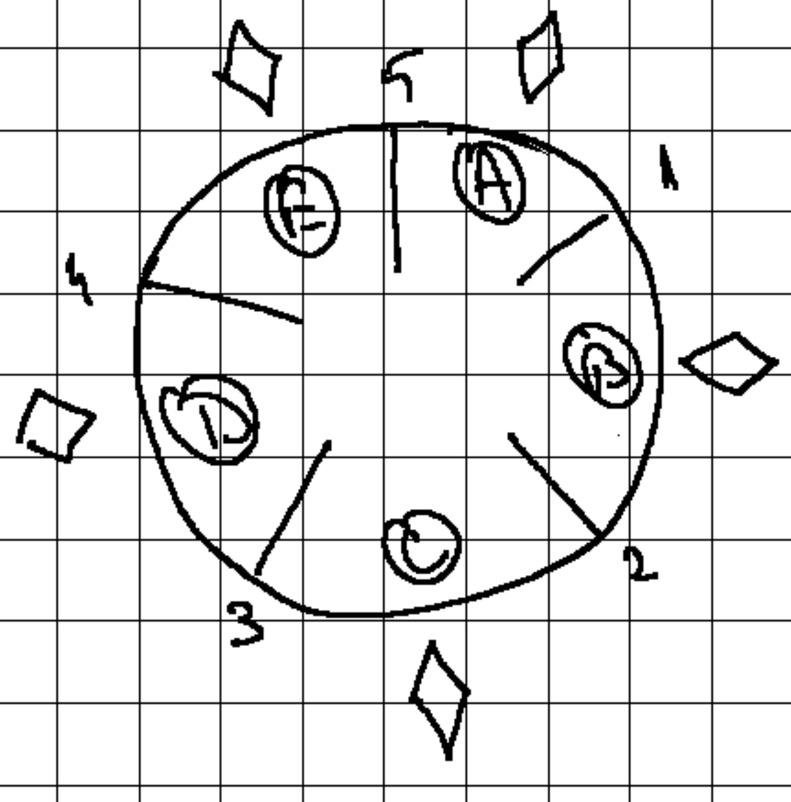


Rule: Always lock resources in the same order

if A, B, C, D, E will pick the sticks (1, 2, 3, 4, 5) with their right hand (or all with the left hand) ⇒ ⇒ deadlock

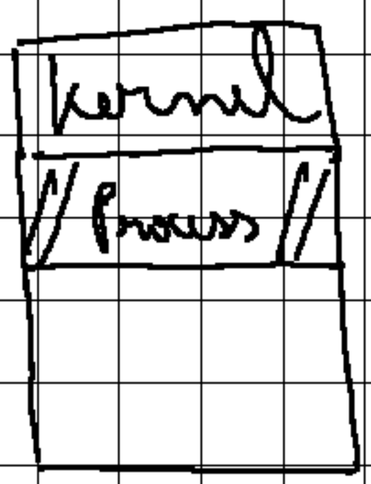
Conditions that make deadlock possible:

1. mutual exclusion
2. lock and wait
3. non-preemption
4. Circular wait

Memory management

- real allocation
  - (A) • single tasking OS
  - • multi tasking OS
    - • fixed partition
      - (B) • absolute
      - (C) • relocatable
    - (D) • variable partition
- virtual allocation
  - (E) • page
  - (F) • segmented
  - (G) • page-segmented

A.



the compiler / linker hard codes physical memory addresses in the executable

Adress calculation:

a.c → a.o ⟶ a.exe ⟶ process

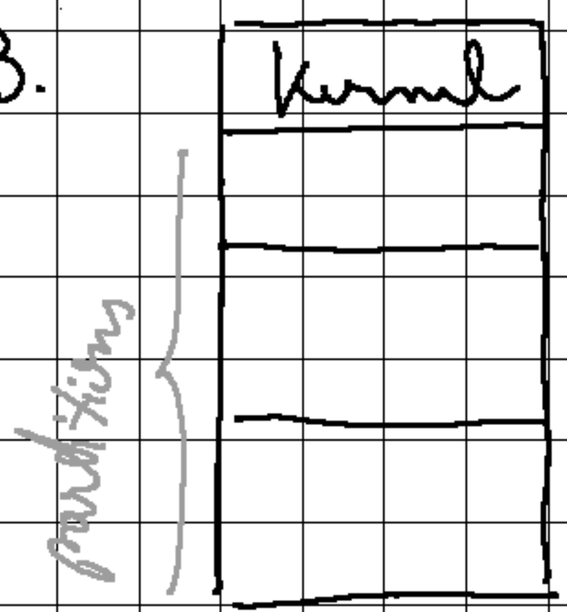
obj file

b.c → b.o ↗ link

exec

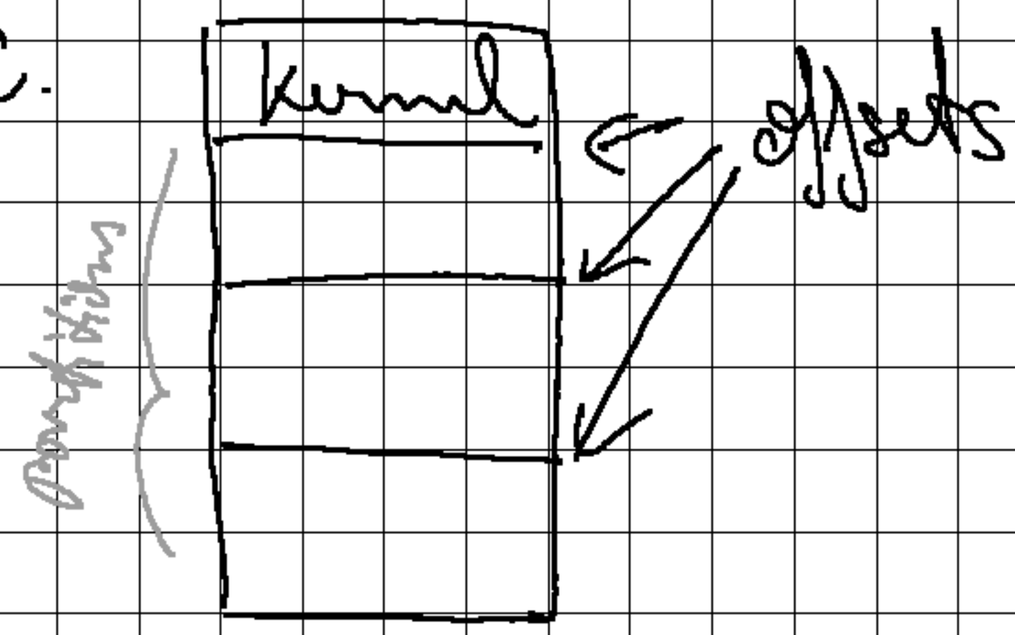
compile

B.



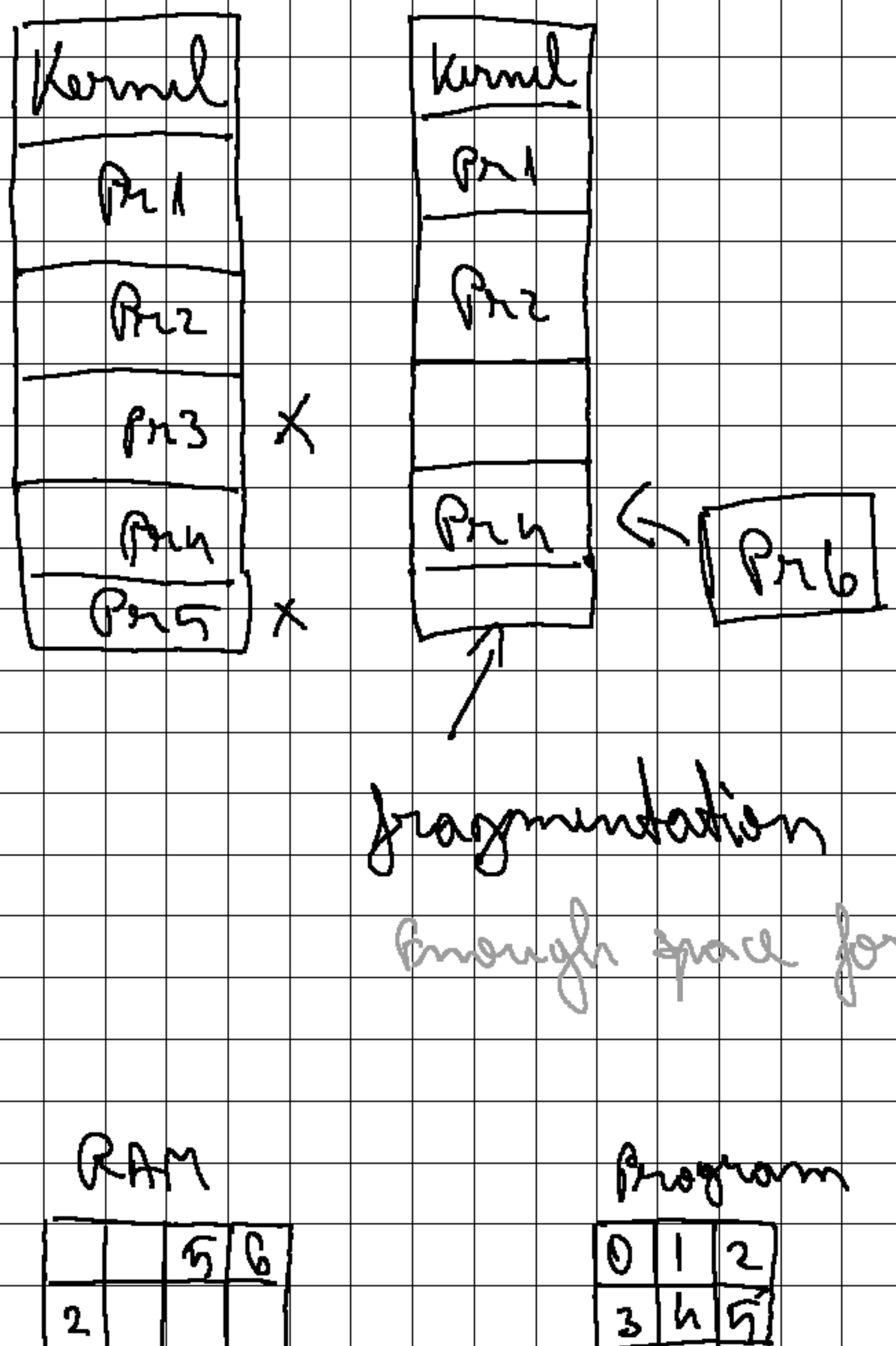
Compile the program for a partition

C.



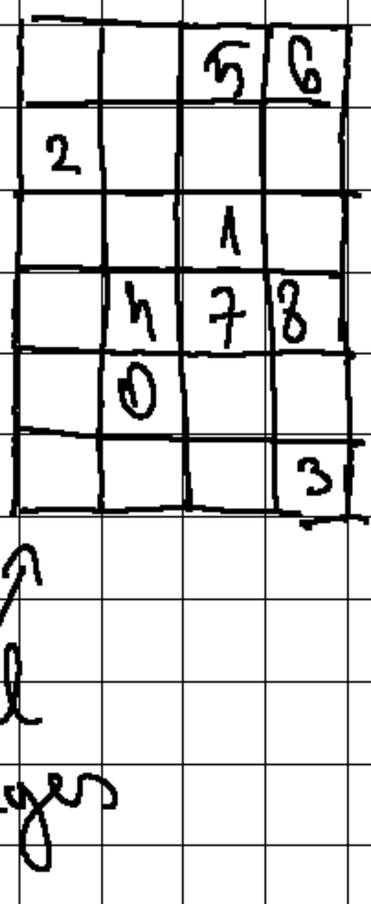
P. Variable partitions

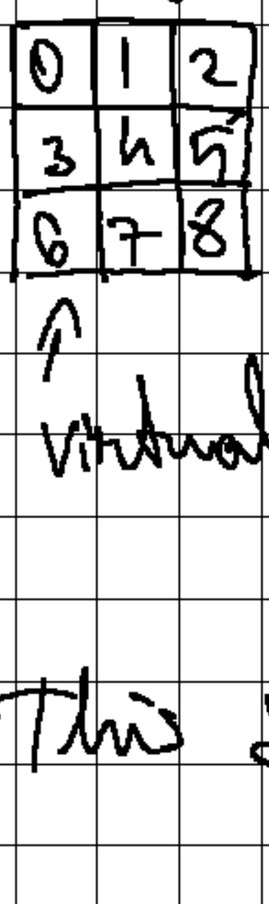
- we define the partition when the process starts



Enough space for process 6, but in 2 diff partitions

F.





pages are of fixed size (4-8kb)

This solves fragmentation

address calculation results in searching which is slow

Loading policies:

- when should pages be loaded?
  - all at process start
    - starts slow
    - faster execution
  - when needed
    - starts slow?
    - slower execution
  - the neighboring page (principle
    - load requested page and a few neighboring pages