1장: 소개

1장: 소개

- 운영 체제가 수행하는 작업컴퓨터 시스템
- 구성컴퓨터 시스템 아키텍처운영 체제 운
- 영리소스 관리보안 및 보호가상화분산 시
- 스템커널 데이터 구조컴퓨팅 환경무료/리
- 브레 및 오픈 소스 운영 체제

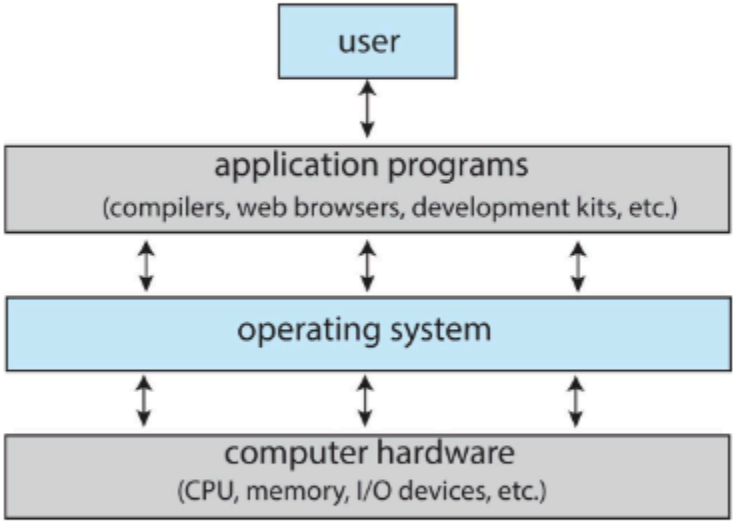
목표

- 컴퓨터 시스템의 일반적인 구성과 인터럽트의 역할을 설명합니다.최신 다중 프로세서 컴퓨터 시스템의 구성 요소를 설명합니다.사용자 모드에서 커널 모드로의 전환을 설명합니다.다양한 컴퓨팅 환경에서 운영 체제가 사용되는 방식에 대해 논의합니다.무료 및 오픈 소스 운영 체제의 예를 제공합니다.

컴퓨터 시스템 구조

- 컴퓨터 시스템은 네 가지 구성 요소로 나눌 수 있습니다.
  - 하드웨어 – 기본 컴퓨팅 리소스를 제공합니다.
    - CPU, 메모리, I/O 장치운
  - 영 체제
    - 다양한 응용 프로그램 간의 하드웨어 사용을 제어하고 조정하며
  - 사용자응용 프로그램 – 사용자의 컴퓨팅 문제를 해결하기 위해 시스템 리소스가 사용되는 방식을 정의합니다.
    - 워드 프로세서, 컴파일러, 웹 브라우저, 데이터베이스 시스템, 비디오
  - 게임사용자
    - 사람, 기계, 기타 컴퓨터

컴퓨터 구성 요소의 추상적 관점



운영 체제의 기능

- 관점에 따라 다름사용자는 편리함, 사용 편의성 및 우
- 수한 성능을 원합니다.
- 리소스 활용은 신경 쓰지 마십시오.그러나 메인프레임이나 미니컴퓨터와 같은 공유 컴퓨터는 모든 사용자를 만족시켜야 합니다.
- 운영 체제는 HW를 효율적으로 사용하고 사용자 프로그램의 실행을 관리하는 리소스 할당 및 제어 프로그램이며, 워크 스테이션과 같은 전용 시스템의 사용자는 전용 리소스를 가지고 있지만 서버의 공유 리소스를 자주 사용합니다.스마트 폰 및 테이블과 같은 모바일 장치는 리소스가 부족하고 유용성 및 배터리 수명이 최적화되어 있습니다.
- 터치 스크린, 음성 인식과 같은 모바일 사용자 인터페이스일부 컴퓨터에는 장치 및 자동차에 내장된 컴퓨터와 같은 사용자 인터페이스가 거의 또는 전혀 없습니다
  - 주로 사용자 개입 없이 실행

운영 체제 정의

- 용어 OS는 많은 역할을 다룹니다.
  - OS의 무수한 설계와 사용으로 인해선박, 우주선, 게임기, TV 및 산업
  - 제어 시스템을 통한 토스터에 존재 군용 고정 사용 컴퓨터가 보다 범용화되고 자원 관리 및 프로그램 제어가 필요했을 때 탄생
  -

운영 체제 정의(계속)

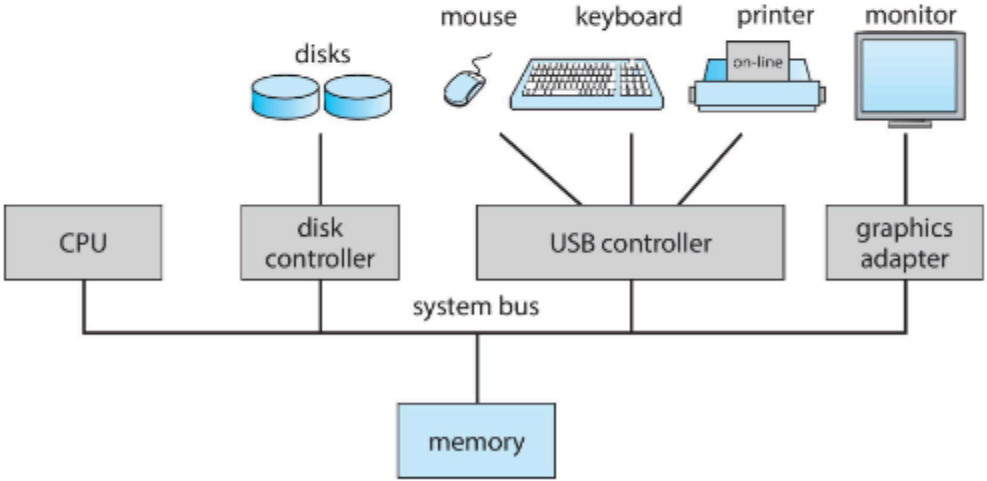
- 보편적으로 받아들여지는 정의는 없습니다."운영 체제를 주문할 때 공급업체가 배송하는 모든 것"은 좋은 근사치입니다

- 그러나 격렬하게 다릅니다."컴퓨터에서 항상 실행되는 하나의 프로그램"은 커널, 운영 체제의 일부입니다.다른 모든 것은 다음 중 하나입니다.

- 시스템 프로그램 (운영 체제와 함께 제공되지만 커널의 일부는 아님), ORAN 응용 프로그램, 운영 체제와 관련되지 않은 모든 프로그램 오늘날의 범용 및 모바일 컴퓨팅용 OS에는 데이터베이스, 멀티미디어, 그래픽과 같은 응용 프로그램 개발자에게 추가 서비스를 제공하는 소프트웨어 프레임 워크 세트 인 미들웨어도 포함됩니다

컴퓨터 시스템 구성

- 컴퓨터 시스템 작동
  - 하나 이상의 CPU, 장치 컨트롤러는 공유 메모리에 대한 액세스를 제공하는 공통 버스를 통해 연결메모리 주기를 놓고 경쟁하는 CPU 및 장치의 동
  - 시 실행



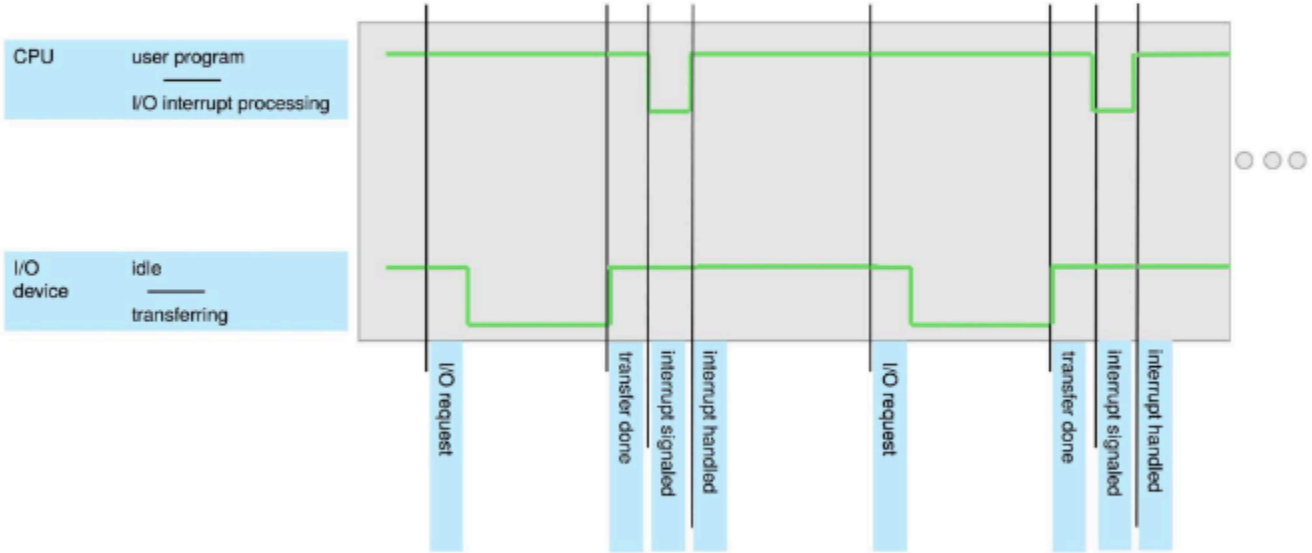
컴퓨터 시스템 운영

- I/O 장치와 CPU는 동시에 실행할 수 있습니다.각 장치 컨트롤러는 특정 장치 유형을 담당합니다.각 장치 컨트롤러에는 로컬 버퍼가 있습니다.각 장치 컨트롤러 유형에는 관리하는 운영 체제 장치 드라이버가 있습니다.CPU는 메인 메모리에서 로컬 버퍼로/로 데이터를 이동합니다.I/O는 장치에서 컨트롤러의 로컬 버퍼로I/O는 장치에서 컨트롤러의 로컬 버퍼로입니다.장치 컨트롤러는 인터럽트를 유발하여 CPU가 작업을 완료했음을 알립니다.

인터럽트의 공통 기능

- 인터럽트는 일반적으로 모든 서비스 루틴의 주소를 포함하는 인터럽트 벡터를 통해 인터럽트 서비스 루틴으로 제어를 전송합니다.인터럽트 아키텍
- 처는 중단된 명령어의 주소를 저장해야 합니다트랩 또는 예외는 오류 또는
- 사용자 요청으로 인해 발생하는 소프트웨어 생성 인터럽트입니다.운영 체제는 인터럽트 구동됩니다
-

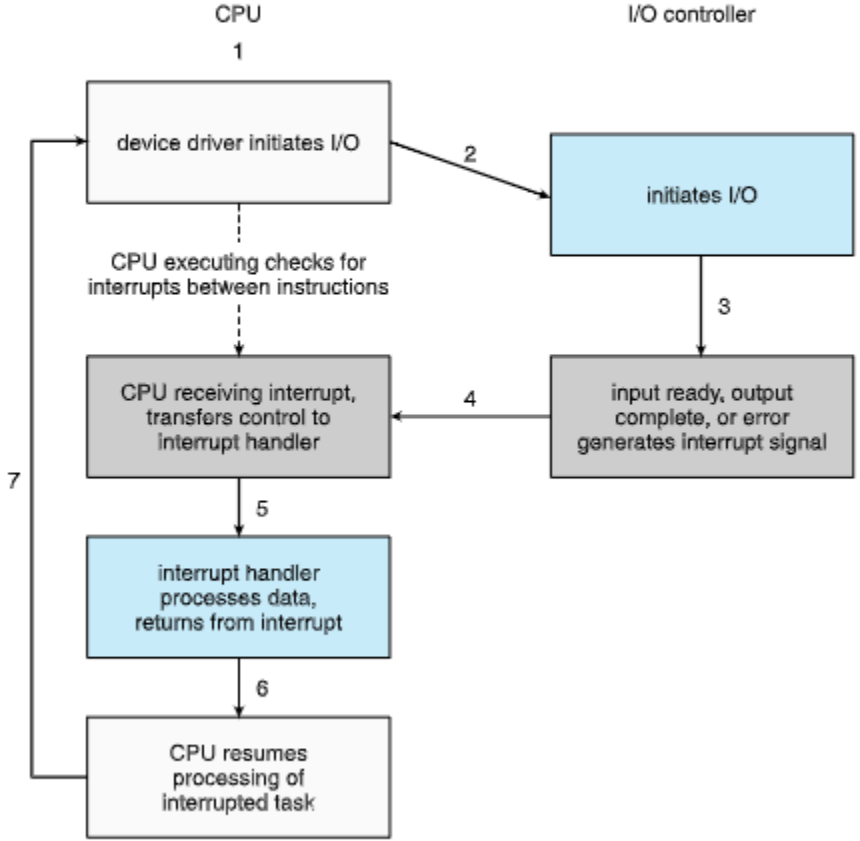
타임라인 인터럽트


CPU
user program
I/O interrupt processing
I/O device
idle
transferring
I/O request
transfer done
interrupt signaled
interrupt handled
I/O request
transfer done
interrupt signaled
interrupt handled

인터럽트 처리

- 운영 체제는 레지스터를 저장하여 CPU의 상태를 보존하고프로그램 카운터어떤 유형의 인터럽트가 발생했는지 확인합니다.
- **폴링벡터화된 인터럽트 시스템별도의 코드 세그먼트는 각 인터럽트 유형에 대해 수행해야 하는 작업을 결정합니다.**

인터럽트 드라이브 I/O 사이클



I/O 구조

- I/O가 시작된 후에는 I/O가 완료된 후에만 사용자 프로그램으로 제어가 반환됩니다

  대기 명령은 다음 인터럽트까지 CPU를 유휴 상태로 만듭니다.대기 루프(메모리 액세스 경합)한 번에 최대 하나의 I/O 요청이 미결되고 동시 I/O 처리가 없습니다.I/O가 시작된 후 I/O 완료를 기다리지 않고 사용자 프로그램으로 제어가 반환됩니다.

-
  - **시스템 호출 – 사용자가 I/O 완료를 기다릴 수 있도록 OS에 대한 요청**
  - **장치 상태 테이블에는 장치 상태를 확인하고 인터럽트를 포함하도록 테이블 항목을 수정하기 위해 I/O 장치 테이블에 대한 유형, 주소 및 상태를 나타내는 각 I/O 장치에 대한 항목이 포함됩니다.**

저장 구조

- 메인 메모리 – CPU가 직접 액세스할 수 있는 대용량 저장 매체만 있습니다.
  - **임의 액세스일반**
  - **적으로 휘발성**
  - 일반적으로 Dynamic Random-access 형태의 임의 액세스 메모리 **메모리(DRAM)**
- 보조 스토리지 – 큰 비휘발성 스토리지 용량을 제공하는 메인 메모리 확장
- 하드 디스크 드라이브(HDD) – 자기 기록 재료로 덮인 단단한 금속 또는 유리 플래터
  - 디스크 표면은 논리적으로 트랙으로 나뉘며, 트랙은 섹터로 세분화됩니다.디스크
  - 컨트롤러는 장치와 컴퓨터 간의 논리적 상호 작용을 결정합니다
- **비휘발성 메모리(NVM) 장치 – 하드 디스크보다 빠르고, 비휘발성**
  - 다양한 기술용량과 성능이 향상됨에 따라 대중화가 높아지고 가격이
  - 하락합니다.

스토리지 정의 및 표기법 검토

컴퓨터 스토리지의 기본 단위는 비트입니다. 비트는 0과 1의 두 값 중 하나를 포함할 수 있습니다. 컴퓨터의 다른 모든 저장소는 비트 모음을 기반으로 합니다. 비트가 충분하다면 컴퓨터가 숫자, 문자, 이미지, 영화, 소리, 문서, 프로그램 등 얼마나 많은 것을 나타낼 수 있는지 놀랍습니다. 바이트는 8비트이며 대부분의 컴퓨터에서 가장 작고 편리한 저장 덩어리입니다. 예를 들어, 대부분의 컴퓨터에는 비트를 이동하라는 명령이 없지만 바이트를 이동하라는 명령이 있습니다. 덜 일반적인 용어는 주어진 컴퓨터 아키텍처의 기본 데이터 단위인 단어입니다. 단어는 하나 이상의 바이트로 구성됩니다. 예를 들어 64비트 레지스터와 64비트 메모리 주소 지정이 있는 컴퓨터에는 일반적으로 64비트(8바이트) 단어가 있습니다. 컴퓨터는 한 번에 바이트가 아닌 기본 단어 크기로 많은 작업을 실행합니다. 대부분의 컴퓨터 처리량과 함께 컴퓨터 스토리지는 일반적으로 바이트 및 바이트 컬렉션으로 측정되고 조작됩니다. 킬로바이트(KB)는 1,024바이트메가바이트(MB)는 1,0242바이트기가바이트(GB)는 1,0243바이트테라바이트(TB)는 1,0244바이트페타바이트(PB)는 1,0245바이트입니다

컴퓨터 제조업체들은 종종 이 수치를 반올림하여 1메가바이트는 100만 바이트, 기가바이트는 10억 바이트라고 말합니다. 네트워킹 측정은 이 일반 규칙의 예외입니다. 그것들은 비트로 주어집니다(네트워크는 한 번에 조금씩 데이터를 이동하기 때문에).

스토리지 계층 구조

- 계층별로 구성된 스토리지 시스템

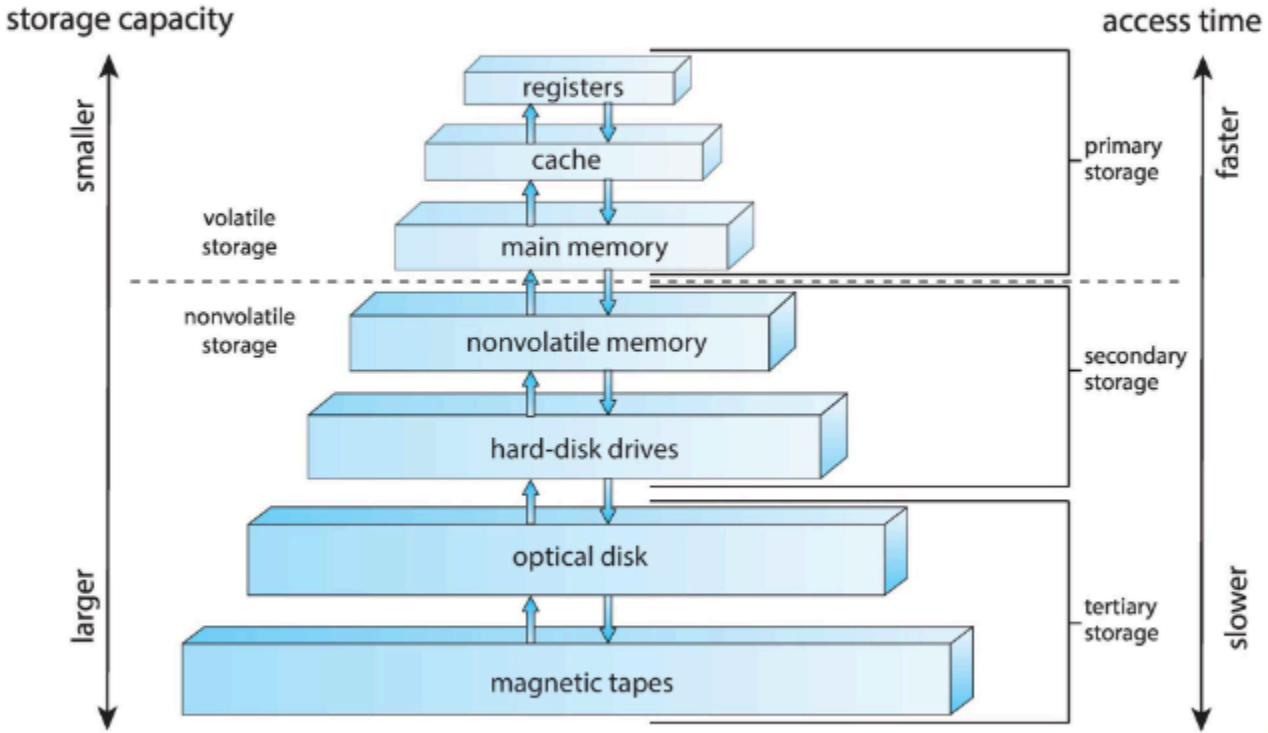
  SpeedCostVolatilityCaching – 더 빠른 스토리지 시스템으로 정보 복사; 메인 메모리는 보조 스토리지의 캐시로 볼 수 있습니다.I/O를 관리하기 위한 각 장치 컨트롤러의 장치 드라이버

-

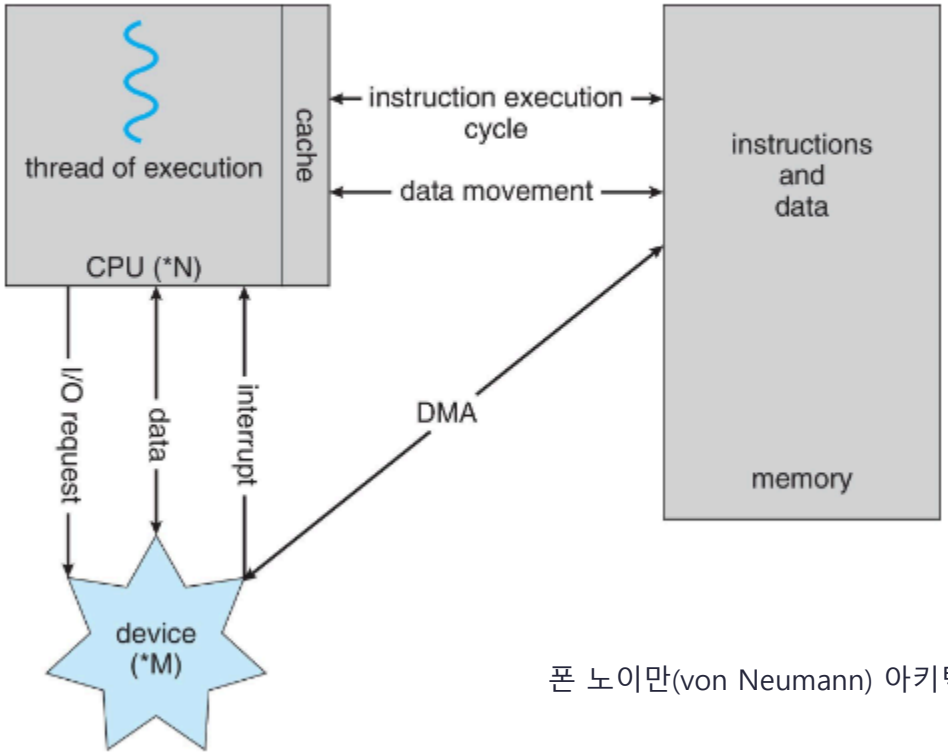
-

  - 컨트롤러와 커널 간의 균일한 인터페이스를 제공합니다.

저장 장치 계층 구조


storage capacity
access time
smaller
larger
faster
slower
registers
cache
main memory
volatile storage
nonvolatile memory
nonvolatile storage
hard-disk drives
optical disk
magnetic tapes
primary storage
secondary storage
tertiary storage

최신 컴퓨터의 작동 방식



폰 노이만(von Neumann) 아키텍처

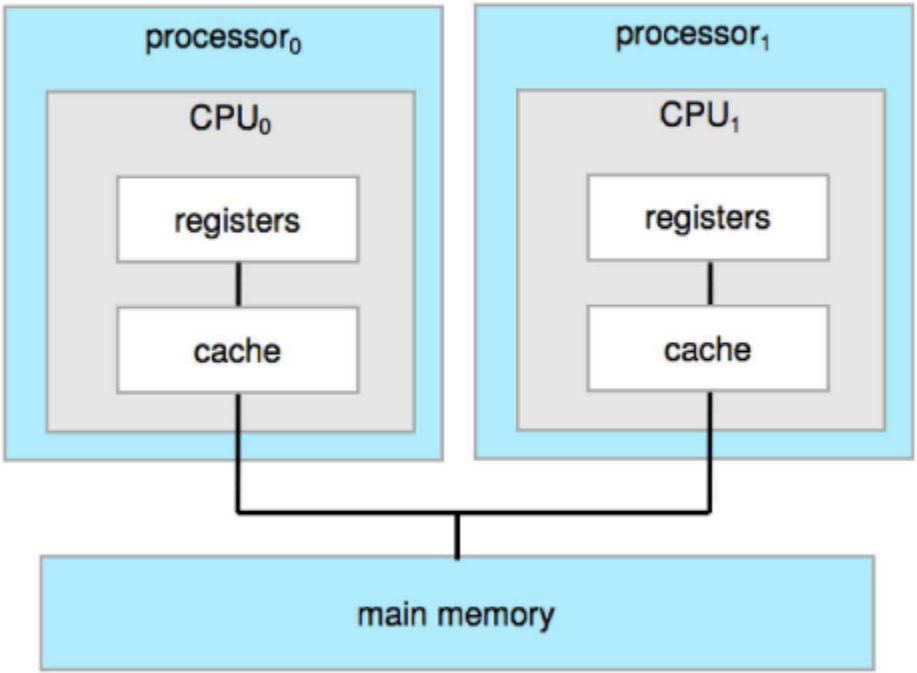
Direct Memory Access 구조

- 메모리 속도에 가까운 속도로 정보를 전송할 수 있는 고속 I/O 장치에 사용됨장치 컨트롤러는 CPU 개입 없이 버퍼 저장소에서 메인 메모리로 직접 데
- 이터 블록을 전송합니다.바이트당 하나의 인터럽트가 아닌 블록당 하나의 인터럽트만 생성됩니다.
-

컴퓨터 시스템 아키텍처

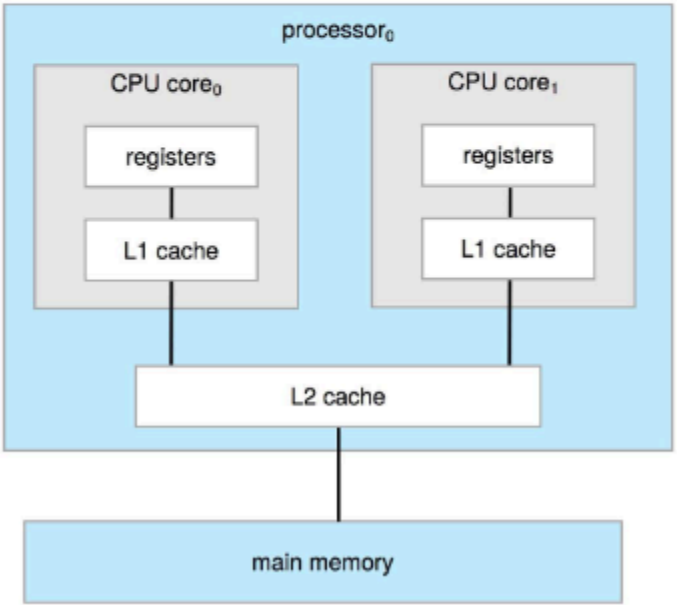
- 대부분의 시스템은 단일 범용 프로세서를 사용합니다
- 대부분의 시스템에는 특수 목적의 프로세서도 있습니다.다중 프로세서 시스템의 사용과 중요성이 증가하고 있습니다.병렬 시스템이라고도 하며 밀접하게 결합된 시스템이라고도 합니다.
  - 장점은 다음과 같습니다.
    1. **처리량 증가**
    2. **규모의 경제**
    3. **신뢰성 향상 – 정상적인 성능 저하 또는 내결함성두 가지**
  - **유형:**
    1. **비대칭 다중 처리 – 각 프로세서에는 특정 작업이 할당됩니다.**
    2. **대칭적 멀티프로세싱 – 각 프로세서가 모든 작업을 수행합니다.**

Symmetric Multiprocessing 아키텍처

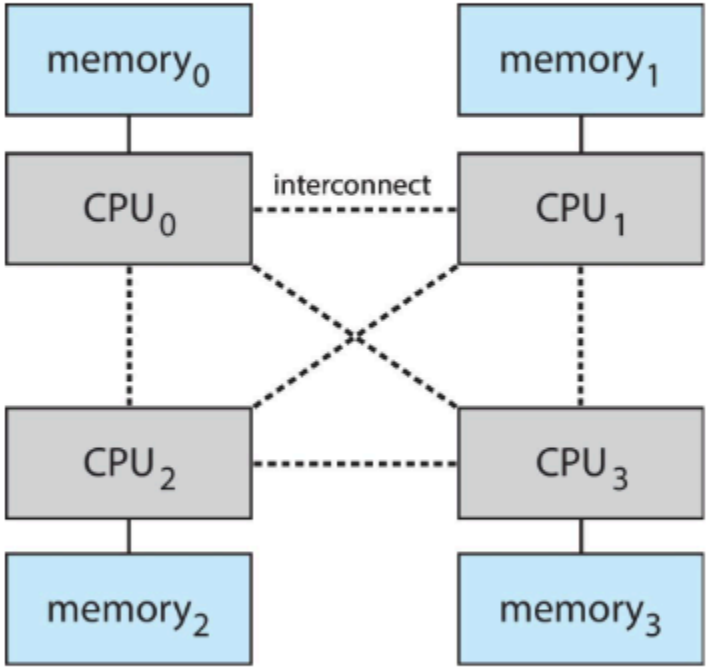


듀얼 코어 설계

- 멀티칩 및 멀티코어모든
- 칩을 포함하는 시스템
  - 여러 개의 개별 시스템을 포함하는 섀시



Non-Uniform Memory Access 시스템



클러스터링된 시스템

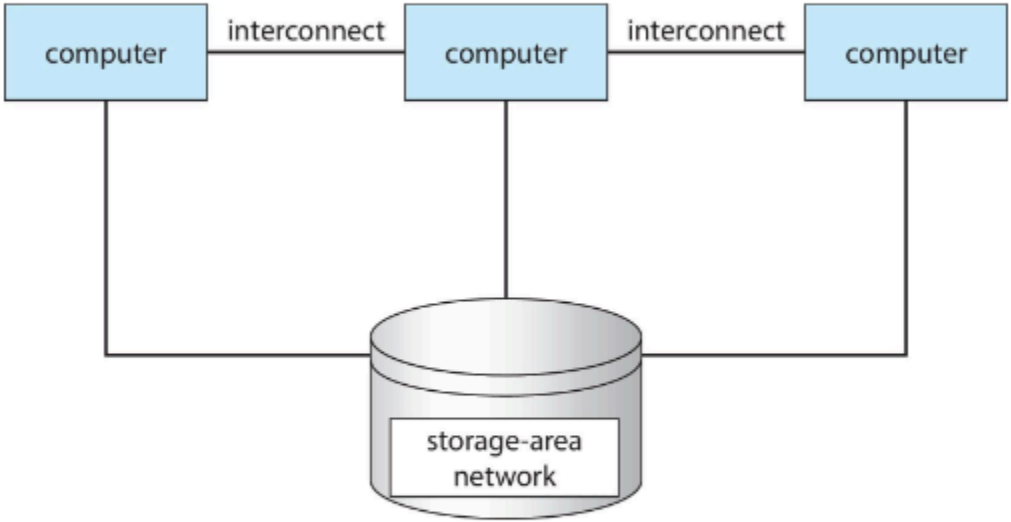
- 다중 프로세서 시스템과 비슷하지만 여러 시스템이 함께 작동합니다.
  - 일반적으로 SAN(Storage-Area Network)을 통해 스토리지를
  - 공유합니다.장애를 견디는 고가용성 서비스를 제공합니다.

    비대칭 클러스터링에는 하나의 시스템이 핫 스탠바이 모드에 있습니다. 대칭 클러스터링에는 여러 노드가 애플리케이션을 실행하고 모니터링합니다.

    서로일부 클러스터는 고성능 컴퓨팅(HPC)용입니다.

  - 응용 프로그램은 병렬화를 사용하도록 작성되어야 합니다.일부는
  - 작업 충돌을 피하기 위해 DLM(분산 잠금 관리자)을 가지고 있습니다.

클러스터링된 시스템


computer
interconnect
computer
interconnect
computer
storage-area
network

운영 체제 운영

- 부트 스트랩 프로그램 – 시스템 초기화, 커널로드를 위한 간단한
- 코드커널 로드시스템 데몬 시작(커널 외부에서 제공되는 서비스)
- 커널 인터럽트 구동(하드웨어 및 소프트웨어)
-
  - 장치 중 하나에 의한 하드웨어 인터럽트소프트웨어 인터럽트(예외 또는 트
  - 랩):

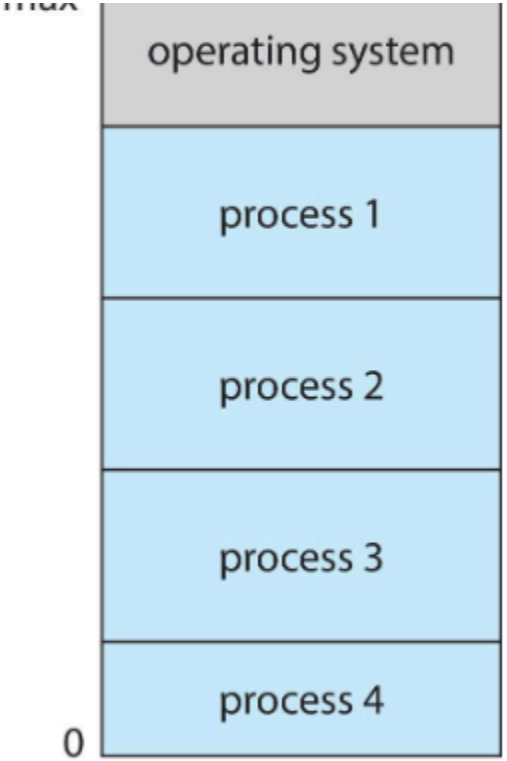
    □ 소프트웨어 오류(예: 0으로 나누기)□ 운영 체제 서비스 요청 – 시스템 호출□ 기타 프로세스 문제로는 무한 루프, 서로 수정하는 프로세스 또는 운영 체제가 있습니다.

멀티프로그래밍과 멀티태스킹

- **효율성을 위해 필요한 멀티프로그래밍(Batch 시스템)**
  - 단일 사용자는 CPU 및 I/O 장치를 항상 사용 상태로 유지할 수 없습니다.다중
  - 프로그래밍은 작업(코드 및 데이터)을 구성하여 CPU가 항상 하나를 실행할
  - 수 있도록 합니다.시스템의 전체 작업 중 하위 집합은 메모리에 보관됩니다.
  - 하나의 작업이 선택되고 작업 스케줄링을 통해 실행됩니다.
  - 대기해야 하는 경우(예: I/O의 경우) OS가 다른 작업으로 전환됩니다

- **시분할(멀티태스킹)은 CPU가 작업을 너무 자주 전환하여 사용자가 실행되는 동안 각 작업과 상호 작용할 수 있도록 하여 대화형 기능을 생성하는 논리적 확장입니다**
  컴퓨팅
  - **응답 시간은 1초 <이어야 합니다.각 사용자는 메모리에서 하나 이상**
  - **의 프로그램을 실행하고 있습니다 □프로세스여러 작업을 동시에 실**
  - **행할 준비가 된 경우 □ CPU 스케줄링프로세스가 메모리에 맞지 않는**
  - **경우 스왑을 통해 프로세스를 안팎으로 이동하여 실행합니다.가상 메**
  - **모리는 메모리에 완전히 있지 않은 프로세스를 실행할 수 있습니다.**

멀티프로그래밍된 시스템을 위한 메모리 레이아웃



듀얼 모드 및 다중 모드 작동

- **듀얼 모드 작동을 통해 OS는 자체 및 기타 시스템 구성 요소를 보호할 수 있습니다.**
  - **하드웨어에서 제공하는 사용자 모드 및 커널 모드모드 비트**
  -

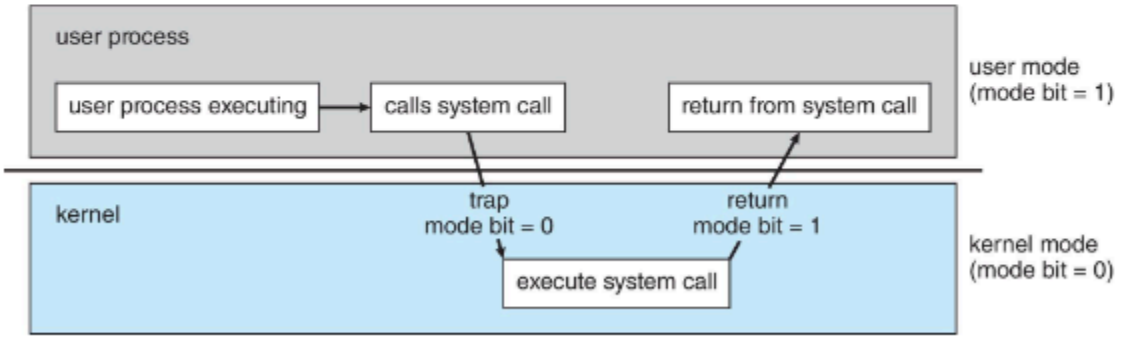
□ 시스템이 사용자 코드 또는 커널 코드를 실행 중일 때를 구별할 수 있는 기능 제공□ 권한 있는 것으로 지정된 일부 명령어는 커널 모드에서만 실행 가능□ 시스템 호출이 모드를 커널로 변경하고, 호출에서 반환하여 사용자로 재설정점점 더 많은 CPU가 다중 모드 작업을 지원함

-

  - 즉, 게스트 VM에 대한 VMM(가상 머신 관리자) 모드

사용자에서 커널 모드로 전환

- 무한 루프를 방지하기 위한 타이머 / 리소스 호깅 처리
  - 타이머는 일정 기간 후에 컴퓨터를 중단하도록 설정됩니다물리적 시
  - 계에 의해 감소하는 카운터를 유지하십시오운영 체제는 카운터를 설
  - 정합니다 (특권 명령어)카운터 0이 인터럽트를 생성 할 때할당된 시
  - 간을 초과하는 프로그램을 제어하거나 종료하기 위해 프로세스를 스
  - 케줄링하기 전에 설정합니다.



프로세스 관리

- 프로세스는 실행 중인 프로그램입니다. 시스템 내의 작업 단위입니다. Program은 수동적인 실체이고, process는 능동적인 실체입니다. 프로세스는
- 작업을 수행하기 위해 리소스가 필요합니다.

  CPU, 메모리, I/O, 파일초기화 데이터프로세스 종료는 재사용 가능한 리소스를 회수해야 합니다.단일 스레드 프로세스에는 실행할 다음 명령의 위치
- 를 지정하는 하나의 프로그램 카운터가 있습니다.
-

  프로세스는 완료될 때까지 한 번에 하나씩 명령을 순차적으로 실행합니다다중
- 스레드 프로세스에는 스레드당 하나의 프로그램 카운터가 있습니다.일반적으
- 로 시스템에는 하나 이상의 CPU에서 동시에 실행되는 많은 프로세스, 일부 사용자, 일부 운영 체제가 있습니다.
  - 프로세스/스레드 간에 CPU를 다중화하여 동시성

프로세스 관리 활동

운영 체제는 프로세스 관리와 관련하여 다음과 같은 활동을 담당합니다.

- 사용자 및 시스템 프로세스 모두 생성 및 삭제
- 프로세스 일시중단 및 재개프로세스 동기화를
- 위한 메커니즘 제공프로세스 통신을 위한 메커
- 니즘 제공교착 상태 처리를 위한 메커니즘 제공
-

메모리 관리

- 프로그램을 실행하려면 명령어의 전부(또는 일부)가 메모리에 있어야 합니다
- 프로그램에 필요한 데이터의 모든(또는 일부)이 메모리에 있어야 합니다메모
- 리 관리는 메모리에 무엇이 있는지, 그리고 언제 메모리에 있는지 결정합니다.

  CPU 사용률 및 사용자에 대한 컴퓨터 응답 최적화메모리
- 관리 활동
  - 메모리의 어떤 부분이 현재 사용되고 있는지, 누가 사용하고 있는지 추
  - 적메모리로 들어가거나 나갈 프로세스(또는 그 부분) 및 데이터 결정필
    요에 따라 메모리 공간을 할당하고 할당 해제합니다.
  -

파일 시스템 관리

- OS는 정보 스토리지에 대한 균일하고 논리적인 보기를 제공합니다.
  - 물리적 속성을 논리적 저장 장치 - 파일로 추상화각 매체는 장치(예: 디스
  - 크 드라이브, 테이프 드라이브)에 의해 제어됩니다.

    □ 다양한 속성에는 액세스 속도, 용량, 데이터 전송 속도, 액세스 방법(순차 또는 임의)이 포함됩니다.

- 파일 시스템 관리
  - 파일은 일반적으로 디렉토리로 구성됩니다.whatOS 활동에
  - 액세스할 수 있는 사람을 결정하기 위해 대부분의 시스템에
  - 서 액세스 제어는 다음과 같습니다.

    □ 파일 및 디렉토리 생성 및 삭제□ 파일 및 디렉토리를 조작하기 위한 프리미티브□ 보조 스토리지에 파일 매핑□ 안정적인(비휘발성) 스토리지 미디어에 파일 백업

대용량 스토리지 관리

- 일반적으로 주 메모리에 맞지 않는 데이터 또는 "장기간" 동안 보관해야 하는 데이터를 저장하는 데 사용되는 디스크적절한 관리가 가장 중요합니
- 다컴퓨터 작동의 전체 속도는 디스크 하위 시스템과 해당 알고리즘에 달
- 려 있습니다.OS 활동
-

  마운트 및 마운트 해제여유
  공간 관리스토리지 할당디
  스크 스케줄링파티셔닝보호
  일부 스토리지는 빠를 필요
  가 없습니다.

-
  - 3차 스토리지에는 광학 스토리지, 자기 테이프가 포함됩니
  - 다.여전히 OS 또는 애플리케이션에 의해 관리해야 합니다.

캐싱

- 컴퓨터의 여러 수준에서 수행되는 중요한 원칙(하드웨어, 운영 체제, 소프트웨어)사용 중인 정보는 느린 저장소에서 빠른 저장소로
- 임시로 복사더 빠른 저장소(캐시)를 먼저 확인하여 정보가 있는지
- 확인합니다.

  그렇다면 캐시에서 직접 정보가 사용됩니다 (빠름)
  그렇지 않은 경우 데이터가 캐시에 복사되어 사용됩
- 니다.캐시되는 스토리지보다 작습니다.
  - 캐시 관리 중요 설계 문제캐시 크기 및
  - 교체 정책

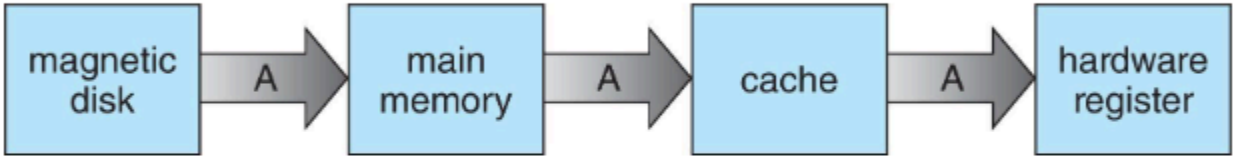
다양한 유형의 스토리지의 특징

Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

스토리지 계층 수준 간의 이동은 명시적이거나 암시적일 수 있습니다

디스크에서 레지스터로 데이터 "A"의 마이그레이션

- 멀티태스킹 환경은 스토리지 계층 구조에 저장된 위치에 관계없이 가장 최근의 값을 사용하도록 주의해야 합니다



- 다중 프로세서 환경은 모든 CPU가 캐시에서 가장 최신 값을 갖도록 하드웨어에 캐시 일관성을 제공해야 합니다.분산 환경 상황은 훨씬 더 복잡합니다
-
  - 여러 데이텀 사본이 존재할 수 있음
  - 19장에서 다루는 다양한 솔루션

I/O 하위 시스템

- OS의 한 가지 목적은 하드웨어 장치의 특성을 담당하는 사용자 I/O
- 하위 시스템에서 숨기는 것입니다.
  - 버퍼링(데이터가 전송되는 동안 임시로 데이터 저장), 캐싱(성능 향상을 위해 데이터의 일부를 더 빠른 저장소에 저장), 스풀링(한 작업의 출력과 다른 작업의 입력이 겹치는 것)을 포함한 I/O의 메모리 관리일반 장치-드라이버 인터페이스특정 하드웨어 장치용 드라이버
  -
  -

보호 및 보안

- **보호 – OSSecurity에 의해 정의된 리소스에 대한 프로세스 또는 사용자의 액세스를 제어하는 모든 메커니즘 – 내부 및 외부 공격에 대한 시스템 방어**
- 서비스 거부, 웜, 바이러스, 신원 도용, 서비스 도용을 포함한 광범위한 시스템일반적으로 먼저 사용자를 구별하여 누가 무엇을 할 수 있는지 결정합니다
  - 사용자 ID (사용자 ID, 보안 ID)에는 이름 및 관련 번호가 포함되며, 사용자 ID 당 하나씩 모든 파일과 연결되며, 해당 사용자의 프로세스로 액세스 제어그룹 식별자 (그룹 ID)를 결정하면 사용자 집합을 정의하고 제어 관리 할 수 있으며 각 프로세스와 연결할 수 있으며, filePrivilege 에스컬레이션을 통해 사용자는 더 많은 권한을 가진 유효 ID로 변경할 수 있습니다

가상화

- 운영 체제가 다른 OS 내에서 응용 프로그램을 실행할 수 있도록 합니다.
- 방대하고 성장하는 산업소스 CPU 유형이 대상 유형과 다를 때 사용되는 에뮬레이션 (예 : PowerPC에서 Intel x86)

- 일반적으로 가장 느린 방법컴퓨터 언어가 네이티브 코드로 컴파일되지 않은 경우 – 해석가상화 – OS가 CPU용으로 기본적으로 컴파일되고 게스트 OS도 기본적으로 컴파일됩니다.

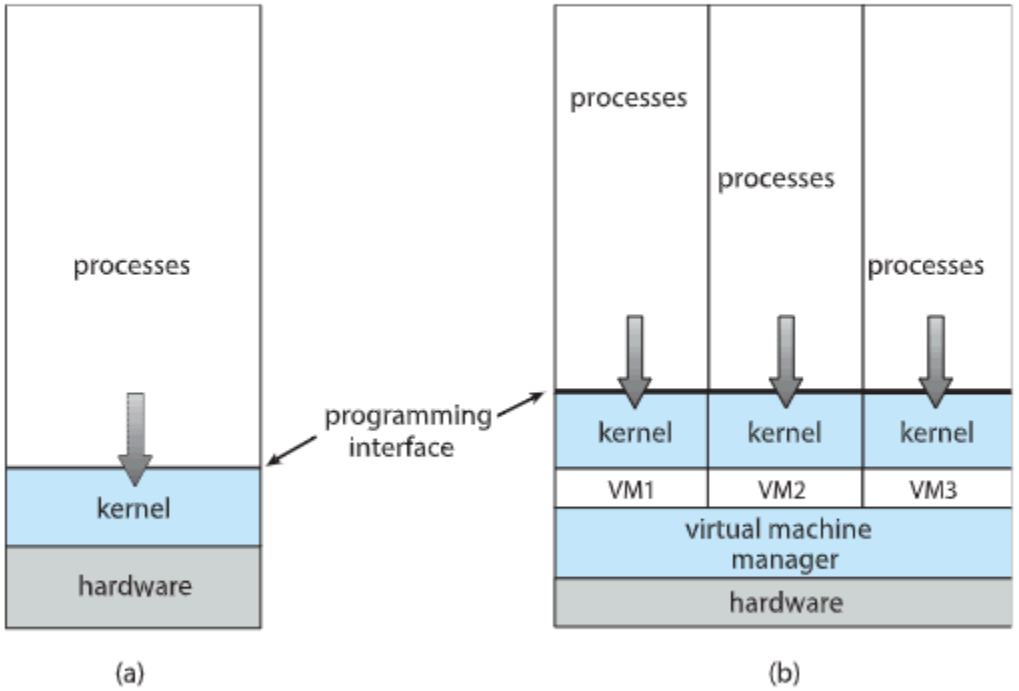
  - WinXP 게스트를 실행하는 VMware, 각 응용 프로그램을 실행하는 모든 기본 WinXP 호스트 OSVMM(가상 머신 관리자)이 가상화 서비스를 제공하는 것을 고려하십시오.

가상화(계속)

- 사용 사례에는 탐색 또는 호환성을 위해 여러 OS를 실행하는 랩톱 및 데스크톱이 포함됩니다

  Mac OS X 호스트, Windows를 게스트로 실행하는 Apple 노트북
  여러 시스템 없이 여러 OS용 앱 개발여러 시스템 없이 애플리케이션 QA 테스트데이터 센터 내에서 컴퓨팅 환경 실행 및 관리
  VMM은 기본적으로 실행할 수 있으며, 이 경우 호스트이기도 합
- 니다.
  - 범용 호스트 (VMware ESX 및 Citrix XenServer)는 없습니다.

컴퓨팅 환경 - 가상화


processes
kernel
hardware
(a)
programming interface
processes
processes
processes
kernel
kernel
kernel
VM1
VM2
VM3
virtual machine manager
hardware
(b)

분산 시스템

- 분산 컴퓨팅
  - 서로 네트워크로 연결된 분리된, 어쩌면 이질적일 수 있는 시스템의 모음
    - 네트워크는 통신 경로이며 TCP/IP가 가장 일반적입니다.

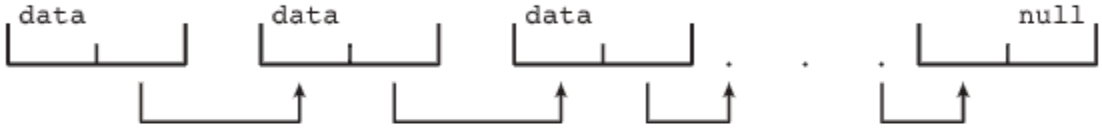
    - 근거리 통신망(LAN)– 광역 네트워크(WAN)– 대도시 지역 네트워크(MAN)– 개인 영역 네트워크(PAN)네트워크 운영 체제는 네트워크 전반의 시스템 간에 기능을 제공합니다

  -

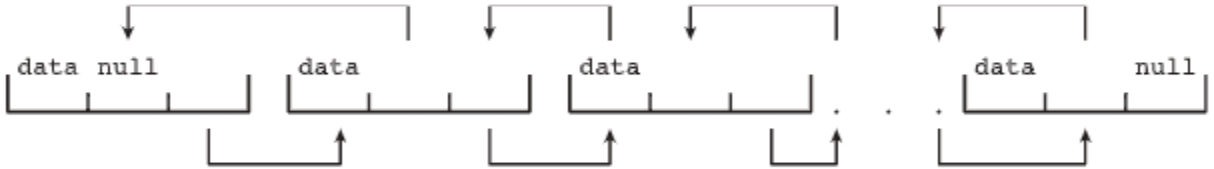
    - 통신 방식을 통해 시스템이 메시지를 교환할 수 있음
    - 단일 시스템의 환상

커널 데이터 구조체

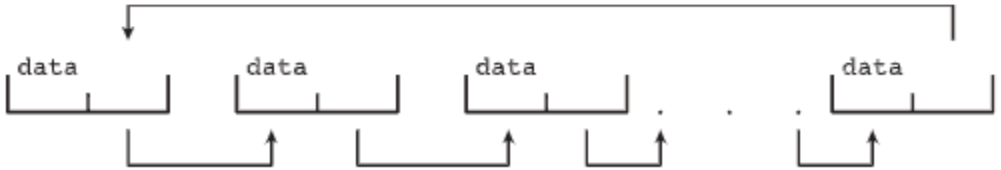
- 표준 프로그래밍 데이터 구조와 많이 유사합니다.
- Singly 연결된 목록



- 이중 연결 목록

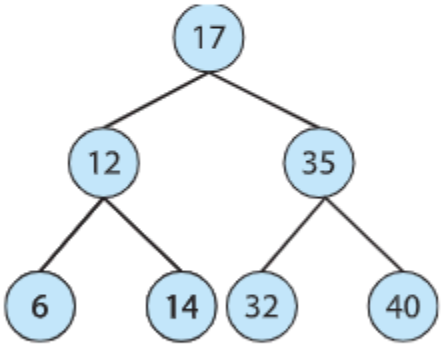


- 순환 연결 목록



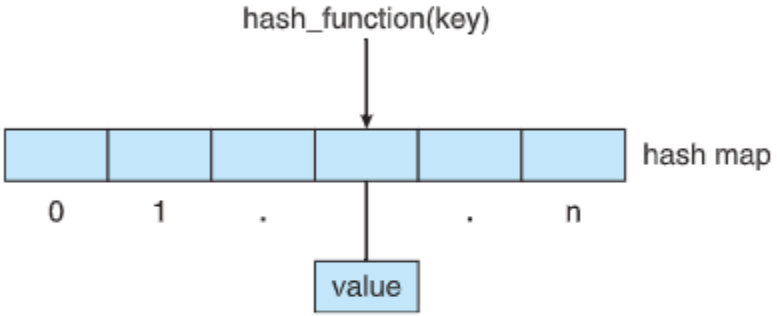
커널 데이터 구조(계속)

- **이진 검색 트리left <= right**
  - 검색 성능은 O(n)균형 이진 검색 트
  - 리는 O(lg n)입니다.



커널 데이터 구조(계속)

- 해시 함수는 해시 맵을 생성할 수 있습니다.



- 비트맵 – <linux/list.h>, <linux/kfifo.h>,<linux/rbtree.h 파일에 정의된 n개
- 의 itemsLinux 데이터 구조의 상태를 나타내는 n개의 이진수 문자열입니다>

컴퓨팅 환경 - 기존

- 독립형 범용 기계그러나 대부분의 시스템이 다른 시스템(예: 인터넷)과 상호
- 연결됨에 따라 모호함포털은 내부 시스템에 대한 웹 액세스를 제공합니다
- 네트워크 컴퓨터(씬 클라이언트)는 웹 터미널과 같습니다모바일 컴퓨터는
- 무선 네트워크를 통해 상호 연결네트워킹은 유비쿼터스가 되고 있으며 가
- 정용 시스템도 방화벽을 사용하여 가정용 컴퓨터를 인터넷 공격으로부터
- 보호합니다.

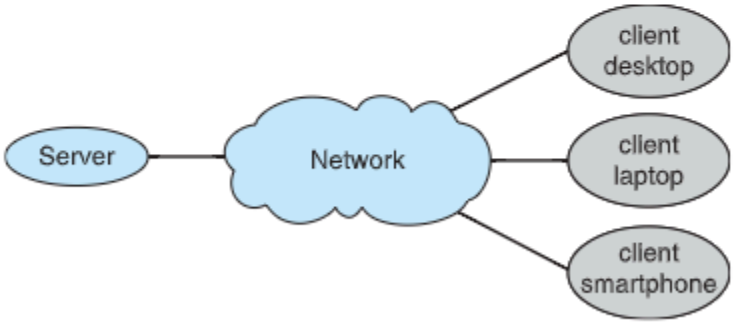
컴퓨팅 환경 - 모바일

- 휴대용 스마트폰, 태블릿 등이들과 "전통적인" 노트북의 기능적 차
- 이점은 무엇입니까? 추가 기능 – 더 많은 OS 기능(GPS, 자이로스
- 코프)증강 현실과 같은 새로운 유형의 앱 허용 연결을 위해 IEEE
- 802.11 무선 또는 셀룰러 데이터 네트워크 사용선두 주자는 Apple
- iOS 및 Google Android입니다.
-

컴퓨팅 환경 – 클라이언트-서버

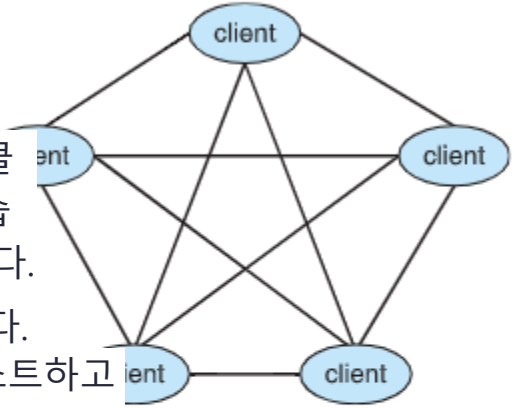
- 클라이언트-서버 컴퓨팅
  - 스마트 PC로 대체된 멍청한 터미널많은 시스템이 이제 서버를 구축하여
  - 클라이언트가 생성한 요청에 응답합니다.

    컴퓨트 서버 시스템은 클라이언트에게 서비스(즉, 데이터베이스)를 요청할 수 있는 인터페이스를 제공합니다. 파일 서버 시스템은 클라이언트가 파일을 저장하고 검색할 수 있는 인터페이스를 제공합니다.



컴퓨팅 환경 - 피어 투 피어

- 분산 systemP2P의 또 다른 모델은 클
- 라이언트와 서버를 구분하지 않습니다
  - 대신 모든 노드는 피어로 간주됩니다각각 클
  - 라이언트, 서버 또는 둘 다로 작동할 수 있습
  - 니다노드는 P2P 네트워크에 가입해야 합니다.
    - 중앙 조회 서비스에 서비스를 등록합니다.
      네트워크 또는 서비스 요청을 브로드캐스트하고
      요청에 응답
    - 디스커버리 프로토콜을 통한 서비스예를 들면 Napster 및
  - Gnutella, Skype와 같은 VoIP(Voice over IP)가 있습니다.



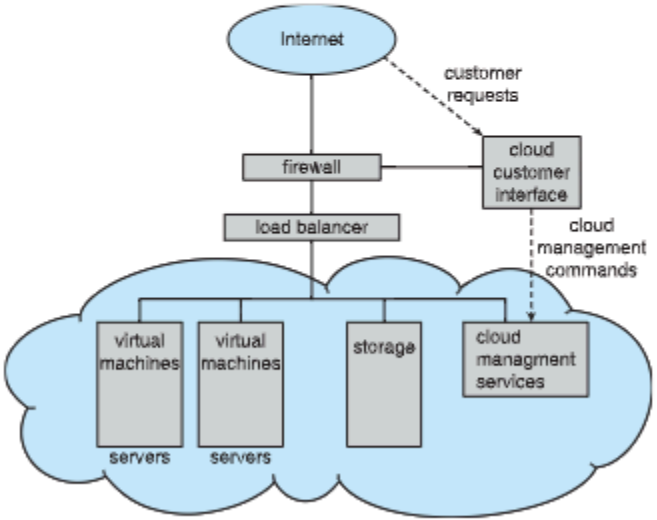
컴퓨팅 환경 – 클라우드 컴퓨팅

- 네트워크를 통해 컴퓨팅, 스토리지, 심지어 앱까지 서비스로 제공가상화를
- 기반으로 사용하기 때문에 가상화의 논리적 확장기능.

- Amazon EC2에는 수천 개의 서버, 수백만 개의 가상 머신, 인터넷을 통해 사용할 수 있는 페타바이트의 스토리지가 있으며 사용량에 따라 지불됩니다.다양한 유형
  - **퍼블릭 클라우드 – 비용을 지불할 의사가 있는 모든 사람이 인터넷을**
  - **통해 사용할 수 있음프라이빗 클라우드 – 회사 자체에서 사용하기 위해**
  - **회사에서 운영하는 하이브리드 클라우드 – 퍼블릭 및 프라이빗 클라우**
  - **드 구성 요소를 모두 포함Software as a Service(SaaS) – 인터넷을 통**
  - **해 사용 가능한 하나 이상의 애플리케이션(즉, 워드 프로세**
  - **서)PaaS(Platform as a Service) – 인터넷을 통해 애플리케이션을 사용**
  - **할 수 있는 소프트웨어 스택(즉, 데이터베이스 서**
  - **버)IaaS(Infrastructure as a Service) – 인터넷을 통해 사용할 수 있는**
  - **서버 또는 스토리지(즉, 백업용으로 사용할 수 있는 스토리지)**

컴퓨팅 환경 – 클라우드 컴퓨팅

- 기존 OS와 VMM, 클라우드 관리 도구로 구성된 클라우드 컴퓨팅 환경
  - 인터넷 연결에는 방화벽과 같은 보안이 필요합니다부하
  - 분산 장치는 트래픽을 여러 애플리케이션에 분산시킵니다.



컴퓨팅 환경 – 실시간 임베디드 시스템

- 실시간 임베디드 시스템: 가장 널리 사용되는 컴퓨터 형태

  다양한, 특수한, 제한적 인 OS, 실시간 OSUse 확장다른 많은 특수 컴퓨팅 환경뿐만 아니라.

-

  일부는 OS를 가지고 있고, 일부는 OS없이 작업을 수행하며
- real-time OS에는 잘 정의 된 고정 시간 제약 조건이 있습니다.
  - 처리는 constraintCorrect operation 내에서 수행
  - 되어야 합니다. 제약 조건이 충족되는 경우에만

무료 및 오픈 소스 운영 체제

- 바이너리가 아닌 소스 코드 형식으로 사용할 수 있는 운영 체제폐쇄 소스 및 독점복사 방지 및 DRM(Digital Rights Management) 운동에 대한 대
- 응"카피레프트" GNU PublicLicense(GPL)를 가진 FSF(Free Software Foundation)에 의해 시작
-
  - 자유 소프트웨어와 오픈 소스 소프트웨어는 서로 다른 그룹의 사람들이 옹호하는 두 가지 다른 아이디어입니다
    - http://gnu.org/philosophy/open-source-misses-the-point.html/
- 예를 들어 GNU / Linux 및 BSD UNIX (Mac OS X의 코어 포함) 등이 있습니다.VMware Player (Windows의 경우 무료), Virtualbox (오픈 소스 및 많은 플
- 랫폼에서 무료 - http://www.virtualbox.com)와 같은 VMM을 사용할 수 있습니다.
  - 탐색을 위해 게스트 운영 체제를 실행하는 데 사용합니다.

운영 체제 연구

운영 체제를 공부하기에 이보다 더 흥미로운 시기는 없었으며 그 어느 때보다 쉬웠습니다. 오픈 소스 이동은 운영 체제를 추월하여 많은 운영 체제가 소스 및 바이너리(실행 가능) 형식으로 제공되었습니다. 두 형식 모두에서 사용할 수 있는 운영 체제 목록에는 Linux, BUSDUNIX, Solaris 및 macOS의 일부가 포함됩니다. 소스 코드를 사용할 수 있기 때문에 운영 체제를 안팎으로 연구할 수 있습니다. 한때 문서나 운영 체제의 동작을 살펴보는 것만으로 대답할 수 있었던 질문들은 이제 코드 자체를 검사함으로써 대답할 수 있습니다.

더 이상 상업적으로 실행 가능하지 않은 운영 체제도 오픈 소스로 제공되어 CPU, 메모리 및 스토리지 리소스가 적은 시대에 시스템이 어떻게 작동했는지 연구할 수 있습니다. 광범위하지만 불완전한 오픈 소스 운영 체제 프로젝트 목록은 다음에서 사용할 수 있습니다.

https://curlie.org/Computers/Software/Operating_Systems/Open_Source/

또한 가상화가 주류(그리고 종종 무료) 컴퓨터 기능으로 부상함에 따라 하나의 핵심 시스템 위에서 많은 운영 체제를 실행할 수 있게 되었습니다. 예를 들어, Vmware (http://www.vmware.com)는 수백 개의 무료 "가상 어플라이언스"를 실행할 수있는 Windows 용 무료 "플레이어"를 제공합니다. Virtualbox(http://www.virtualbox.com)는 많은 운영 체제에서 무료 오픈 소스 가상 머신 관리자를 제공합니다. 이러한 도구를 사용하여 학생들은 전용 하드웨어 없이 수백 개의 운영 체제를 사용해 볼 수 있습니다.

오픈 소스 운영 체제의 출현으로 학생에서 운영 체제 개발자로의 이동도 더 쉬워졌습니다. 약간의 지식, 노력 및 인터넷 연결만 있으면 학생은 새로운 운영 체제 배포판을 만들 수도 있습니다. 불과 몇 년 전만 해도 소스 코드에 액세스하는 것이 어렵거나 불가능했습니다. 이제 이러한 액세스는 학생이 얼마나 많은 관심, 시간 및 디스크 공간을 가지고 있는지에 의해서만 제한됩니다.

챕터 1의 끝