* Kernel architectures:

Depending upon the size & functionality inclusion in the Kernel & the orgn of different components within the OS, there are 3 broad approaches for designing an OS:

1) Monolithic System Structure

- In monolithic system structure there is no specific structure. The OS is written as collection of procedures each of which can call any of other procedures whenever it need to.
- When this technique is used each procedures in the system has a well defined interface in term of parameters & results & each one is free to call any other one if the latter provides some useful computation that the former needs.
- To construct the actual object prog of the OS one compiles all the individual procedures & binds them all together in terms of information hiding. There is essentially none - every procedure is visible to very other one. ~~In mono~~
- In monolithic system there is possible to have some structure. The services (system calls) provided by OS are requested by putting in well defined places such as in registers or on stack & executing a special trap instn known as a kernel call or supervisor call

: The Kernel switches the machine from user mode to Kernel mode & transfers controls to OS.

* Layered System Architecture:

In this approaches, the OS is organized as the hierarchy of layers. Each one constructed upon the one below it.

* Level 0: processors allocation & multiprogramming. It deals with allocation of processor, switches both process when interrupt occur or timer's expired, layer 0 provide the basic multi progn of CPU.

Interrupt → CPU (multi prog)

* layer 1: does the memory mgnt. It allocates space for processes → MM, processor in MM & one office on the drum used for holding parts of processor for which there was no room in MM.

* layer 2: handled the comm^n for each process & operator console.

* layer 3: took care of managing of I/O devices & buffering the info streams to & from them.

* layer 4: was where the user programs were found. They did not have to worry about processes, memory, console or I/O mgnt.

* layer 5: where system operator process was located.

Microkernel OS:

The basic concept of a Microkernel OS is very easy to understand. In this approach, the kernel provides on the most essential OS functn like process mgnt, comm^n primitives & low level memory management. System program or user level programs implemented outside the kernel provide the remaining OS services. These programs are known as servers. As a result, the size of kernel reduces dramatically, making it a Microkernel.

- If there is some probl^n with a particular server, then the service can be reconfigured & restarted without having to restart the entire OS.