Memory Model

what is it?

Memory Model

- describes formal specification of how memory system will appear to program.
- reorders operations in program execution
- governs interactions between threads - shared memory

Weak Memory Model

- reorder load/store operations
  - 대신, reordering이 correctness of single thread execution에 영향 주면 안됨!
  - follow true dependency ★ 순서 지켜야 함 ★
    (read-after-write dependency)

true dependency

- RAW (read-after-write)

  1. $R2 \leftarrow R1 + R3$
  2. $R4 \leftarrow R2 + R3$
  3. $R5 \leftarrow R4 + R7$

  write 이후에 read를 해야 correct data 얻을 수 있음.
  - 주로 pipelined execution에서 발생

  structure hazard 해당함

- WAR (write-after-read)

  1. $R2 \leftarrow R1 + R7$
  2. $R7 \leftarrow R1 + R3$

  read 한 이후에 write 해야 correct data 가능
  - 주로 parallelized execution에서 동시에 처리할때 나타남

- WAW (write-after-write)

  1. $R2 \leftarrow R1 + R7$
  2. $R2 \leftarrow R1 + R3$

  ★ 2번 instruction이 최종 결과로 되어야 함
  1번으로 overwrite되면 incorrect하게 됨!

Strong memory model

- acquire and release semantics
  - aquire semantics = read acquire : read-aquire 이후의 reordering 막기
  - release semantics = write release : write-release 이전의 reordering 막기

  그래서 multiple write 있으면
  2 순서는 무조건 semantically no problem
  이어야 함!!!!!

Producer-Consumer Synchronization

Example

· global flag 사용

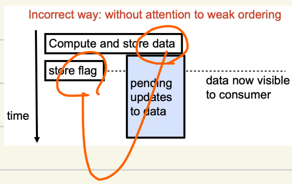
- Producer : it is done with data by setting a flag
  → flag는 update가 consumer에게 visible 해야 set 가능
- Consumer : waits until flag is set before reading data

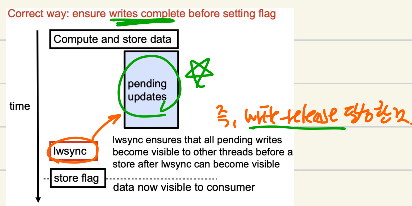
consumer starter flag ob

√ **Producer**

true dependency 거치기

· weak ordering 고려하면, reordering 방지해서 data write 먼저 끝내길 기다려야 함





flag-data는 compiler에겐 관계없는 애들 😡

Consumer

· flag와 data와의 관계는 compiler에겐 무의미해서 reorder 가능.