



2주차 퀴즈

📅 Date	@2022년 12월 15일
☰ Tags	취준스터디
🔗 출처	

TCP/IP

1. TCP와 UDP의 차이점과 그 특징은 무엇인지 서술하시오

TCP : 연결지향형 전송규약

- 흐름 중심 프로토콜
- 통신을 주고받는 것을 중요시함
- 중간에 패킷이 손실되는 경우 재전송을 통해(SYN-ACK handshaking) 신뢰성을 보장함(느림) 대부분의 통신에서 사용됨
- 파일이나 데이터 전송 시에 사용
- 가상회선

UDP : 비연결지향형 전송규약

- 데이터 중심 프로토콜, 주고받는 통신보다 데이터를 일방적으로 보내는 것을 중요시함
- 데이터 전송의 신뢰성 보장 X, (빠름)
- P2P, 스트리밍, 전화에 사용
- 데이터그램

2. 3 handshaking , 4 handshaking 과정을 서술하시오

SYN (Synchronization : 동기화) - S : 연결 요청 플래그

TCP에서 세션을 성립할 때 가장 먼저 보내는 패킷

시퀀스 번호를 임의적으로 설정하여 세션을 연결하는 데에 사용되며 초기에 시퀀스 번호를 보내게 됨

ACK (Acknowledgement : 승인) - ACK

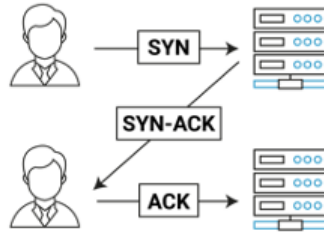
상대방으로부터 패킷을 받았다는 걸 알려주는 패킷, 다른 플래그와 같이 출력되는 경우도 있다.

받는 사람이 보낸 사람 시퀀스 번호에 TCP 계층에서 길이 또는 데이터 양을 더한 것과 같은 ACK 를 보낸다. 일반적으로 +1 하여 보냄

ACK 응답을 통해 보낸 패킷에 대한 성공,실패를 판단하여 재전송하거나 다음 패킷을 전송한다.

04. CONTENTS

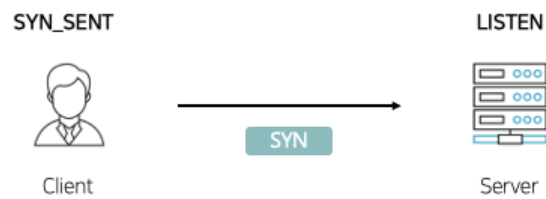
3 Way Handshaking



1. 서버와 클라이언트 사이에 커넥션을 생성
2. 서버는 클라이언트 노드로부터 SYN 패킷을 받습니다.
3. 클라이언트 노드는 서버로부터 SYN/ACK 패킷을 받은 뒤 ACK 패킷을 보내 응답합니다.

04. CONTENTS

3 Way Handshaking



클라이언트가 서버에 연결 요청을 하기 위해 SYN 데이터를 보냄

04. CONTENTS

3 Way Handshaking



서버는 LISTEN 상태에서 SYN_RCV로 상태로 변경한 뒤
요청을 정상적으로 받았다는 것 (ACK)과 포트를 열어달라는 (SYN) 을 클라이언트로 보냄

04. CONTENTS

3 Way Handshaking



클라이언트는 ESTABLISHED로 상태를 변경한 뒤
요청을 정상적으로 받았다는 것 (ACK) 을 서버에 보냄

04. CONTENTS

3 Way Handshaking



ACK 패킷을 받은 서버는 상태가 ESTABLISHED로 변경됨

04. CONTENTS

4 Way Handshaking



1. 클라이언트가 연결을 종료하겠다는 FIN 플래그를 전송
2. 서버는 확인 메시지를 보내고 자신의 통신이 끝날때까지 기다린다 (TIME_WAIT 상태)
3. 서버가 통신이 끝났으면 연결이 종료되었다고 클라이언트에게 FIN 플래그를 전송한다.
4. 클라이언트는 확인했다는 메시지를 보낸다.

04. CONTENTS

4 Way Handshaking



클라이언트가 서버에 연결 종료를 위해 FIN 패킷을 보냄

04. CONTENTS

4 Way Handshaking



서버는 클라이언트로부터 FIN 을 받고 CLOSE_WAIT 상태로 변경함

요청을 정상적으로 받았다고 (ACK) 클라이언트로 보냄

04. CONTENTS

4 Way Handshaking



연결을 종료할 준비를 마친 서버는
클라이언트에게 FIN 패킷을 보내고 LAST_WAIT 상태가 됨

04. CONTENTS

4 Way Handshaking

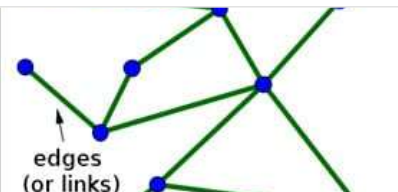


클라이언트는 서버의 연결 종료를 확인했다는 ACK를 보냄
이후, 클라이언트는 TIME_WAIT 상태가 됨

[네트워크(network)] 면접 예상 질문과 답변

1. 네트워크란? : 물리적 전송 매체를 사용하여 서로 연결된 장치 세트 - 컴퓨터 네트워크는 하드웨어, 데이터 및 소프트웨어와 같은 정보 및 리소스를 통신하고 공유하기 위해 서로 연결된 컴퓨터 그룹 - 네트워크에서 노드는 둘 이상의 네트워크를 연결하는 데 사용됨 1-1.

<https://hyonee.tistory.com/136>



운영체제

데드락의 발생조건

상호배제

자원은 하나의 프로세스만 사용할 수 있다. **비공유**를 전제해야함

근본적으로 상호배제 조건을 방지한다는 것은 불가능하다. 자원을 근본적으로 공유하는 것 자체가 불가능하기 때문. 상호배제 조건을 방지한다는 것으로는 교착상태를 예방하기에는 어렵다.

점유대기

프로세스가 할당된 자원을 가진 상태에서 다른 자원을 기다린다.

비선점

자원들을 선점할 수 없다. 자원이 강제적으로 방출될 수 없고, 점유하는 프로세스가 태스크를 종료한 후 그 프로세스에 의해서만 방출될 수 있다.

즉, 프로세스가 어떤 자원의 사용을 끝낼 때 까지 그 자원을 뺏을 수 없다.

순환대기

각 프로세스는 순환적으로 다음 프로세스가 요구하는 자원을 가지고 있다.

시스템 상태가 아래와 같을 때(이미지), 이 시스템은 안전한 상태인가? 맞다면 Safe Sequence를 도출하시오. (Need 행렬 표시)

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	1	1	1	7	5	4	1	2	3
P1	2	1	0	4	1	2			
P2	3	0	2	7	1	6			
P3	2	1	0	2	1	3			
P4	0	0	1	5	3	2			

은행원 알고리즘 → 자원의 할당 허용 여부를 결정하기 전에 미리 결정된 모든 자원의 최대 가능한 할당량을 시뮬레이션하여 안전 여부를 검사한다. 그런 다음 대기중인 다른 모든 활동의 교착 상태 가능성을 조사하며 “안전상태” 여부를 검사 확인한다.

need 행렬이란 = 최대요구량 - 할당량

6	4	3
2	0	2
4	1	4
0	0	3
5	3	1

work, finish 초기화

work = available(1,2,3) / finish = 프로세스 갯수 (false,false,false,false,false)

finish[0] = false need[0] = 6,4,3 ≤ work[0] = 1 2 3

finish[1] = false need[1] = 3,0,2 ≤ work[1] = 1 2 3

얼마나 필요한지, 현재 할당가능한 량을 비교하면서 만약, 할당가능하다면 동작을 수행한 뒤, 할당한 작업량들을 해제한다 (이 과정에서 사용가능한 작업량이 늘어난다) 그렇지 않다면 수행하지 않고 다음 프로세스로 넘어간다.

3 → 1 → 2 → 0 → 4

도커와 쿠버네티스

1. 쿠버네티스에 대해 설명하세요

조건 1. 쿠버네티스의 특징을 최소 2가지 이상 언급하세요

조건 2. 다음의 키워드는 반드시 포함해야 합니다(컨테이너, 오케스트레이션,도커)

쿠버네티스는 컨테이너를 오케스트레이션(여러 서버에 걸친 컨테이너 및 사용하는 환경 설정을 관리하는 행위)하는 도구 .즉, 컨테이너 런타임을 통해 컨테이너를 다루는 도구. 여러서버에 컨테이너를 분산해서 배치하거나 문제가 생긴 컨테이너를 교체하거나 컨테이너가 사용할 비밀번호나 환경설정을 관리하고 주입해주는 일을 함 이것을 컨테이너 오케스트레이션이라고함.

특징에는 특정 컨테이너가 죽었을 때 컨테이너를 복제 생성하여 서비스를 유지하거나 , 사용자의 접속량이 준비된 리소스가 감당할 수 있는 선을 넘을 경우 새로운 컨테이너를 생성하는 로드밸런싱을 한다.

추가적으로 무중단서비스를 제공한다. 서비스를 중단하지 않고도 애플리케이션을 업데이트 할 수 있다. 호환성의 경우 사용자가 클라우드의 환경을 이전하고 싶을 때 서로 다른 업체의 클라우드 제품간에 호환 문제가 발생하지 않도록 함

2. 도커 아키텍처의 작동원리를 설명하세요

도커는 내부적으로 클라이언트 - 서버 구조를 사용한다. 도커 데몬이라고 불리는 데몬을 실행하여 컨테이너와 관련된 모든 요청을 처리한다. 도커의 기본 CLI 클라이언트는 UNIX 소켓 혹은 네트워크 인터페이스를 통해 도커 데몬으로 REST API 요청을 보내게 된다.

도커는 기본적으로 clinet - server로 나누어져 있다. 클라이언트에서 docker 커맨드를 사용하면 도커 서버는 REST API형식으로 명령을 전달받아 수행하게 된다. 이 때문에 원격으로 도커 서버에 명령을 전달하거나 여러 개의 도커 서버에 접근 할 수 있으며 하나의 서버 또한 여러개의 클라이언트에서 명령을 받을 수 있다.

이미지 생성 , 이미지 컨테이너

컴퓨터는 운영체제 없이도 동작하나요?(Yes or No) 그 이유는 무엇인가요?

동작합니다. 운영체제란 컴퓨터와 유저사이에 다리를 놓아주는 가교역할을 하는것뿐이기 때문에 사용자가 동작에 대한 코드를 직접 짜주면 가능합니다..

운영체제 없이 돌아가기 위해서는 사용자 입력을 했을때 컴퓨터가 어떻게 인식을 해야하고 컴퓨터에서는 들어온 데이터를 어떻게 처리해야하고 남는 데이터(가비지) 혹은 남는 프로세서를 어떻게 사용해야하는지 등 가이드라인을 사용자가 직접 정의해주어야 합니다.

인터페이스 제공, 자원의 효율적인 관리를 위함 → 성능향상과 접근성을 위함이기 때문

시스템 콜의 정의는 무엇인가요?

시스템 콜은 운영체제의 커널이 제공하는 서비스에 대해, 응용프로그램의 요청에 따라 커널에 접근하기 위한 인터페이스입니다.

시스템 콜은 프로그램의 거의 모든 코드의 실행에서 발생하며 파일 생성이나 쓰기 또는 읽기,키보드 입력,스레드 생성 및 제어 같은 것들이 시스템 콜을 통해 커널에 요청하여 커널모드에서 처리합니다.

응용프로그램의 기능들이 수행되기 위해서는 커널의 도움이 필요하다. 그러나 유저 프로세스의 경우 커널에 접근이 불가능한데, 왜냐하면 커널에 관련된 것들은 커널 모드로 전환한 후에야 작업을 수행할 수 있기 때문이다. 이렇기 때문에 커널모드를 통한 작업은 시스템 콜을 거쳐야지 수행할 수 있기 때문에 항상 시스템 콜을 거쳐야 한다.

페이징은 내부 단편화가 일어난다. 일어나는 이유를 설명하라

페이징은 분할 분산 메모리 할당기법으로 크기가 동일한 영역인 페이지로 나누어 처리하는 방법을 말합니다. 페이징의 경우 일정 크기의 페이지로 나누어 할당하기 때문에 필요한 메모리보다 더 큰 크기로 나누어 질 경우 남는 공간이 발생한다. 이럴 때 메모리가 낭비되는 현상을 내부 단편화라고 한다.

고정된 크기로 인해 내부 단편화가 일어난다.

세그멘테이션은 내부 단편화가 일어나지 않는다. 그 이유를 페이징과 비교하며 서술하시오

세그멘테이션의 경우 가상메모리를 크기를 똑같이 나누어 물리적 단위로 나눈 페이징과 달리 서로 크기가 다른 논리적 단위인 세그먼트로 분할해 메모리를 할당하여 실제 메모리 주소로 변환을 하게 된다. 세그먼트들의 크기가 다르기 때문에 미리 분할해 둘 수 없다. 그 때문에 메모리에 적재될 때 빈공간을 찾아 할당하는 기법이다. 외부 단편화 문제가 발생한다.