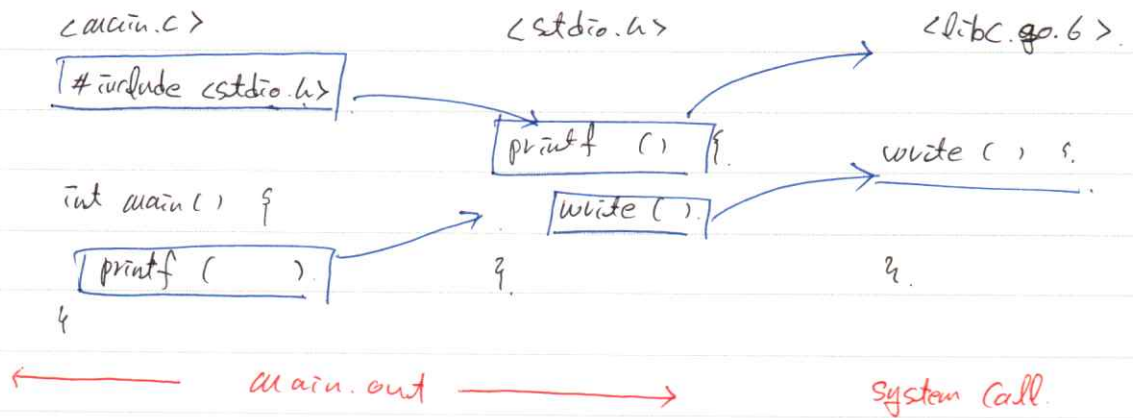


# 컴파일



- System Call <sup>반복</sup> ~~방법~~: 바이너리에 없으면 프로그램마다 중복 코드 ↑.

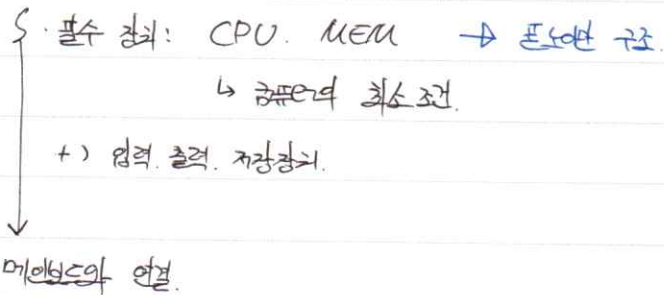
Common 라이브러리 → 중복 코드 줄임!!



정적 링크.

- 정적 링크 내용 ... 정적 링크 포함.
- 메모리 ↓ 성능 ↑.

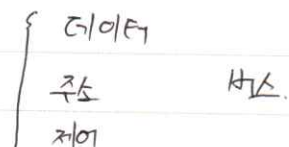
# HW 구성.



# 버스

데이터 전송하는 통로인 버스

- 메인보드에 연결된 장치, 버스 내의 버스 장치 등



# CPU

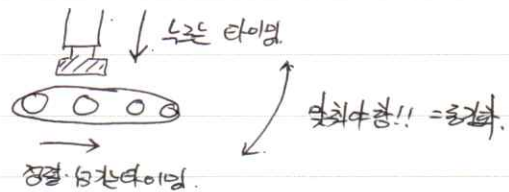
· RAM 바깥을 옮겨 실행 → 저장장치 액세스 x.

· 메모리 = CPU의 유일한 작업 공간.

# 클럭과 버스

· 컴퓨터 클럭 (clock) 단위로 작업 ... Hz

· 동기화 필요함.

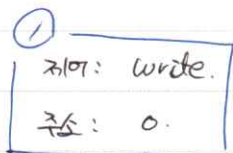


# CPU의 버스 이용, 데이터 R/W.

CPU

MEM.

10, w

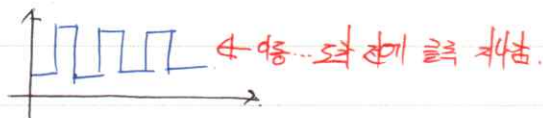


addr	data
...	
10	
...	

한 바이트  
→ 32비트? word에 따라 다름.  
32bit → 4바이트  
64bit → 8바이트

\* transform graph:  $\begin{pmatrix} Page 10, \\ Material Chap 3 \end{pmatrix}$

\* CPU 클럭을 높이면? 전압이 더 높을 필요 ...



# CPU 메모리 병목 현상 해결하기.

물리적 설계.

1. 표준 캐시 ... DDR (Double Data Rate).
2. 듀얼 채널 ... 병렬 처리.
3. CPU와 가까운 거리.

구조적 설계.

4. 바이트.
  - \* 디바이스 입사 저장 공간 = 바이트.
  - ↳ CPU 캐시.
    - 바이트.
    - CPU 내장 ... 코어와 가까운.

+ Page 14, Material.

# 지역성, Locality.

\* 코어에서 ...



(공간 지역성) 배열 ... 연속된 주소의 원소를 많이 참조하는가?

$arr[0], arr[1]$  —

\* 지역성 이용해 처리 예측 성공 → Cache hit.  
실패 Cache miss.

# Random Access

· 임의 주소를 "영" 저장할 때 바로 접근하는 것.

· 주소 목적지를 찾기 위해 코어를 반복적으로 뒤적거리는 것 X.