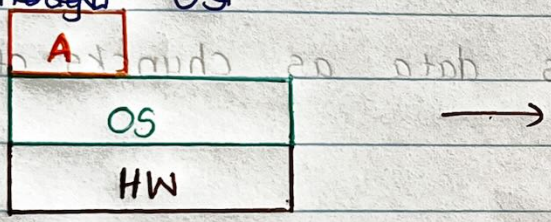
Operating System Concepts

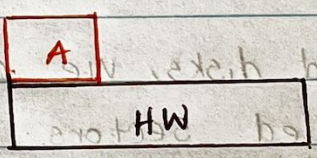
* Open source systems are more secure as more experts can read the code, identify the vulnerabilities & release the patches quickly.

* On the otherhand, closed source systems are less secure as people are legally prohibited to read the source & external parties cannot release any patches due to legal limitations.

* ~~Eventhough~~ ~~OS~~



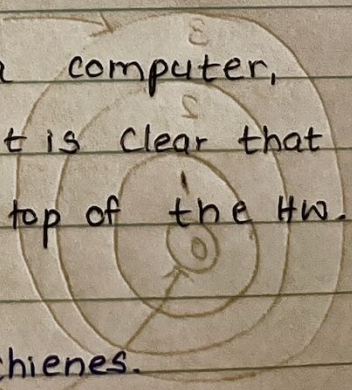
✗ Not correct.



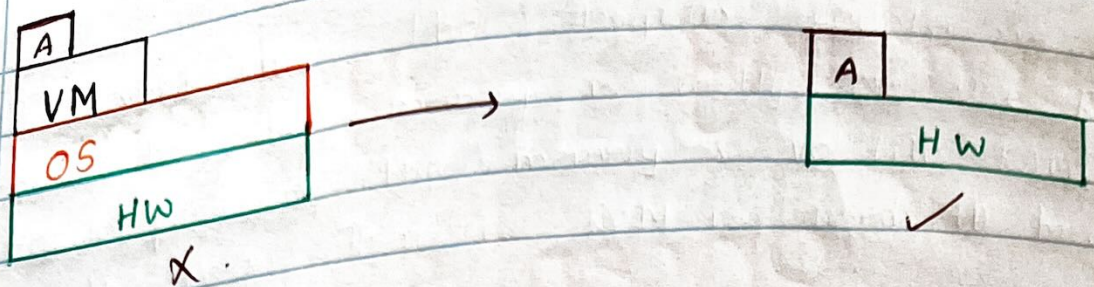
Correct interpretation.

* When you run any program, it runs directly on H/W & OS is not associated with it.
  - Consider a computer with
    1 processer
    1 program counter
    1 instructions register.

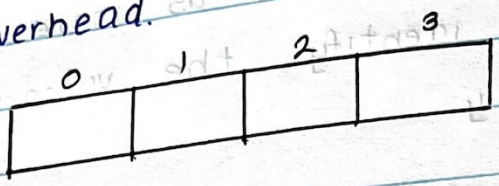
  If we run a program in such a computer, where does the OS run? So, it is clear that the program runs directly on top of the HW.



* The same applies for virtual machienes.



* We need our app to run. Not the OS. Therefore, it is a overhead.



* Usually the memory is byte addressable. It can never access individual bits.

Can't access 1024th bit

Can access 1024th byte

Hard Disk

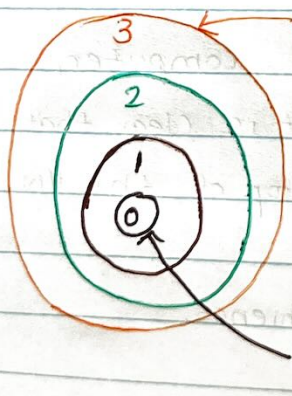
* In hard disks, we address data as chuncks of bytes which are called sectors.

512 bytes
512 bytes
512 bytes
⋮

1 sector → 512 bytes.

x86 Intel Hardware

* This is simple & old hardware technology.



- Intel processor runs in different ring levels.

- OS typically use Ring 0 for the kernal & Ring 3 for user applications, while rings 1 and 2

are generally unused today.

Ring 0 (Highest) - Used for OS kernal. It has full aocess for all HW & resources & can execute all privileged instructions.

Ring 1 & 2 (Intermediate) - These rings offer more protection than ring 3, less than ring 0. Once used for components like device drivers.

Ring 3 (Lowest) - Also known as "user mode". It is where most user applicati run with restricted access to hardware & system resources.

* When you boot the machiene, the first instruction goes to ring level 0.

Memory

* When the memory is just turned on, (at the booting) th memory contains a bunch of random values.

* Since Program Counter (PC) is also a register, it also contains random values, ~~we~~ if we don't change it.

* Then the Information Register fetches some random instruction & executes it. Then after booting, the Computer will execute some random instruction,

* To avoid this, the Program Counter is hard coded. Since hardcoding memory is costly, we use ROM in small part of the memory & volatile RAM is used for the rest. Therefore, ROM is used to hardcode.

For the processor both RAM & ROM are the same.