Array의 가장 큰 특징은 순차적으로 데이터를 저장하는 것(순서보장)

중간에 요소가 삽입되거나 삭제되는 경우 그 뒤의 모든 요소들에도 조치를 취해줘야 되서

자주 삭제되거나 추가되는 데이터를 담기에는 적절하지 않음

좋은 예)주식 차트

스택과 큐는 선형 구조 Array와 LlinkedList로 구현 가능

스택은 자바의 Stack 메모리영역(지역변수와 매개변수 데이터값이 저장되는 공간이며 LIFO구조를 가짐

큐는 OS의 스케줄러

스케줄러는 자원의 할당과 회수를 담당하는데 선입선처리 스케줄링에서 사용할 수 있음

Tree는 비선형 자료구조이며 계층적 관계를 표현하기에 적합

Heap은 최대값 또는 최솟값을 찾아내는 연산을 쉽게 하기 위해 고안된 구조

우선순위 큐는 들어간 순서에 상관없이 우선순위가 높은 데이터를 먼저 꺼내기 위해 고안된 자료구조 시간복잡도가 O(logN)을 보장하기 떄문에 일반적으로 완전 이진트리 형태의 힙을 이용해 구현

Array는 크기가 고정적이고 ArrayList는 크기가 가변적

Array는 ArrayList보다 속도가 더 빠르다

Array는 검색이 빠르지만 추가나 삭제가 느리다

LinkedList는 삽입삭제가 빠르지만 검색이 느리다

Hash Map은 멀티쓰레드환경에서 동기화처리가 필요함 null값을 허용

HashTable은 멀티쓰레드환경에서 동기화처리가 필요하지않음 null값 허용 x

프로세스는 실행중인 프로그램을 말하며 독립적이기 때문에 메모리 영역(Code, Data, Heap, Stack)을 다른 프로세스와 공유하지 않음 최소 1개의 쓰레드를 가지고 있음

쓰레드는 프로세스 내에서 스택만 따로 할당 받고 그 외의 메모리영역은 공유하기 떄문에 다른 쓰레드의 실행결과를 확인할 수 있음 쓰레드는 프로세스 내에 존재하며 프로세스가 할당받은 자원을 이용하여 실행됨

멀티 프로세스는 서로 다른 프로세스가 영향을 끼치지 않지만 많은 메모리 공간과 cpu 시간을 차지함

멀티 쓰레드는 하나의 쓰레드의 문제가 발생시 전체의 쓰레드가 영향을 받으며 동기화 문제가 생기지만 적은 메모리 공간을 차지하고 문맥전환이 빠르다는 장점이 있음

동시성은 싱글 코어에서 여러 개의 쓰레드가 번갈아 실행하는 것을 말함

병렬성은 멀티코어에서 각 코어들을 동시에 실행하는 것을 말함

멀티 쓰레드 환경에서 주의 해야할 점 동기화 기법을 통해 동시성 문제 또는 교착상태가 발생하지않도록 주의해야함

데드락은 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 상황

4가지조건 :

비선점 순환대기 점유대기 상호배제

콘보이 현상이란 작업시간이 긴 프로세스가 먼저 큐에 도착해서 다른 프로세스의 실행시간이 전부 늦어지는 것

FCFS(first come First served)스케줄링은 비선점형으로 콘보이 현상이 발생할 수 있음

선점형 스케쥴링은 하나의 프로세스가 다른 프로세스 대신에 CPU를 차지할 수 있음을 말하고

비선점형 스케쥴링은 하나의 프로세스가 끝나지 않으면 다른 프로세스는 CPU를 사용할 수 없음을 말함

동기는 순차적 서버에 데이터를 요청하고 데이터가 응답할 때까지 다른 데스크들은 블락(작업중단)되고

비동기는 병렬적으로 서버에 데이터를 요청한 이후 서버로부터 응답할 때까지 대기하지않고(non-blocking) 다음 데스크를 수행함

Crical Section(임계영역)

임계영역이란 프로세스간에 공유자원을 접근하는데 있어 문제가 발생하지 않도록 한번에 하나의 프로세스만 이용하게끔 보장해줘야되는 영역

해결조건 3가지

상호배제 하나의 프로세스가 임계영역에 들어가있다면 다른 프로세스는 들어갈 수 없다

진행 임계영역에 들어간 프로세스가 없는 상태에서 들어가려 하는 프로세스가 여러 개라면 어느 것이 들어갈지 결정 해주어야 한다

한정대기 다른 프로세스의 기아를 방지하기 위해 한번 임계구역에 들어간 프로세스는 다음번 임임계영역 들어갈 때 제한을 두어야 한다

뮤텍스와 세마포어의 차이

뮤텍스는 Lock을 사용해 하나의 프로세스나 쓰레드를 단독 실행하게 합니다

세마포어는 공유자원에 세마포어 변수만큼의 프로세스가 접근할 수 있음

뮤텍스 : 임계 구역을 가진 스레드들의 실행시간이 서로 겹치지 않도록 각각 단독으로 실행되게하는 기술

세마포어 : 멀티프로그래밍 환경에서 공유 자원에 대한 접근을 제한하는 방법

페이징 교체 알고리즘이란

페이징 기법으로 메모리를 관리하는 운영체제에서 필요한 페이지가 주기억장치에 페이징 부재시 어떤 페이지 프레임을 선택해 교체할것인지 결정하는 방법

FIFO 메모리에 올라온지 가장 오래된 페이지를 교체함

최적페이지 교체 앞으로 가장 오랫동안 사용되지 않을 페이지를 교체

하지만 프로세스가 앞으로 사용할 페이지를 미리 알아야되서 구현불가능한 알고리즘

LRU(Least-recently-used)

가장 오래 사용되지않는 페이지를 교체하는 알고리즘

LFU(Least-frequency-used)

참조횟수가 가장 적은 페이지를 교체하는 알고리즘

MFU(Most-frequency-used)

참조횟수가 가장 많은 페이지를 교체하는 알고리즘

컨텍스트(문맥)이란 CPU가 해당 프로세스를 실행하기 위한 해당 프로세스의 정보들

멀티 프로세스 환경에서 CPU가 어떤 하나의 프로세스를 실행하고 있는 상태에서 인터럽트 요청에 의해 다음 우선 순위 프로세스가 실행되어야할 때 기존의 프로세스 상태 값을 저장하고 CPU가 다음 프로세스를 수행하도록 새로운 프로세스의 상태를 교체하는 작업

운영체제란 ?

컴퓨터 하드웨어 바로 위에 설치되어 사용자 및 다른 모든 소프트웨어와 하드웨어를 연결하는 소프트웨어 계층

**블로킹/논블로킹**

= 요청받는 함수가 제어권(함수실행권)을 언제 넘겨주느냐의 차이

* 블로킹 : 요청받는 함수가 작업을 모두 마치고 나서야 요청자에게 제어권을 넘겨줌 (그동안 요청자는 아무것도 하지않고 기다림)
* 논블로킹 : 요청받은 함수가 요청자에게 제어권을 바로 넘겨줌 (그동안 요청자는 다른 일을 할 수 있음)

**동기/비동기**

= 요청받은 함수가 작업을 완료했는지를 체크해서 순차적 흐름의 차이

* 동기 : 요청자가 요청받은 함수의 작업이 완료되었는지 계속 확인 (여러 함수들이 시간을 맞춰 실행됨)
* 비동기 : 요청자는 요청후 신경X, 요청받은 함수가 작업을 마치면 알려줌 (함수들의 작업 시작/종료 시간이 맞지 않을수도)

**동기와 비동기**는 **프로세스의 수행 순서 보장에 대한 매커니즘**이고 **블록킹과 논블록킹**은 **프로세스의 유휴 상태에 대한 개념**으로 완전한 별개의 개념이라는 것이다.