OPERATING SYSTEM STRUCTURE - I

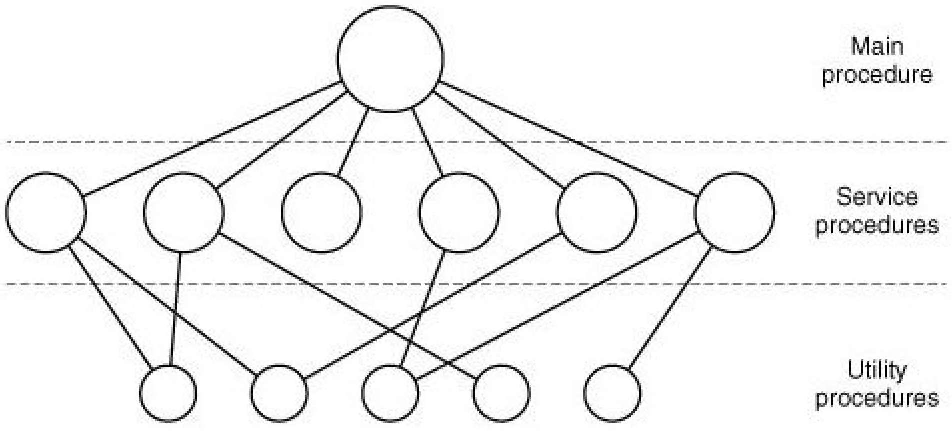
Operating System Structure:

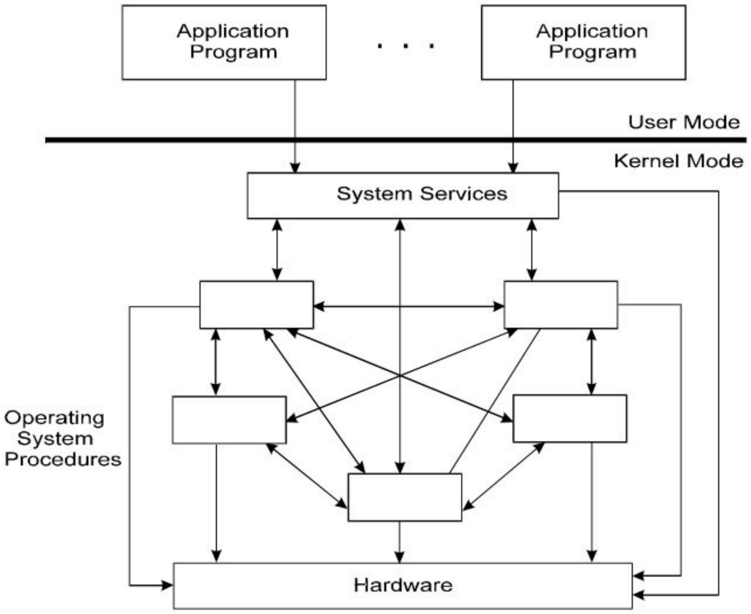
The structure of an operating system is dictated by the model employed in building them. An operating system model is a broad framework that unifies the many features and services the operating system provides and tasks it performs. Operating systems are broadly classified into following categories, based on the their structuring mechanism as follows:

a. Monolithic System
b. Layered System
c. Virtual Machine
d. Exokernels
e. Client-Server Model

Monolithic System

The components of monolithic operating system are organized haphazardly and any module can call any other module without any reservation. Similar to the other operating systems, applications in monolithic OS are separated from the operating system itself. That is, the operating system code runs in a privileged processor mode (referred to as kernel mode), with access to system data and to the hardware; applications run in a non-privileged processor mode (called the user mode), with a limited set of interfaces available and with limited access to system data. The monolithic operating system structure with separate user and kernel processor mode is shown in Figure.





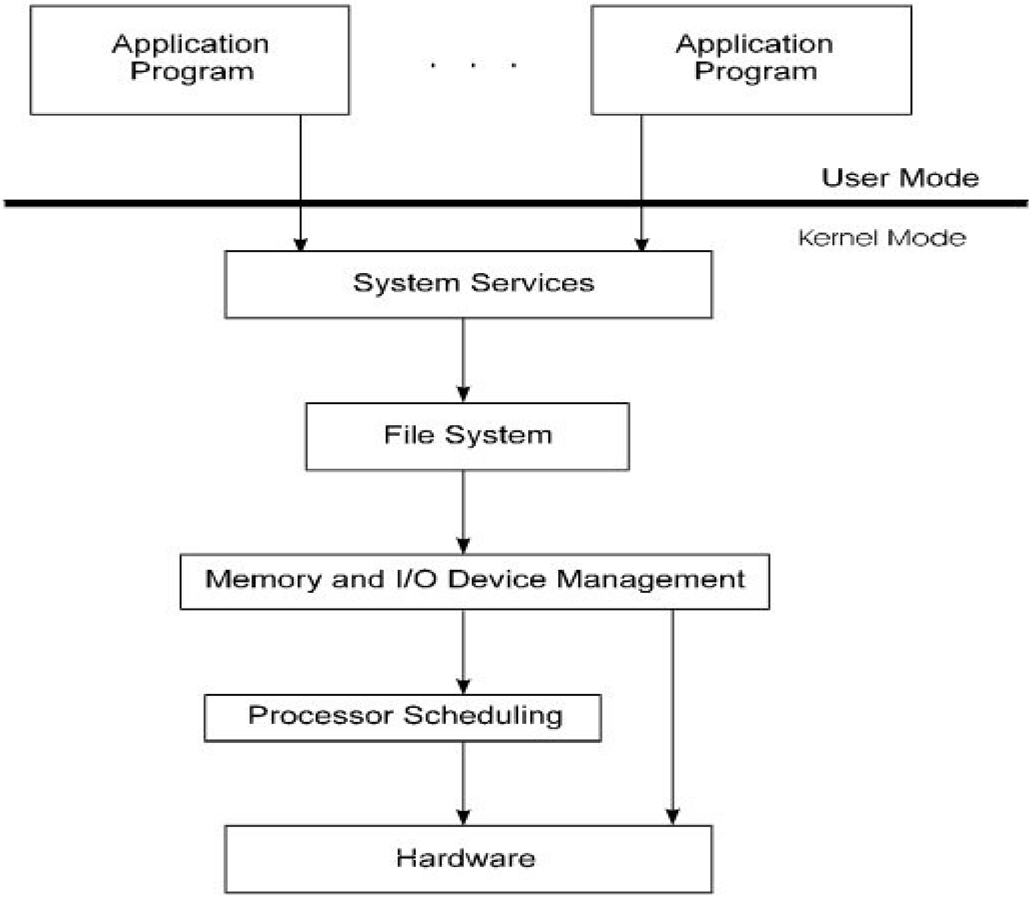
This approach might well be subtitled "The Big Mess." The structure is that there is no structure. The operating system is written as a collection of procedures, each of which can call any of the other ones whenever it needs to. When this technique is used, each procedure in the system has a well-defined interface in terms of parameters and results, and each one is free to call any other one, if the latter provides some useful computation that the former needs.

Example Systems: CP/M and MS-DOS

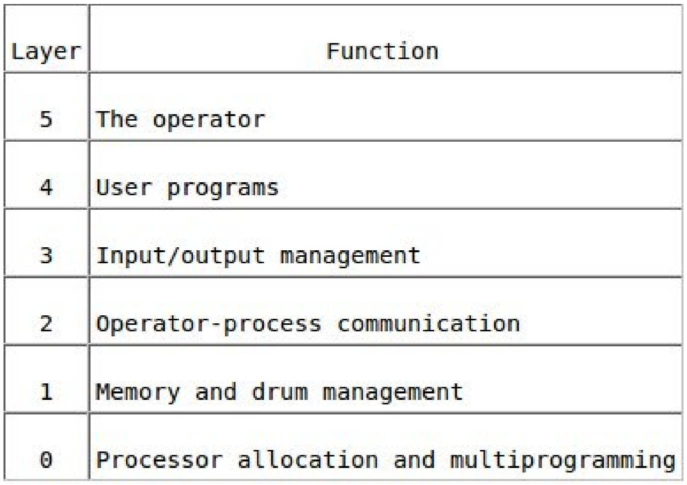
Layered Operating System

The layered approach consists of breaking the operating system into the number of layers(level), each built on the top of lower layers. The bottom layer (layer 0) is the hardware layer; the highest layer is the user interface.

The main advantages of the layered approach is modularity. The layers are selected such that each uses functions (operations) and services of only lower-level layers. This approach simplifies debugging and system verifications. That is in this approach, the Nth layer can access services provided by the (N-1)th layer and provide services to the (N+1)th layer. This structure also allows the operating system to be debugged starting at the lowest layer, adding one layer at a time until the whole system works correctly. Layering also makes it easier to enhance the operating system; one entire layer can be replaced without affecting other parts of the system.



The layer approach to design was first used in the THE operating system at the Technische Hogeschool Eindhoven. The THE system was defined in the six layers , as shown in the fig below.



Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Example Systems: VAX/VMS, Multics, UNIX

Source: http://dayaramb.files.wordpress.com/2012/02/operating-system-pu.pdf