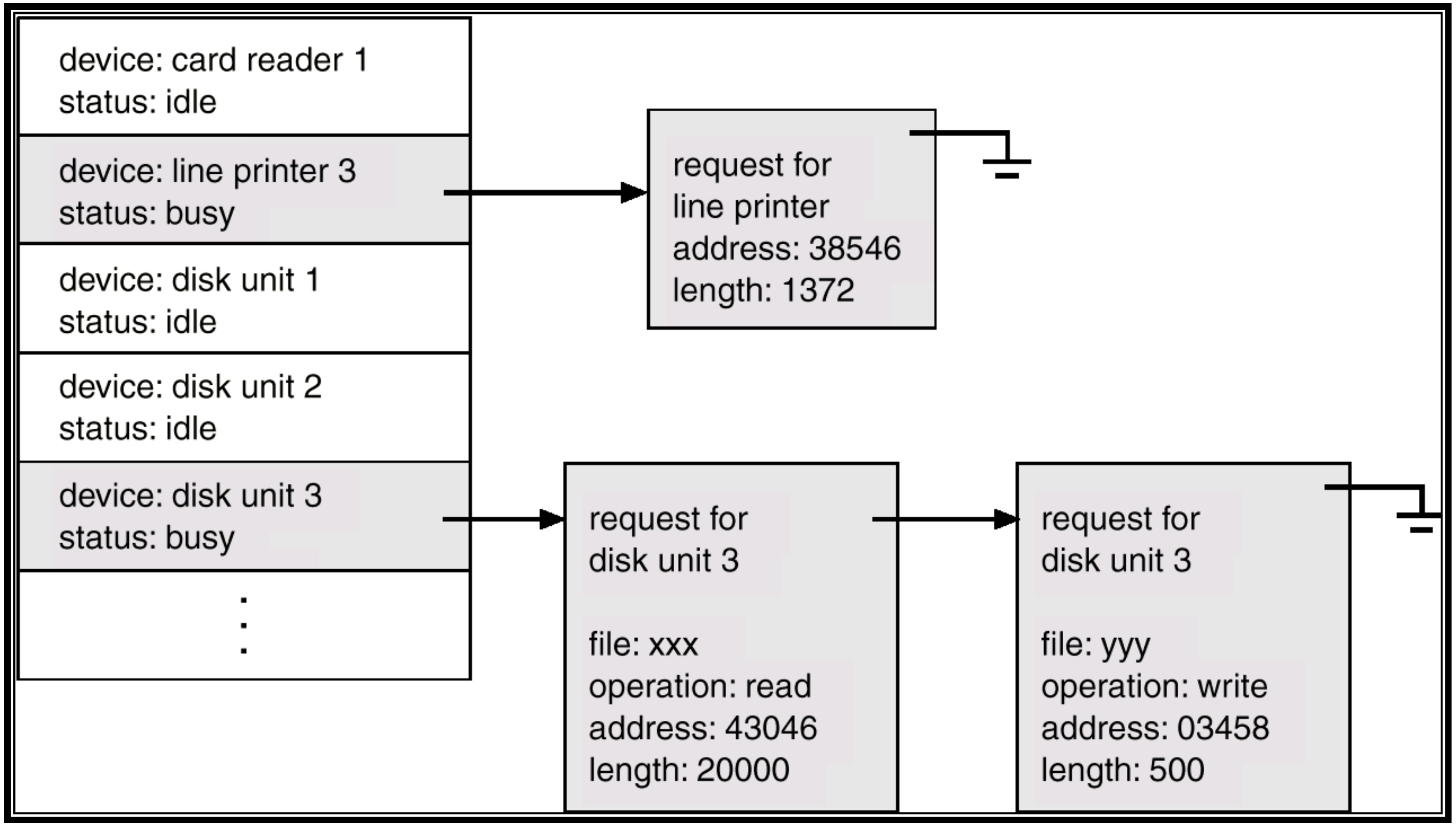
- After I/O starts, control returns to user program only upon I/O completion.(Synchronous)
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access).
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing.

- After I/O starts, control returns to user program without waiting for I/O completion.(Asynchronous)
  - *System call* – request to the operating system to allow user to wait for I/O completion.
  - *Device-status table* contains entry for each I/O device indicating its type, address, and state.
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt.

Two I/O Methods



Device-Status Table



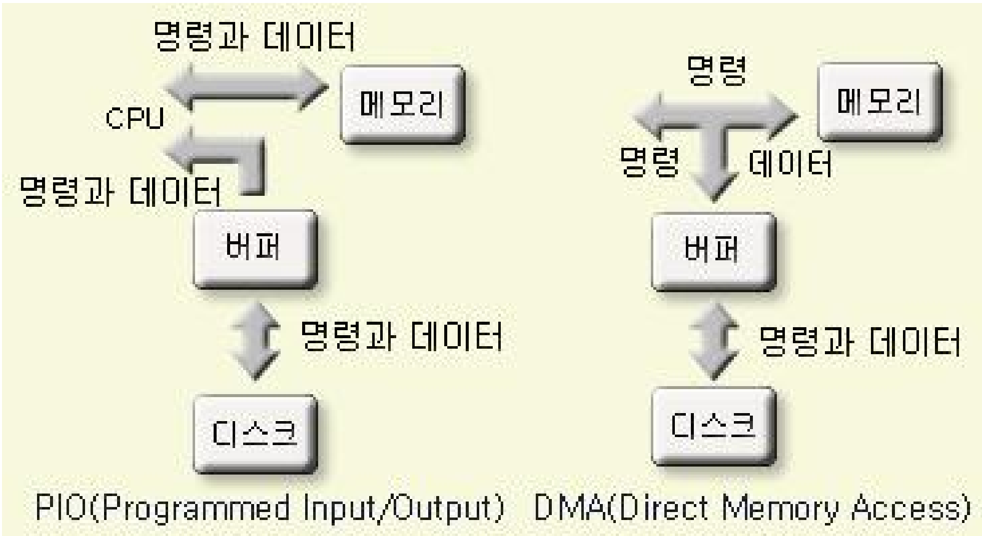
Direct Memory Access Structure

DMA = Direct Memory Access

- 디스크, 테이프, 또는 통신 네트워크와 같은 high-speed I/O device들이 memory speed에 근접하게 정보를 전송하는 경우 사용됨
  - 즉, 고속 I/O에 의해 빈번한 interrupt가 호출되는 것을 줄이고 고속으로 데이터를 전송하기 위해 사용
- DMA는 Device controller가 데이터들의 block들을 buffer 저장장소에서 main memory로 CPU intervention 없이 전송할 수 있게 함
- Only on interrupt is generated per block, rather than the one interrupt per byte.
  - cycle stealing에 의해 CPU로부터 memory cycle을 훔쳐서 실행하므로 DMA전송 동안 CPU실행이 늦어질수 있음

Direct Memory Access Structure

- DMA의 실행 구조

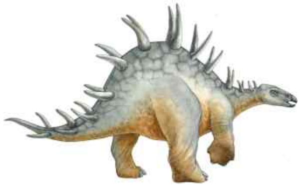


Storage Structure

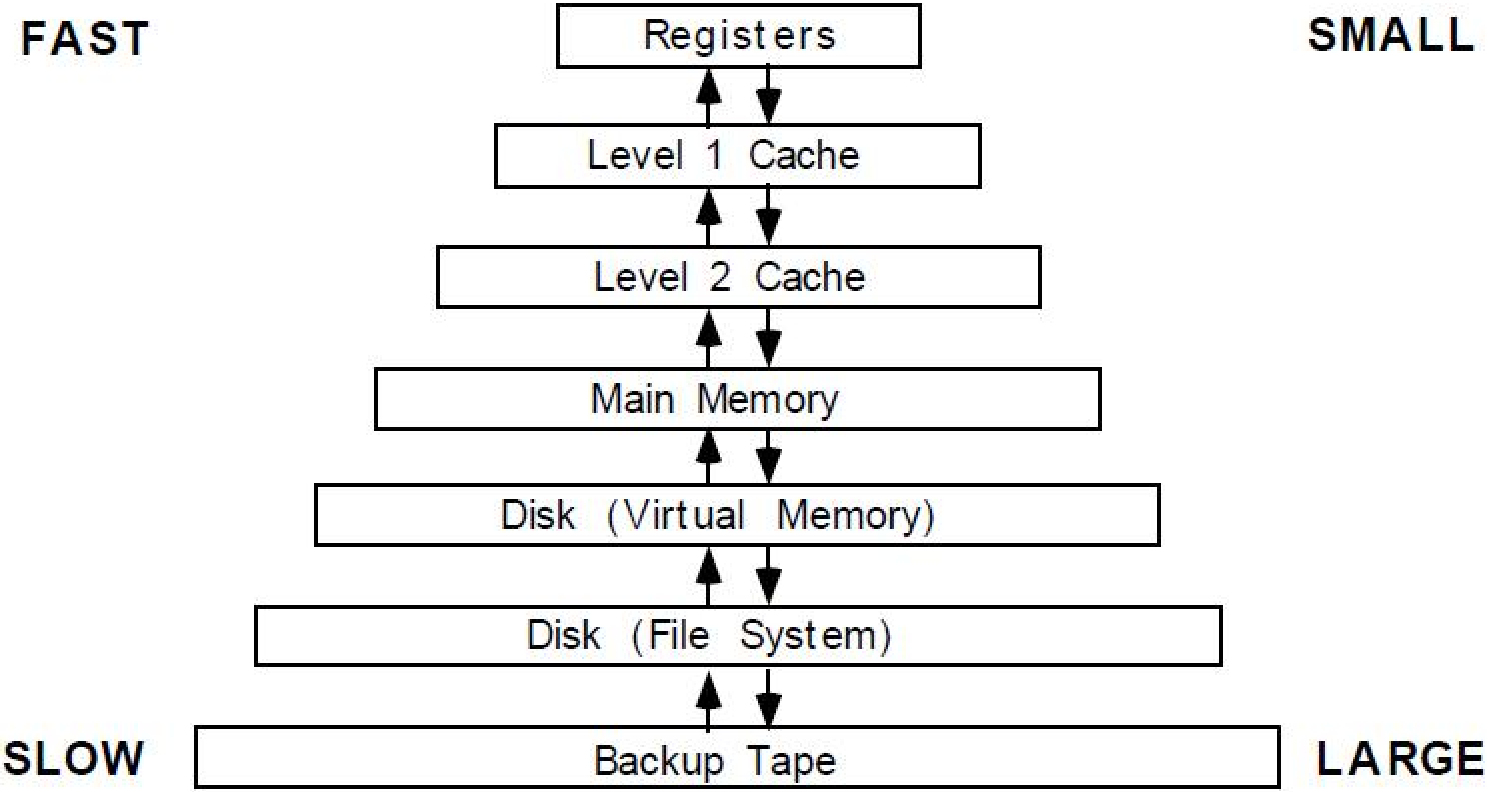
- Main memory – only large storage media that the CPU can access directly.
  - DRAM(Dynamic RAM) : data refresh 필요
  - SRAM(Static RAM) : 불필요
  - SDRAM(Synchronous DRAM) : data bus와 동기화(PC133)

- Secondary storage – extension of main memory that provides large nonvolatile storage capacity.

  - Magnetic disks – rigid metal or glass platters covered with magnetic recording material
    - Disk surface is logically divided into *tracks*, which are subdivided into *sectors*.
    - The *disk controller* determines the logical interaction between the device and the computer.



Storage-Device Hierarchy



현재 PC CPU의 Cache

제품명	Intel Core™2 Duo E6850	Intel Core™2 Duo E6550
생산 공정	65 nm	
코어 스테핑	G0	
물리적 실행코어 수	2	
소켓	FC-LGA775	
FSB	1333 MHz (333 MHz Quad Pumped)	
작동클럭	3.00 GHz	2.33 GHz
L1 캐시	L1 Data Cache : 32 KB x 2 L1 Instruction Cache : 32 KB x 2	
L2 캐시	4 MB (shared)	
CPU 배수	x9.0 (x6 ~ x9)	x7.0 (x6 ~ x7)
입력전압(VID, Max)	1.350 V	
열설계전력(TDP)	65W	
주요 사항	Intel 64 Architecture Enhanced Intel SpeedStep™ Technology (EIST) Extended Halt State (C1E) Execute Disable Bit (XD) Intel Trusted Execution Technology (TXT) Intel Virtualization Technology (VT) Intel Wide Dynamic Execution Intel Advanced Smart Cache Intel Smart Memory Access Intel Advanced Digital Media Boost 기술 등.	
판매가	268,000원	160,000원
	(다나와 최저가 기준, 2007년 8월 6일 현재)	

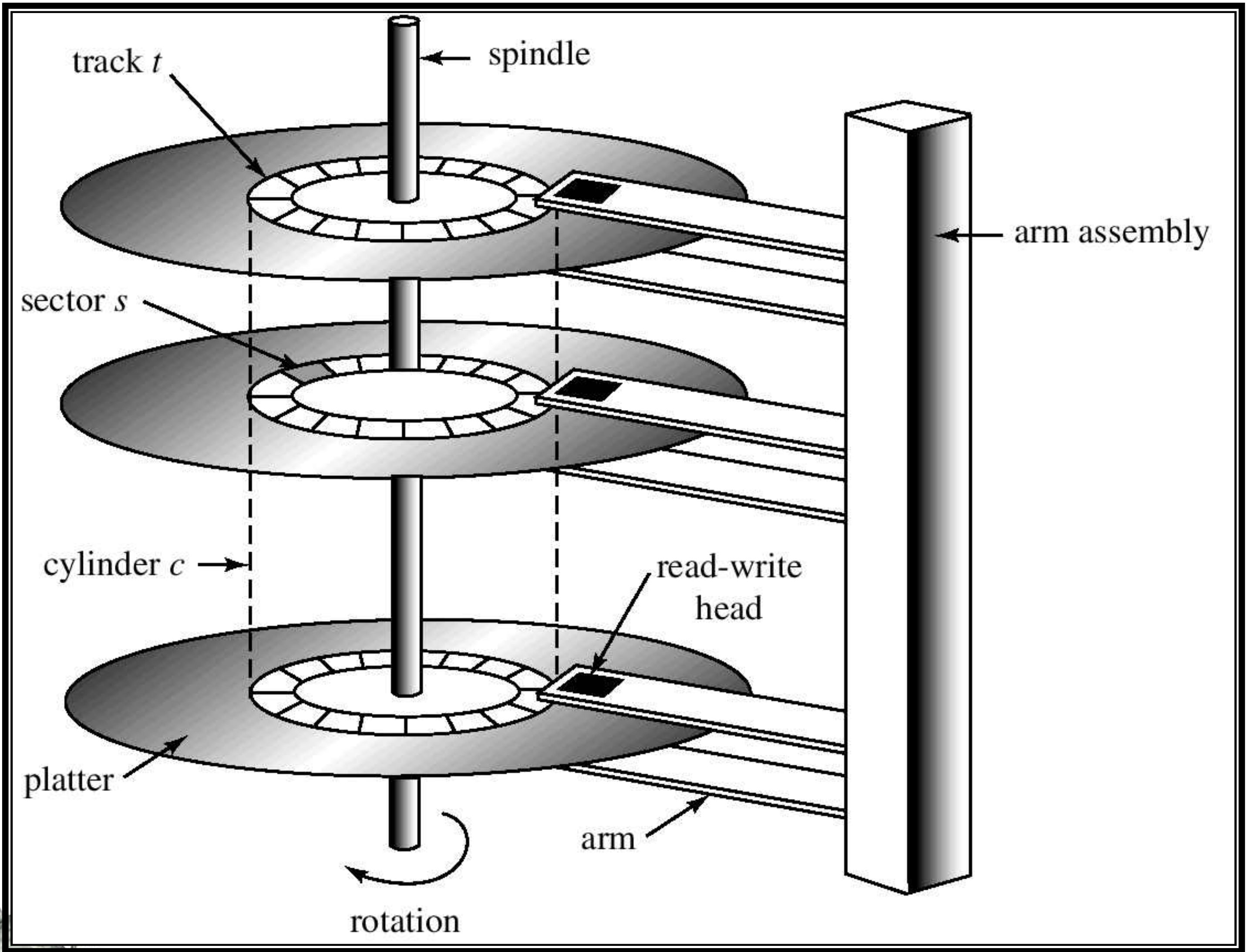
제품명	Intel Core™2 Quad Q6600
생산 공정	65 nm
물리적 실행코어 수	4
소켓	FC-LGA775
FSB	1066 MHz
작동 클럭	2.40 GHz
L1 캐시	L1 Data Cache : 32 KB x 4 L1 Instruction Cache : 32 KB x 4
L2 캐시	4 MB (shared) x 2
CPU 배수	x9.0 (x6 ~ x9)
입력전압(VID, Max)	1.350 V
주요 사항	EM64T Enhanced Intel SpeedStep™ Technology (EIST) Extended Halt State (C1E) Execute Disable Bit (XD) Intel Virtualization Technology (VT) Intel Wide Dynamic Execution Intel Advanced Smart Cache Intel Smart Memory Access Intel Advanced Digital Media Boost 기술 등.
판매가	990,000원 → 현재 306,000원 (다나와 최저가 기준, 2007년 1월 25일 현재)

작년 10월, 세계 최초로 4개의 물리적 실행코어를 가진 켄츠필드(kentsfield) 코어 기반의 CPU '코어2 익스트림 QX6700(Core2 Extreme QX6700)'을 발표했던 인텔이 예고했던 바대로 후속제품인 '코어2 쿼드 Q6600(Core2 Quad Q6600)'을 최근 선보였다.

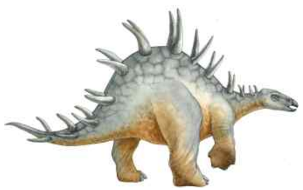
Storage Structure

현재의
PC 메모리

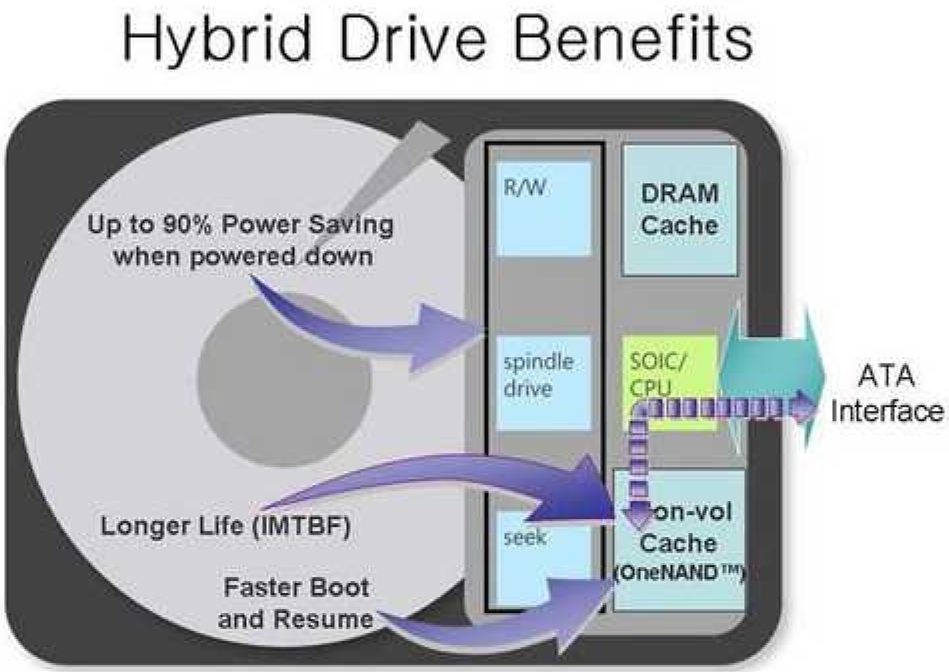
Moving-Head Disk Mechanism



Moving-Head Disk Mechanism



Hybrid HDD


Hybrid Drive Benefits
Up to 90% Power Saving when powered down
Longer Life (IMTBF)
Faster Boot and Resume
R/W
spindle drive
seek
DRAM Cache
SOIC/ CPU
on-vol Cache
(OneNAND™)
ATA Interface

Storage Structure:SSD

초고속 전송속도
충격에도 끄덕없는 단단한 내구성
1W의 소비전력

제품명 : 2,5" 64GB SSD SATA-2 SLC
제조사 : 삼성전자
규격 : 2.5" Type(100.2 X 70.0 X 9.5mm)
인터페이스 : S-ATA2 (SAS, S-ATA1 호환)
동작속도 : 읽기속도(평균)-100MB/s
쓰기속도(평균)-80MB/s
칩구성 : SLC
중량 : 73g(MAX)
MTBF : 2,000,000 시간
작동온도 : 0~70도
소비전력 : 1W
Operating Vibration : 20G(10~2000Hz)
Shock resistance : 1,500G for 0.5ms
Acoustic Noise : None

Storage Hierarchy

- Storage systems organized in hierarchy.
  - Speed
  - Cost
  - Volatility
- *Caching* – copying information into faster storage system; main memory can be viewed as a last *cache* for secondary storage.