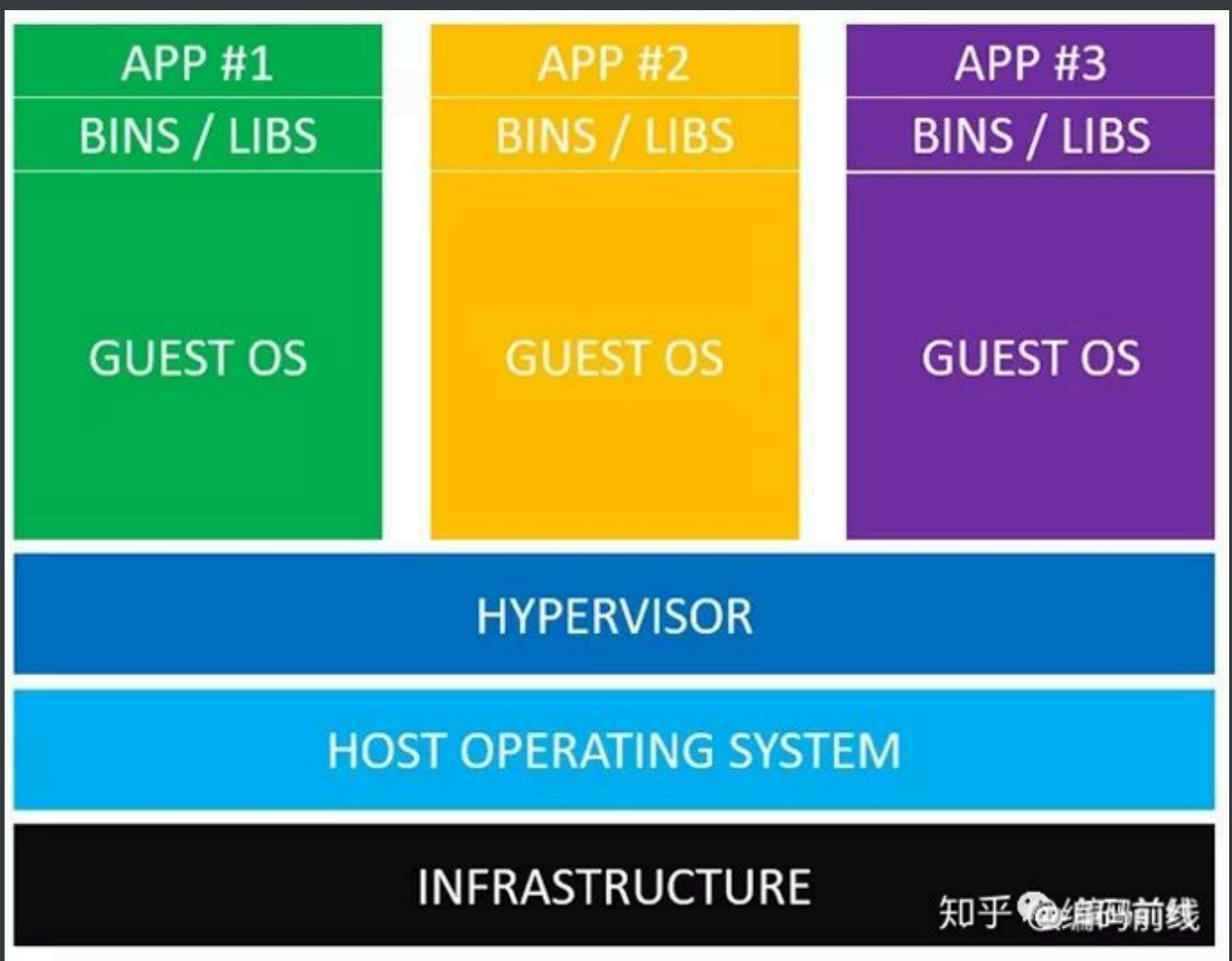


# Operating System HW3

- a. VM is the software to simulate the hardware in order to isolate the different system or different environment or protect the original system in same physical machine  
A VM are many guest oses(Ring 1) execute on the hypervisor(VMM), which is used to forward the system call from guest to host(Ring 0). User can execute program(Ring 3) on the guest host with isolated kernel, file system, even network environment.



b. The advantages of virtual machine is that we can protect the host os and run a different kernel in the same machine.

The disadvantages of virtual machine is that we have some cost on performance because we have to forward the system call(because of the

different behavior of instruction in ring 0 and other spaces on x86) to the host os and the virtual memory(TLB and other mechanism on it)

have to be forward, too.

The advantages of VMware is that compare to other full-virtualization solution, VMware has a better graphic driver and is more stable than other(Virtualbox ...etc)

The disadvantages of VMware is that it cost a lot of money.

2. a. Layered approach means system is broken up into several module and each module can't see other's inner implementation.

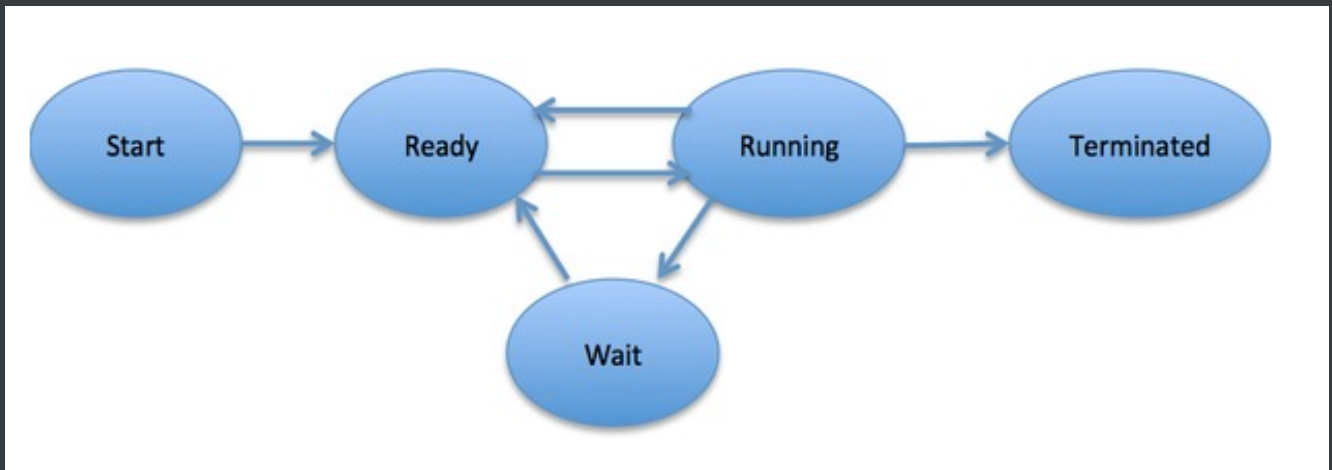
It's lower the development cost of os(because we don't have to change other module due to encapsulation) and make the system become more robust.

b. Modular kernel means each functionality of kernel is divided into many modules. It makes programmer easy to add or modify module to extend the system.

c. Both approach divide the large program into small and encapsulated part and every different part(module, layer) can only communicate by using public and narrow interfaces.

3. a.

1. Start->Ready: System finishing create the necessary resource for the process(initialize task\_struct(PCB) and associate TSS and LDT with GDT... etc) and load program into memory and add process into scheduler's queue.
2. Ready->Running: Scheduler tell this process to run. In other word, it make context switch(based on TSS or hardware) and the PC and other register are in the state of running this process.
3. Running->Wait: Process are stuck by some operation(e.g i/o busy) and need to wait.
4. Wait->Ready: Process is kicked by the scheduler of operating system and need to wait next turn to execute.
5. Running->Ready: Process is kicked by the scheduler of operating system and need to wait next turn to execute.
6. Running->Terminated: Process is stopped by interrupt(e.g. abort) or return directly.



b.

1. Short-term scheduling: Decide which process needs to execute in the ready queue
2. Mid-term scheduling: Decide which process is not execute for a while and swap it out from memory
3. Long-term scheduling: Decide which process needs to be put in to ready queue(system memory)

c.

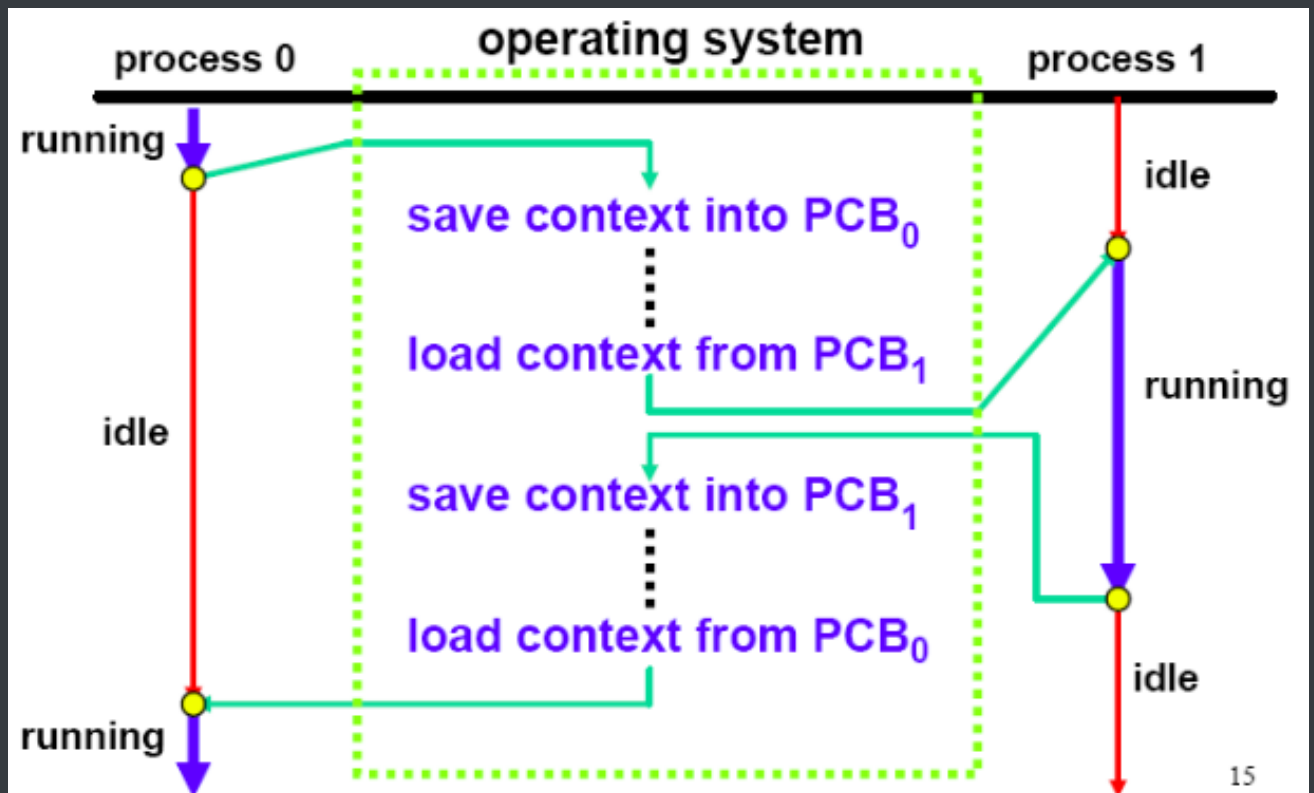
1. Shared memory: using a public memory space which is hold by both processes want to communicate and just direct write and read the memory space.
2. Message passing: Process communicate by kernel, the process want to send the message will be sender. In contract, the other process is receiver. Sender sends messages to OS, and OS send messages to Receiver.

4. ac.

Save context into PCB0: System stores previous process's status(pid, virtual memory mapping status, thread status) to PCB.

Load context from PCB1: System load next process's status and jump to the next process(usually use stack and return to achieve long jump)

Save context into PCB1 and load context from PCB0 is same as the privious step but do reversely and with different process.



b.

Timeout: Take Free RTOS for example: when program running a while, it will trigger SysTick interrupt, OS and CPU will enter Ring 0, and add a counter to determine if the process is out of time. If it is, it will trigger PendSV exception, and do context switch by it.

I/O syscall: Instead of context switch between process like timeout above, I/O interrupt is a CPU context switch. Because it still in the same process. The and OS enter ring 0 without storing user space information(Virtual Memory status... etc) and invoke I/O interrupt.