OS Structure

Monolithic System:

* single program in kernel mode.
* Linked together into a single large executable binary program.
* Basic structure:
  + Main program: requested service procedure.
  + A set of service procedures: carry the **system calls**
  + A set of utility procedures: help service procedure, do things needed by service procedures.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Simple and fast | * running a single program * run one single large executable binary program, if there’s a crash in any of procedures will take down the entire OS) * do not protection (free to call any other, can uses low level language, cannot manage user environments by using procedure) |
|  |  |

Layered System:

* OS broken into a number of layers.
* The bottom layer is hardware, and the highest layer is the UI. A layer can use serveral functions and services of lower-layer (not all).
* Lower layers (or kernel) contains the most fundamental functions for managing system resources.
* System has 6 layers:

|  |  |
| --- | --- |
| Layer 5 (toppest) | The operator |
| Layer 4 | User programs |
| Layer 3 | IO management |
| Layer 2 | Operator-process communication |
| Layer 1 | Memory and drum management |
| Layer 0 | Processor alloc and multi-programming |

Advantages:

* Easy to maintain, support, manage errors
* Modularity, encapsulation (đóng gói)

Disavantages:

* Slow

Microkernel (chia để trị)

* Microkernel will split system into many small pieces. If there’s a bug in a module, it will not bring down all the system.
* Remove all nonessential components from the kernel and implement them on user mode.
* Running independent process in every component 🡪 resist crashing.
* Run user mode 🡪 protection against bugs.
* Put mechanism inside kernel without policy.
* Communication facility using message passing between the client program and various series in user mode.

Advantages:

* Easy to extend and port the OS to new architectures.
* Advance security and reliability. (less code running in kernel)

Disadvantages:

* How we know that which one is nonessential?
* Performance overhead of user space to kernel.

Client-Server Model

* Server provides services for client through request.
* Client uses the server’s service.
* Message passing is used to communication.
* Run on different or same computers.

VM

* Not an extended machine
* is identical to the true hardware.
* Run any OS.
* Some software: VMware, virtual pc, virtual box, JVM (java vm), etc.

Advantages: Completely protecting system resources, system compatibility, do not disrupt the normal system operation.

Disadvantages: can not allocate all disk to VM.

* VM monitor: a heart of the system, runs on the bare hardware and does the multiprogramming. Another name: virtualization layer or hypervisor.

Exokernel

An additional component in VM, protect the resources in VM.

Exokernels make the VM thinking that it has its own disk.

Advantages:

* Save a layer of mapping
* Keep track of which VM has been assigned which resource
* Separate the multiprogramming in user mode.
* Less overhead.

**An actually, it’s a program.**