Design principles:

Separation of mechanisms and policies: implement flexible mechanisms to support policies. Eg: LRU, LFU, random.

User kernel protection boundary:

User level:

* Applications
* Threads
* un-privileged mode

Kernel level:

* memory management
* CPU management
* privileged mode
* hardware access

trap instructions are those mistakes made by user in application inputs.

System call:

(user mode) User process execution --- > (user mode) calls system call using system number --- > (kernel mode) executes the system call --- > (user mode) gives the output

Synchronous calls are made when they wait after they call the kernel mode.

* Monolithic OS consumes more memory with interactions directly from applications to the hardware devices.
* Modular OS requires less resource needs as it has an intermediate layer that communicates between the different layers. It manages the traffic of instructions but has less performance and high maintenance.
* Micro kernel uses OS as an interface which uses addresses and threads for inter process communications. But it disadvantages with portability and cost of kernel is high which is more complex.