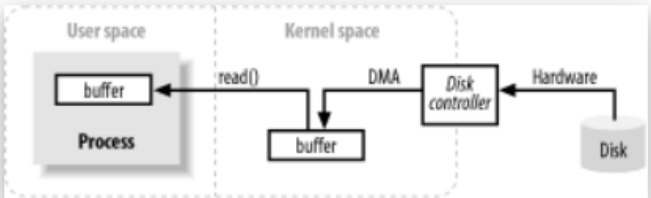
22.12.26

IO

- 커널 버퍼에서 JVM내의 Buffer로 복사해서 한번 더 옮겨주는 과정이 필요하다.
- JVM 내부 버퍼 복사시 발생하는 CPU연산, GC관리, IO 요청에 대한 스레드 블록 현상으로 효율이 안좋아진다.
  - `커널 I/O 버퍼 -> 유저 I/O 버퍼`로 복사 후 이용해야 하기에 복사완료될때 까지 스레드가 봉쇄, 시스템 콜이 발생할 수 있다.
  - 입출력 데이터가 준비될때까지 무한정 `block`되어 여러 클라이언트의 입출력을 동시에 처리하려면 스레드를 여러개 만들어야 한다.
- IO는 스트림에서 읽은 데이터를 즉시 처리하므로 스트림으로부터 입력된 전체 데이터를 별도로 저장하지 않으면, 입력된 데이터의 위치를 이동해 가면서 자유롭게 이용할 수 없다
- 데이터 IO를 stream으로 읽는다.
- BufferedInputStream, BufferdOuputStream 단방향
- 블로킹
- 연결 클라이언트가 적고 IO가 큰경우

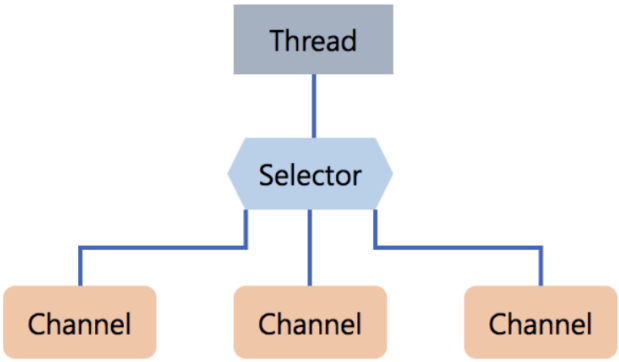


(출처 : https://howtodoinjava.com/java/io/how-java-io-works-internally/)

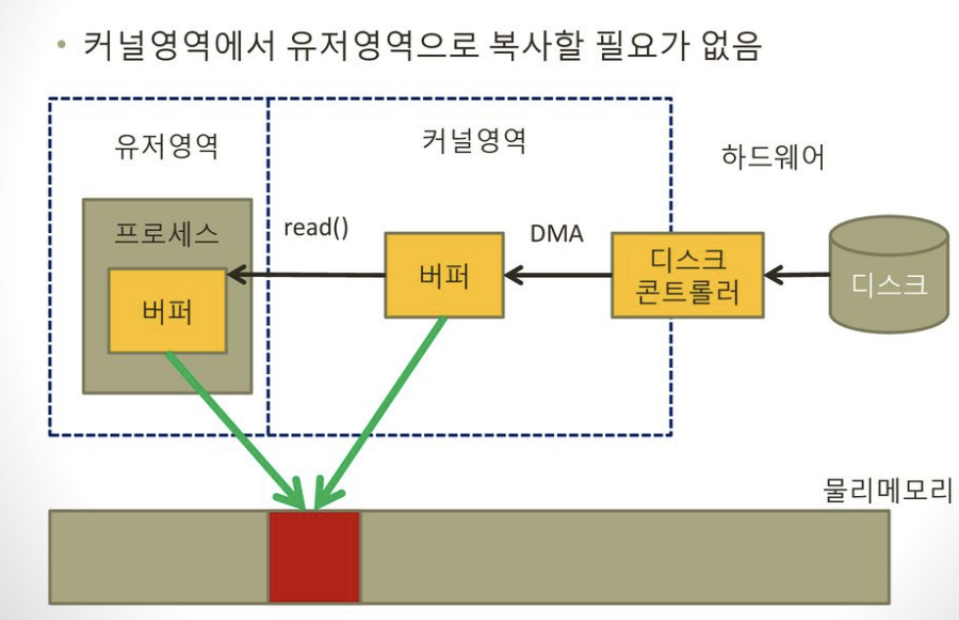
1. Disk controller가 Disk 에서 데이터 read
2. Disk controller가 커널 영역 메모리 버퍼에 데이터 write
3. 커널은 사용자 영역 메모리 버퍼로 데이터를 copy

NIO

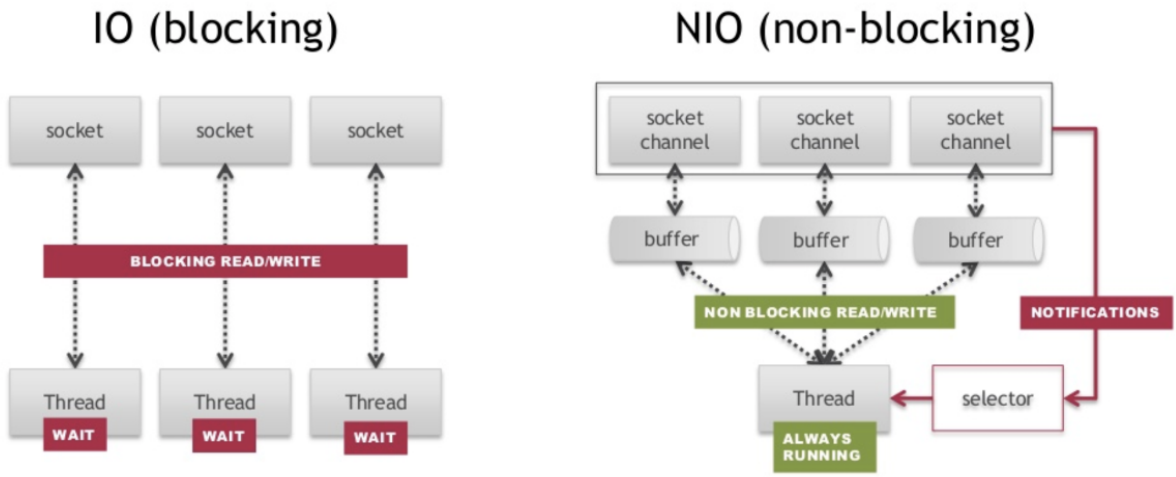
- 읽는 데이터를 무조건 buffer에 저장해서 버퍼내 데이터의 위치를 이동해 필요한 부분만 읽고 쓴다.
- 데이터 IO를 channel을 통해 읽는다.
  - channel : non-blocking read를 할 수 있도록 지원하는 connection
  - 양방향으로 입력과 출력이 가능하다.
  - 채널에서 데이터를 주고 받을 때 사용하는 것이 버퍼이다.
  - ServerSocketChannel과 SocketChannel을 이용하여 TCP 네트워크 프로그램에 이용한다.
- 넌블로킹, 블로킹 둘다 가짐
  - 파일을 읽는 File I/O 는 blocking. 그 외는 non blocking
- selector
  - 여러개의 채널에서 발생하는 이벤트(연결이 생성됨, 데이터가 도착함 등)를 **모니터링할 수 있는 객체**다. 하나의 selector(스레드)에서 여러 채널에 대해 지속적으로 모니터링 한다.
  - 실제 수신되는 데이터가 없음에도 무한루프를 돌며 버퍼에 데이터 여부를 계속 확인하게 되는데 이를 방지하기 위해 사용된다.



- 연결 클라이언트가 많고 IO가 작은 경우
  - NIO는 버퍼할당크기가 문제가 되고, 모든 입출력 작업에 버퍼를 무조건 사용해야 하므로 즉시 처리하는 IO보다 성능 저하가 발생할 수 있다.


• 커널영역에서 유저영역으로 복사할 필요가 없음
유저영역
커널영역
하드웨어
프로세스
버퍼
read()
버퍼
DMA
디스크
콘트롤러
디스크
물리메모리

정리


IO (blocking)
socket
socket
socket
BLOCKING READ/WRITE
Thread
WAIT
Thread
WAIT
Thread
WAIT
NIO (non-blocking)
socket channel
socket channel
socket channel
buffer
buffer
buffer
NON BLOCKING READ/WRITE
NOTIFICATIONS
Thread
ALWAYS RUNNING
selector