OSL Assignment

Types of OS.

(1) **MICROKERNEL**   (IPC ⇒ Inter-Process Communication)
Block diagram :-   (UNIX ⇒ UNIplexed Information Computing System)

Application IPC	UNIX Server	Device Driver	File Server	} User mode

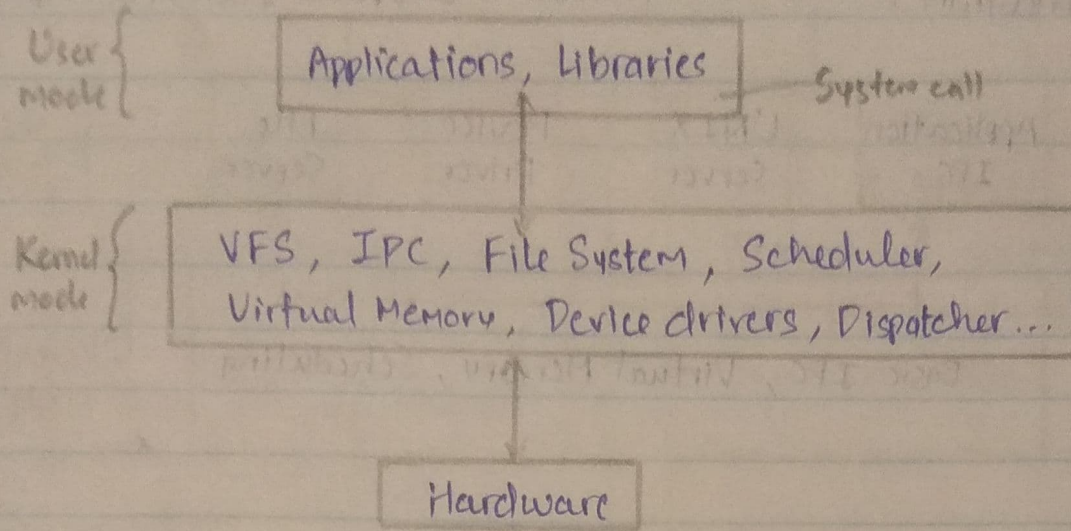| Basic IPC, Virtual Memory, Scheduling	} Kernel mode

Hardware
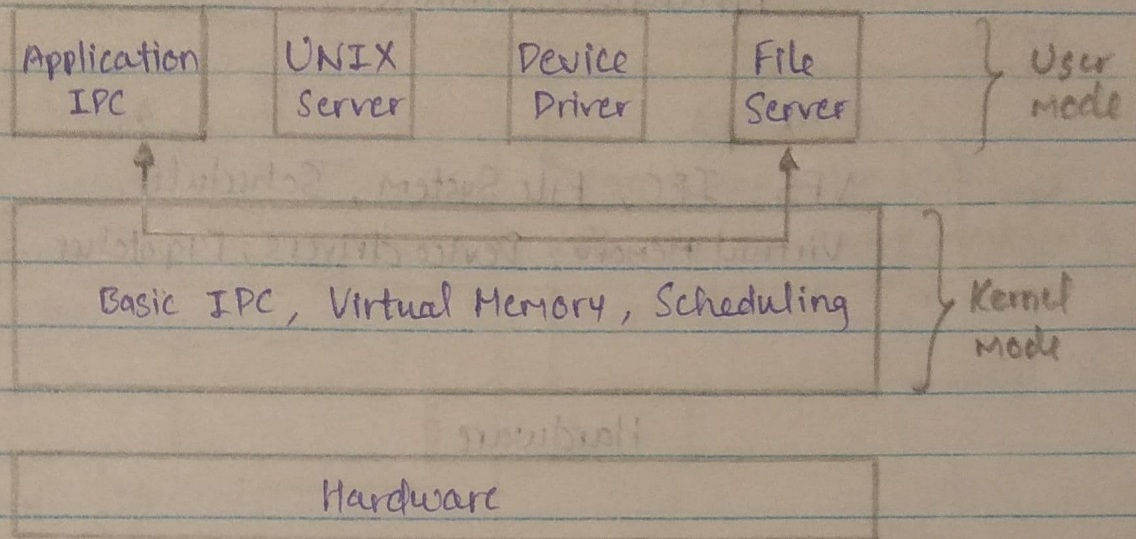
- A microkernel is a minimum-requirement software that can provide the mechanisms needed to correctly ~~oper~~ implement an Operating system
- It makes sure that the code can be easily managed because the services are divided in the user space, thus resulting in increased security & stability
- Eg.- Eclipse IDE
- The minimum functionalities included in Microkernel are:
  (i) Memory Management mechanisms like address spaces etc. It also includes memory protection features
  (ii) Processor scheduling mechanisms that contain process & thread schedulers
  (iii) Interprocess communication: needed to manage the servers that run their own address spaces

② MONOLITHIC ARCHITECTURE

Block diagram :

User mode { | Applications, Libraries | —— System call

Kernel mode { | VFS, IPC, File System, Scheduler, Virtual Memory, Device drivers, Dispatcher... |

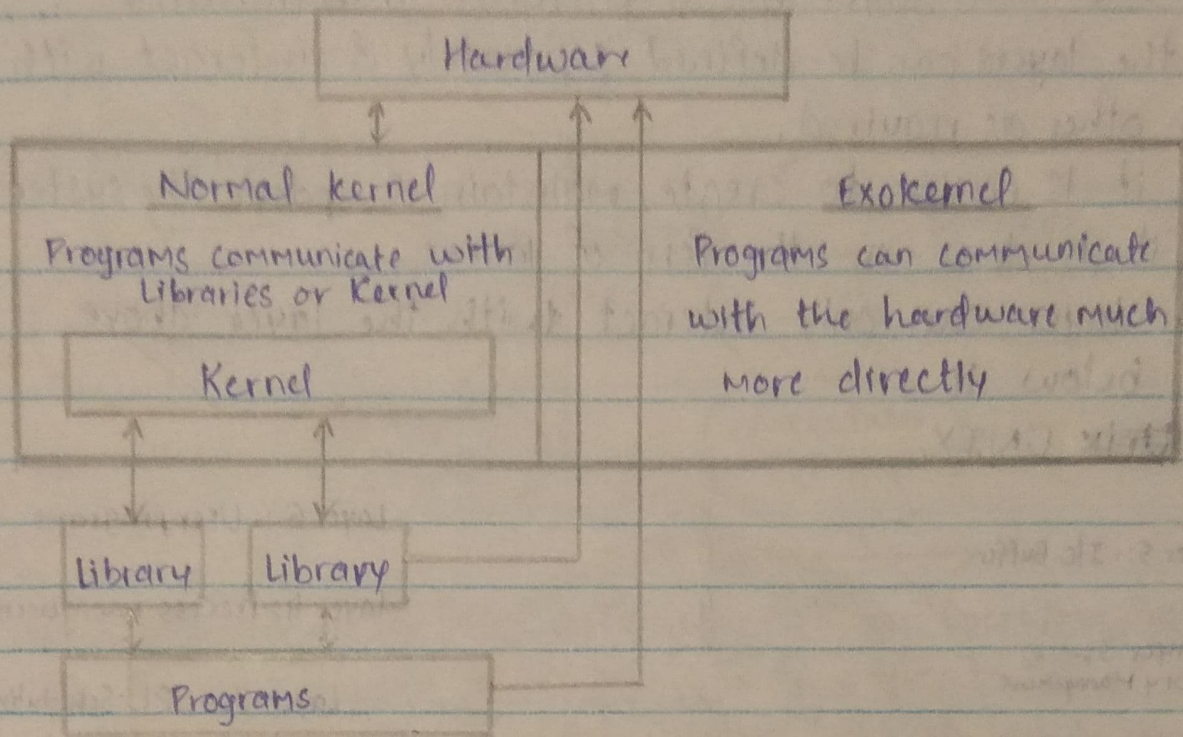| Hardware |

- In monolithic system, the entire operating system works in Kernal space
- This increases the size of the kernel as well as the operating system
- The execution of operating system is quite fast as the services are implemented under the same address space
- The monolithic kernel is not easy to extend
- Eg. - Linux, BSD etc.
- The monolithic kernel is a single static binary file
- If any service fails, It leads to the failure of the entire system
- To add any new service, the entire operating system needs to be modified by the user
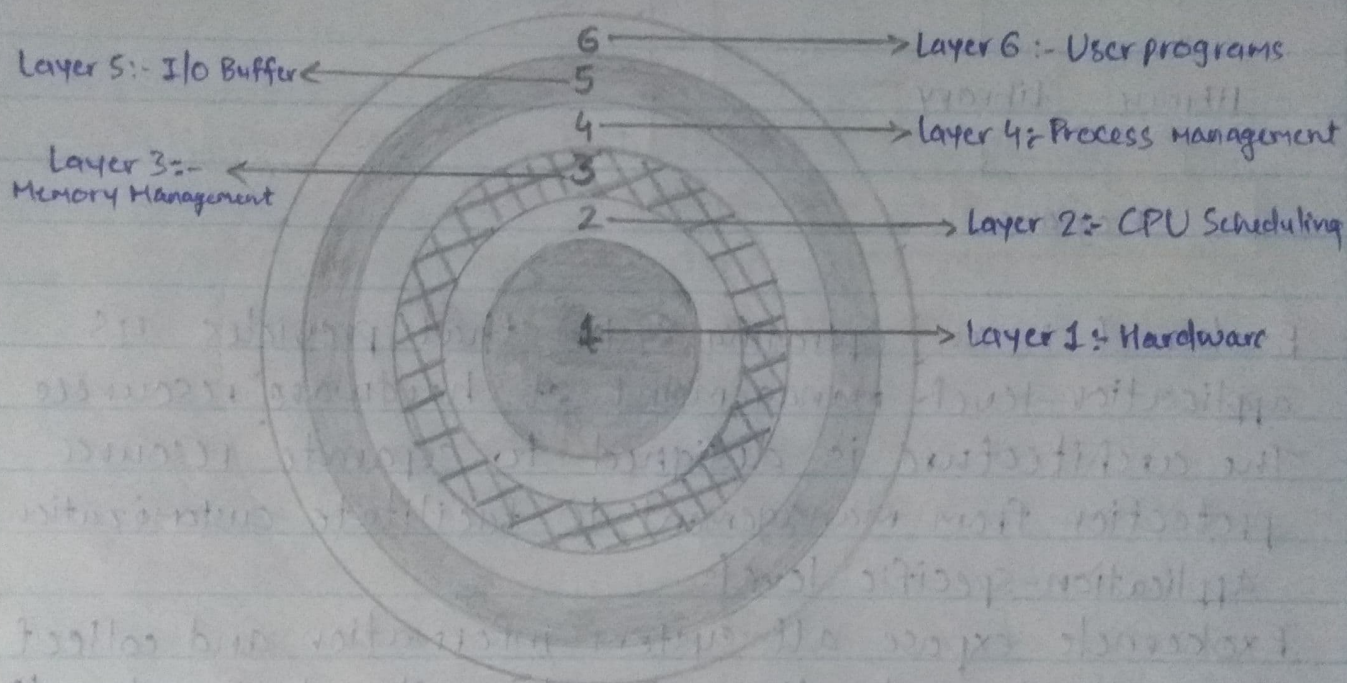
③ EXOKERNEL

Block diagram :-

```
                    ┌─────────────────────┐
                    │      Hardware       │
                    └─────────────────────┘
                       ↕        ↑    ↑
  ┌──────────────────────────┬──────────────────────┐
  │      Normal kernel       │      Exokernel        │
  │                          │                       │
  │  Programs communicate with│  Programs can communicate│
  │    Libraries or Kernel    │   with the hardware much │
  │  ┌────────────────────┐  │     more directly        │
  │  │       Kernel       │  │                          │
  │  └────────────────────┘  │                          │
  └──────────────────────────┴──────────────────────┘
       ↕          ↕
  ┌─────────┐ ┌─────────┐
  │ Library │ │ Library │
  └─────────┘ └─────────┘
       ↕
  ┌──────────────────────┐
  │      Programs        │
  └──────────────────────┘
```

- Exokernel is an Operating System that provides 4S application-level management of hardware resources
- The architecture is designed to separate resource protection from management to facilitate customization at application-specific level.
- Exokernels expose all system information and collect data that applications cannot easily derive locally
- It lets applications choose which instance of a resource to give up
- Resource management is restricted to functions necessary for protection
- Design of exokernel interfaces is quite complex and there is less consistency
- In exokernel, it is easy to develop & test new OS ideas
- Applications can make more efficient & intelligent use of hardware resources by being aware of resource availibility, revocation & allocation

④ **LAYERED OS ARCHITECTURE**

- The operating system is split into various layers in the layered OS and each layer has a different functionality
- All the layers can be defined separately & interact with each other as required.
- Also, it is easier to create, maintain & update the system if it is done in the form of layers
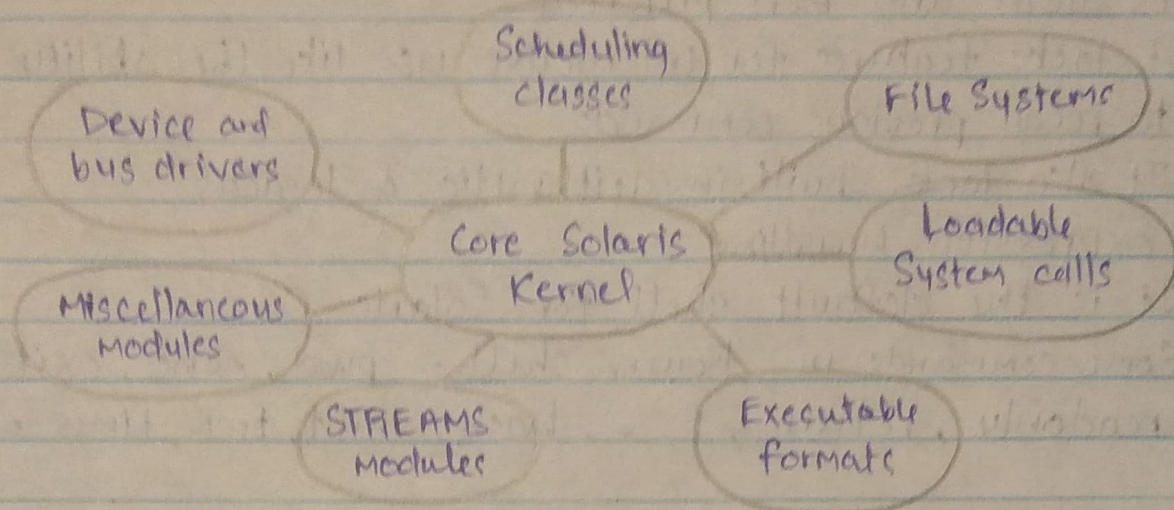- Each layer can only interact with the layer above & below it
- Eg.- ~~Unix~~ UNIX

Layer 5:- I/O Buffer ←   6 ———→ Layer 6 :- User programs
                                          5
                                          4 ———→ Layer 4:- Process Management
Layer 3:- ←   3
Memory Management   2 ———→ Layer 2:- CPU Scheduling

                                          1 ———→ Layer 1:- Hardware

Layers :-

1) Hardware layer :- Interacts with system hardware & coordinates with all peripheral devices
2) CPU Scheduling :- Scheduling processes for the CPU
3) Memory Management :- Deals with memory & moving of processes from disk to primary memory for execution & back again
4) Process Management :- Assigning the processor to a process at a time
5) I/O Buffer :- Means of interacting with the system
6) User programs :- deals with applications like Browser, Games etc.

MODULAR O.S.

Block diagram (Solaris OS) :-

```
              ┌─────────────┐
              │  Scheduling │
              │   classes   │
              └─────────────┘
┌─────────────┐                    ┌─────────────┐
│  Device and │                    │ File Systems│
│ bus drivers │                    └─────────────┘
└─────────────┘
              ┌─────────────┐      ┌─────────────┐
              │ Core Solaris│      │  Loadable   │
              │   Kernel    │      │ System calls│
┌─────────────┐└─────────────┘      └─────────────┘
│Miscellaneous│
│   Modules   │
└─────────────┘
    ┌─────────────┐      ┌─────────────┐
    │  STREAMS    │      │ Executable  │
    │  Modules    │      │  formats    │
    └─────────────┘      └─────────────┘
```

- It is considered the best approach for an Operating System
- The kernel has only a set of core components & other services are added as dynamically loadable modules to the kernel either during runtime or boot time
- It resembles layer structure due to the fact that each kernel has defined & protected interfaces but it is more flexible than the layered structure as one module can call any other module

SHELL
- The shell is the outermost layer of the operating system. It incorporates a programming language to control processes and files, as well as to start & control other programs
- The different types of shells are
(i) The C shell (sh)
  - It incorporates features such as aliases & command history
  - It includes helpful programming features like built-in arithmetic & C-like expression syntax
  - Some commands in c-shell :- alias, bg, break, breaksh, case, cd, chdir etc.

(ii) The Bourne shell (sh)
- It is the original UNIX shell
- It's faster and more preferred
- It lacks features for interactive use like the ability to recall previous commands
- It also lacks 'built-in arithmetic & logical' expression-handling
- It is the default shell for Solaris OS
- Some commands in Bourne shell :- pwd, echo, exit, readonly, return, :, ., hash, shift, test, times etc.

(iii) The Korn shell (ksh)
- It is a derivative of the Bourne shell, so it supports everything from the Bourne shell
- It has interactive features along with built-in arithmetic & C-like arrays, functions & string manipulation facilities
- It is faster than C-shell and is compatible with scripts written for C-shell
- Some commands of Korn shell :- :, .file, break, break n, cd dir, echo args, eval cmd, exit, exit n, etc.

(iv) *GNU Bourne Again shell (bash)
- It's compatible with the Bourne shell
- It includes features from Korn & Bourne shell
- Some commands in bash :- ls, echo, touch, mkdir, grep, man, pwd etc.