



1주차 퀴즈

📅 Date	@2022년 12월 8일
🏷️ Tags	취준스터디
📎 출처	

네트워크

인터넷은 어떻게 동작하는가 ?

Q1. 네트워크의 시작인 Arpanet은 ____을 목적으로 만들어졌다. 괄호안에 들어갈 말을 작성하고, 괄호안에 들어가는 단어의 뜻을 예를들어 설명하시오.

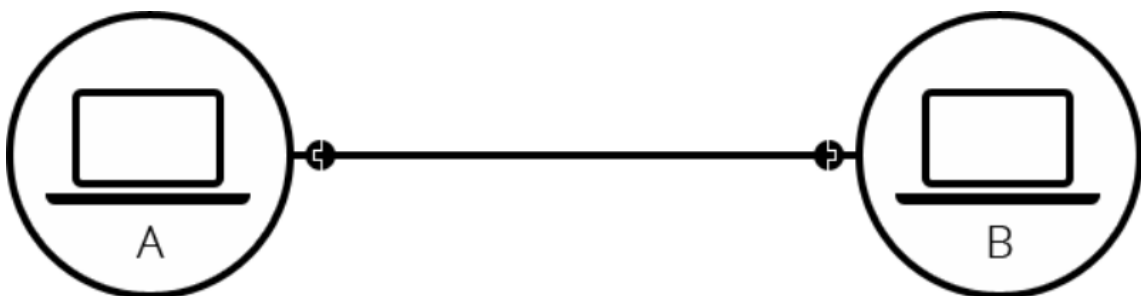
시분할

시분할 컴퓨팅이란 ? 각 사용자들에게 컴퓨터 자원을 시간적으로 분할하여 사용할 수 있게 해 주는 것을 말한다. 시분할이란 다음과 같은 상황을 극복하게 위해 만들어졌다. 출력이 사용자에게 표시되고 입력을 키보드에서 읽어들이는 대화식 인터페이스를 제공한다고 가정하자. 대화식 입출력 속도는 보통 사람들의 처리속도와 같으므로 종료할 때까지 오랜시간이 걸린다. 만약, 시분할 시스템이 없다면 컴퓨터는 입출력 처리를 끝마칠때까지 기다려야한다. 만약 시분할 시스템이 있다면 운영체제는 CPU를 그냥 쉬게 하지 않고 다른 사용자의 프로그램을 수행하도록 전환시킨다.

시스템은 한 사용자에서 다음 사용자로 빠르게 전환함으로써 각 사용자에게 자신만이 컴퓨터를 사용하고 있는 것과 같은 착각을 주지만,실제로는 여러 사용자가 하나의 컴퓨터를 공유하고 사용하고 있는 것이다.

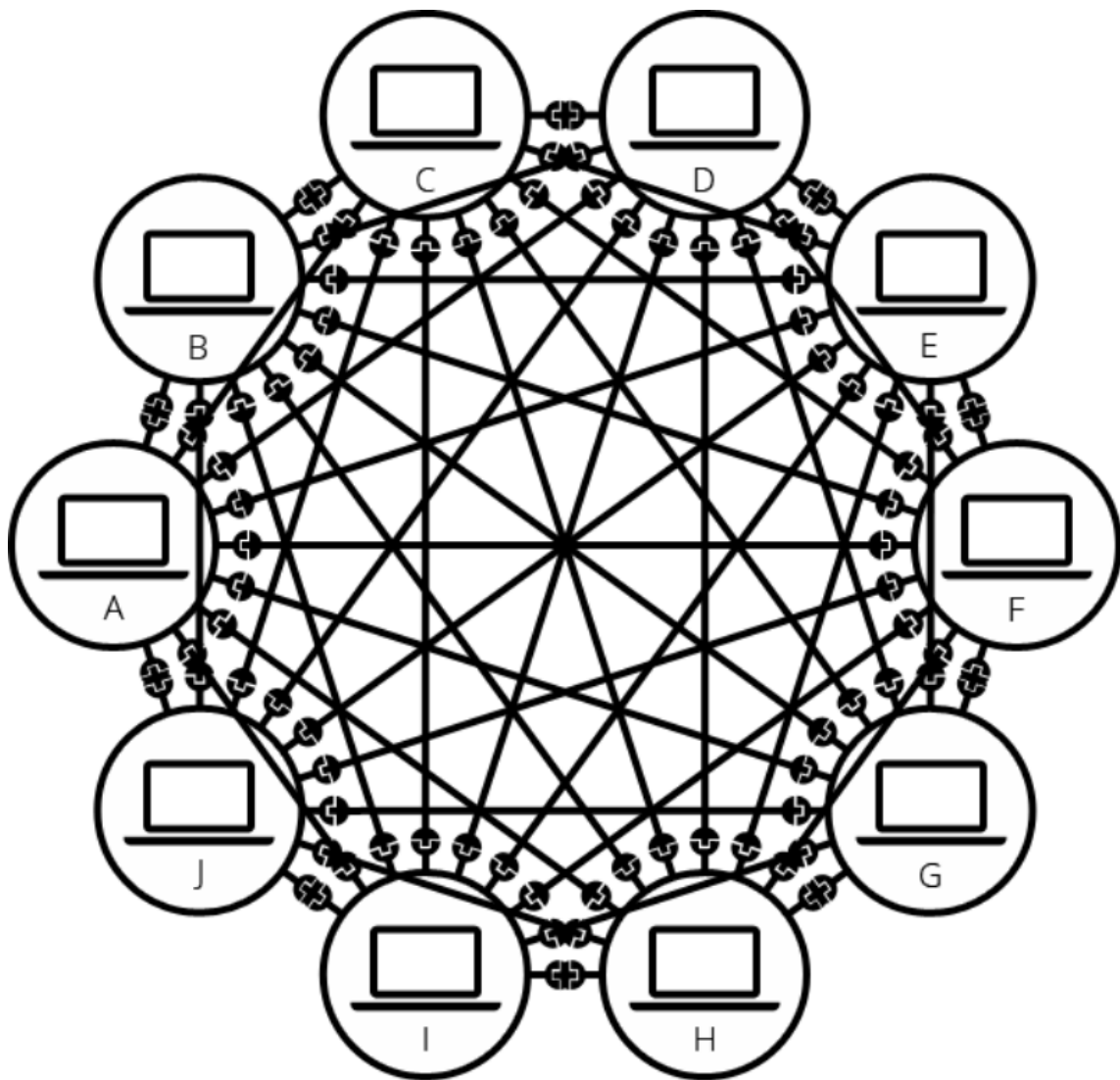
- 시분할 운영체제는 CPU 스케줄링과 다중 프로그래밍을 이용해서 각 사용자들에게 컴퓨터 자원을 시간적으로 분할하여 사용할 수 있게 해준다.
- 여러 명의 사용자가 사용하는 시스템에서 컴퓨터가 사용자의 프로그램을 번갈아가며 처리해줌으로써 각 사용자에게 독립된 컴퓨터를 사용하는 느낌을 주는 것
- 라운드로빈 방식이라고도 불림
- 하나의 CPU는 같은 시점에 여러개의 작업을 동시에 수행할 수 없기 때문에 CPU의 전체 사용시간을 작은 작업 시간량으로 쪼개어 그 시간량 동안만 번갈아 가면서 CPU를 할당하여 각 작업을 처리한다.
- 다중 프로그래밍 방식과 결합하여 모든 작업이 동시에 진행되는 것처럼 대화식 처리가 가능하다.

Q2. 인터넷의 작동원리는 어떻게 되는가 ? 소규모 네트워크부터 대규모 네트워크 인프라까지 어떻게 연결되고 구성되어있는지 서술하시오.

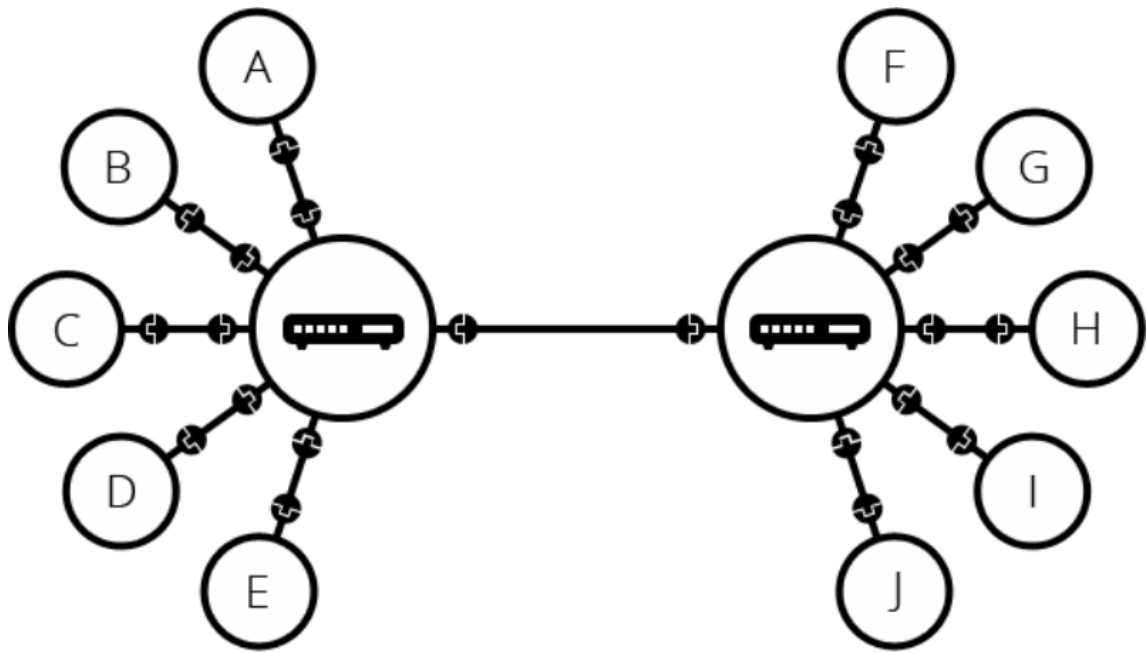


두개의 컴퓨터가 연결될때는 블루투스나 케이블선등 하나로 쉽게 연결이 가능하다.

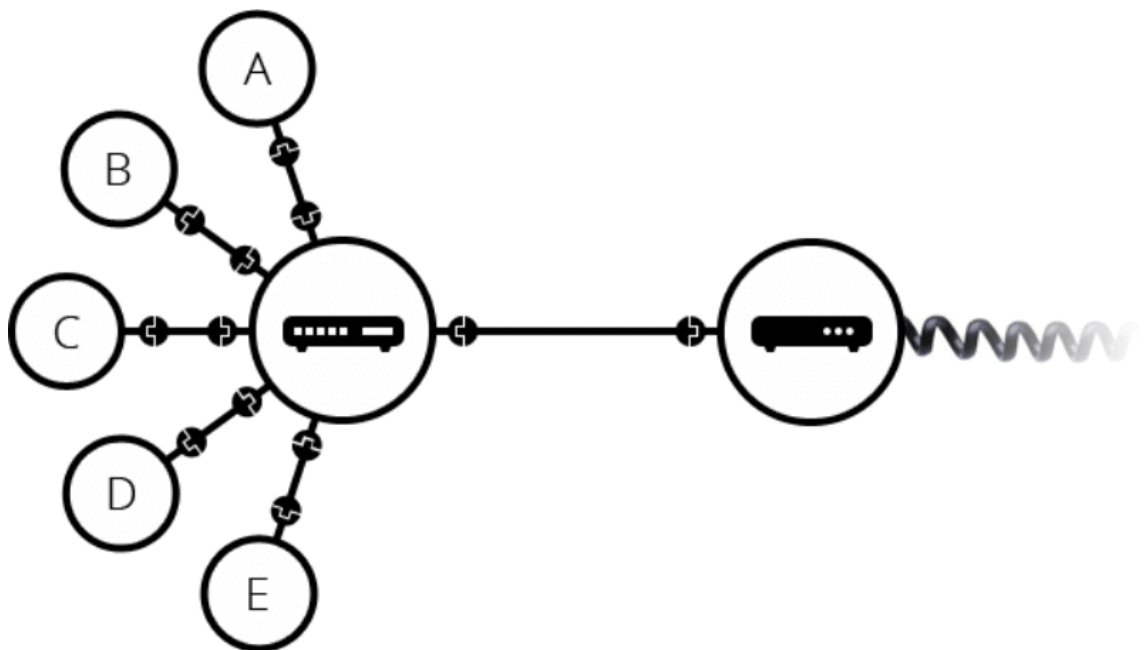
하지만 컴퓨터 열대를 연결할때는 ?



이렇게 보이는 것과 같이 수많은 갯수의 케이블이 필요하다. 즉, 소수의 컴퓨터를 연결할 경우와는 달리 여러개의 컴퓨터를 연결할 때 필요한 매체가 있는데 그것이 바로 라우터이다. 라우터의 없다면, 10대의 컴퓨터를 연결하는데 각 컴퓨터당 9개씩 필요할 것이고 총 90개의 케이블이 필요하게 될 것이다. 만약 라우터가 있다면? 모든 컴퓨터는 라우터로 데이터를 보내고 그 라우터에서는 알맞는 컴퓨터에 데이터를 보내주게 됩니다.



아쉽게도 수십 수천대의 컴퓨터는 라우터만으로는 확장이 불가능합니다. 아주 먼곳에 있는 지역과는 케이블 연결이 불가능하기 때문이죠, 이를테면 서울에 사는 사람과 부산에 있는 사람이 통신을 한다고 할 때 둘사이를 케이블로 연결하기는 힘듭니다. 하지만 이미 구성되어있는 케이블을 이용한다면 이는 어렵지 않습니다 전화 및 전력과 같이 이미 연결된 케이블이 있기 때문이죠. 전화기 반시설은 세계 어느곳과도 연결이 되어있기 때문에 네트워크는 이미 구성이 되어있는 상태인 것이다. 전화 시설과 연결하기 위해서는 전화시설이 처리할 수 있는 정보로 바꾸어주는 매개체가 필요한데 그것이 바로 모뎀입니다.

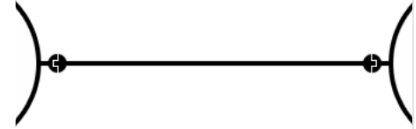


모뎀을 통해 전화국과 연결된 데이터가 다른 네트워크와 주고받기 위해서는 인터넷 서비스 제공업체인 ISP에 연결해야합니다. ISP는 모두 함께 연결되는 몇몇 특수한 라우터를 관리하며 다른 ISP 라우터에도 액세스 할 수 있습니다. 우리의 네트워크 메시지는 ISP 네트워크의 네트워크를 통해 대상 네트워크로 전달된다.

[IT 기술] 인터넷의 작동원리

인터넷의 작동원리 안녕하세요. 개발자 개발 까마귀입니다. 오늘 알려드릴거는 인터넷의 작동원리에 대해서 알려드리겠습니다. 1. 인터넷이란? 여러분들이 편하게 쓰는 인터넷은 사실 오랜 역사와 생각보다 복잡하게 구성되어 있습니다. 일단 인터넷을 알기 전에 우리는 TCP/IP가 뭔지에 대해서 알아야 합니다. TCP/IP란?

<https://development-crow.tistory.com/3>



알고리즘

1. 새로운 것을 삽입, 삭제하는 트랜잭션의 경우 Array와 Linked List중 어떤 것의 성능이 더 좋을까요? 배열 각각의 특징을 기반으로 설명해주세요

array의 경우 요소에 인덱스를 통해 접근할 수 있으며, 메모리 위치에 연이어 저장되기 때문에 중간에 요소가 삽입 삭제되면 요소를 하나하나 땡겨오는 작업이 필요하기 때문에 $O(n)$ 의 복잡도를 가지지만, 링크드 리스트의 경우 새로운 요소에 할당된 메모리 위치 주소가 linkedlist 이전 요소에 저장되기 때문에 중간에 삽입과 삭제가 있더라도 $O(1)$ 이 소요된다.

레퍼런스 = 참조 단어가 들어가면 좋을 것 같음

2. 단순연결리스트, 이중연결리스트, 원형 연결리스트를 연결리스트의 기본 구조 관점에서 비교하세요

단순연결리스트의 경우 노드가 하나의 포인터에 의해 다음 노드와 연결되는 구조를 가진다. 즉, 노드들을 한방향으로 연결하여 리스트를 구현하는 자료구조이다.

반면, 이중 연결 리스트의 경우 양쪽 방향으로 소통이 가능하게 노드를 연결한 리스트이다.

즉, 각 노드가 두개의 레퍼런스를 가지고 각각 이전 노드와 다음 노드를 가리키는 연결리스트이다.

단순연결리스트 중에서도 마지막 노드가 첫 노드와 연결된 것을 원형 연결 리스트라고 한다. 마지막 노드를 참조하는 포인터가 단순 연결 리스트의 head 역할을 한다.

운영체제

Q1. 프로세스와 스레드의 정의를 간략히 설명하시오.

프로세스의 경우 실행될 때 운영체제로부터 프로세서, 필요한 주소 공간, 메모리 등의 자원을 할당받는 작업의 단위이다. 스레드는 프로세스가 할당받은 자원을 이용하는 실행의 단위이다.

Q2. 프로세스 문맥은 현재 프로세스가 어디까지 실행되었는지를 파악하는 것이다. 그렇다면, 프로세스 문맥 파악이 필요한 이유에 대해 설명하시오.

프로세스의 현재 상태를 나타내는데 필요한 모든 요소들을 프로세스 문맥이라고 한다. 특정 시점에 프로세스가 어느만큼 작업을 했고 이 프로세스가 어디까지 와있는지 확인하기 위해서 필요하다. CPU는 계속 여러 프로세스를 번갈아 담당하기 때문에 어떤 프로세스를 어디까지 작업했는지 알아야한다. 그렇지 않으면 매번 처음부터 다시 실행해야 하기 때문이다.

없다면, 프로세스 제어권이 넘어왔을 때 다시 실행해야한다.

운영체제

Q1. 비동기와 논블로킹의 차이에 대해서 설명해주세요

동기와 비동기는 처리해야 할 작업들을 어떠한 흐름으로 처리 할 것인가에 대한 관점. 즉, 호출되는 함수의 작업 완료 여부를 신경 쓰냐에 따라 함수 실행/리턴 순차적인 흐름을 따르느냐, 안따르느냐가 관심사이다.

비동기는 시작,종료가 일치하지않는다. A함수가 다른 함수인 B를 호출할 때 콜백함수를 함께 전달하여 B의 작업이 완료되면 함께 보낸 콜백 함수를 실행되는것을 말한다. 함수 A는 함수 B의 작업 완료 여부에는 신경쓰지 않는다.

블로킹과 논블로킹의 경우 처리되어야 하는 하나의 작업이 전체적인 작업 흐름을 막느냐 안막느냐에 대한 관점이다 → 즉, **제어권이 누구한테 있느냐**가 관심사이다.

논블로킹이란 다른 주체의 작업에 관련없이 자신의 작업을 하는 것이다. A함수가 B 함수를 호출해도 제어권은 그대로 자신이 가지고 있다.

비동기의 경우 다음 작업이 요청되는 시간을 신경쓴다. 현재 작업의 응답이 끝나지 않은 상태에서 다음 작업을 요청한다. 이와 반대로 동기는 현재 작업의 응답이 끝남과 동시에 다음 작업을 요청한다.

논블로킹의 경우 함수의 리턴시점과 제어권을 신경쓴다. 제어권이 계속 호출한 함수에 있기 때문에 작업의 완료여부와 관계없이 새로운 작업을 수행할 수 있다. 이와 반대로 블로킹의 경우 제어권이 호출된 함수로 넘어가서 호출된 함수가 작업을 마무리 된 후 함수를 호출한 함수로 제어권이 넘어간다.

예시

선생님(상위 프로세스) 학생(하위 프로세스)

동기 + 블로킹

- 입실 후 학생들의 자습이 모두 끝날때까지 아무것도 하지 않으면서 기다리고, 학생들의 자습이 끝나면 퇴실한다.
- 모든 실행과 흐름이 순차적이기 때문에 개발자가 프로그램을 제어하기 쉽다.
- 선생님은 학생들의 자습완료 여부를 신경쓴다.
- 블로킹 방식이기 때문에 하위 프로세스인 학생의 작업이 완료되지 않으면 상위프로세스인 선생님은 다른 작업을 할 수 없다.

// 출력 결과

선생님: 입실

학생들: 1번 문제 푸는중...

학생들: 2번 문제 푸는중...

학생들: 3번 문제 푸는중...

학생들: 4번 문제 푸는중...

학생들: 5번 문제 푸는중...

학생들: 6번 문제 푸는중...

학생들: 7번 문제 푸는중...

학생들: 8번 문제 푸는중...

학생들: 9번 문제 푸는중...

학생들: 10번 문제 푸는중...

선생님: 퇴실

동기 + 논블로킹

- 딴짓을 열심히 하는 선생님, 입실 후 학생들의 자습이 모두 끝날때까지 기다리면서 딴짓을 하면서 기다리고, 학생들의 자습이 끝나면 퇴실한다.
- 선생님은 학생들의 작업완료 여부를 신경쓴다.
- 논블로킹 방식이기 때문에 학생들의 작업 완료 여부와 상관없이 선생님은 다른 작업을 할 수 있다.

선생님: 입실
학생들: 1번 문제 푸는중...
선생님: 딴짓
학생들: 2번 문제 푸는중...
선생님: 딴짓
학생들: 3번 문제 푸는중...
선생님: 딴짓
학생들: 4번 문제 푸는중...
선생님: 딴짓
학생들: 5번 문제 푸는중...
선생님: 딴짓
학생들: 6번 문제 푸는중...
선생님: 딴짓
학생들: 7번 문제 푸는중...
선생님: 딴짓
학생들: 8번 문제 푸는중...
선생님: 딴짓
학생들: 9번 문제 푸는중...
선생님: 딴짓
학생들: 10번 문제 푸는중...
선생님: 딴짓
선생님: 퇴실

비동기 + 논블로킹

- 자신의 시간이 소중한 선생님. 입실하자마자 자습을 지시하고 바로 퇴실한다. 그리고 학생들의 자습이 끝나면 반장을 통해 보고받는다.
- 선생님은 학생들의 작업완료여부를 신경쓰지 않는다.
- 논블로킹 방식이기 때문에 학생들의 작업완료 여부와 상관없이 선생님은 다른 작업을 할 수 있다.

```
// 출력 결과
선생님: 입실
선생님: 퇴실
학생들: 1번 문제 푸는중...
학생들: 2번 문제 푸는중...
학생들: 3번 문제 푸는중...
학생들: 4번 문제 푸는중...
학생들: 5번 문제 푸는중...
학생들: 6번 문제 푸는중...
학생들: 7번 문제 푸는중...
학생들: 8번 문제 푸는중...
학생들: 9번 문제 푸는중...
학생들: 10번 문제 푸는중...
반장: 선생님께 자습이 끝났음을 보고
```

비동기 + 블로킹

- 선생님은 학생들의 작업 완료 여부를 신경쓰지 않는다.
- 블로킹 방식이기 때문에 학생들의 작업이 완료되지 않으면 선생님은 다른 작업을 할 수 없다.
- 사실상 동기, 블로킹과 비슷한 작업 효율이 나온다

Q2. 비동기와 블로킹은 함께 쓰이는 경우가 잘 없습니다. 그러한 이유가 무엇일까요?

- 동기, 블로킹과 비슷한 작업 효율이 나오기 때문이다. 사실상 이렇게 되면, 비동기를 사용하는 이유가 없어지는 것이나 마찬가지이기 때문이다.
- 해당 방식은 논블로킹으로 하려다가 실수하는 경우에 발생하는등의 의도치 않게 사용되는 경우가 많다.
- Node.js + MySQL의 조합이 대표적이다. 노드에서 비동기적으로 DB작업을 수행하더라도 블로킹 방식으로 작동하는 MySQL 드라이버를 거치게된다.

