.

VEDA Solutions

inspiring linux learning

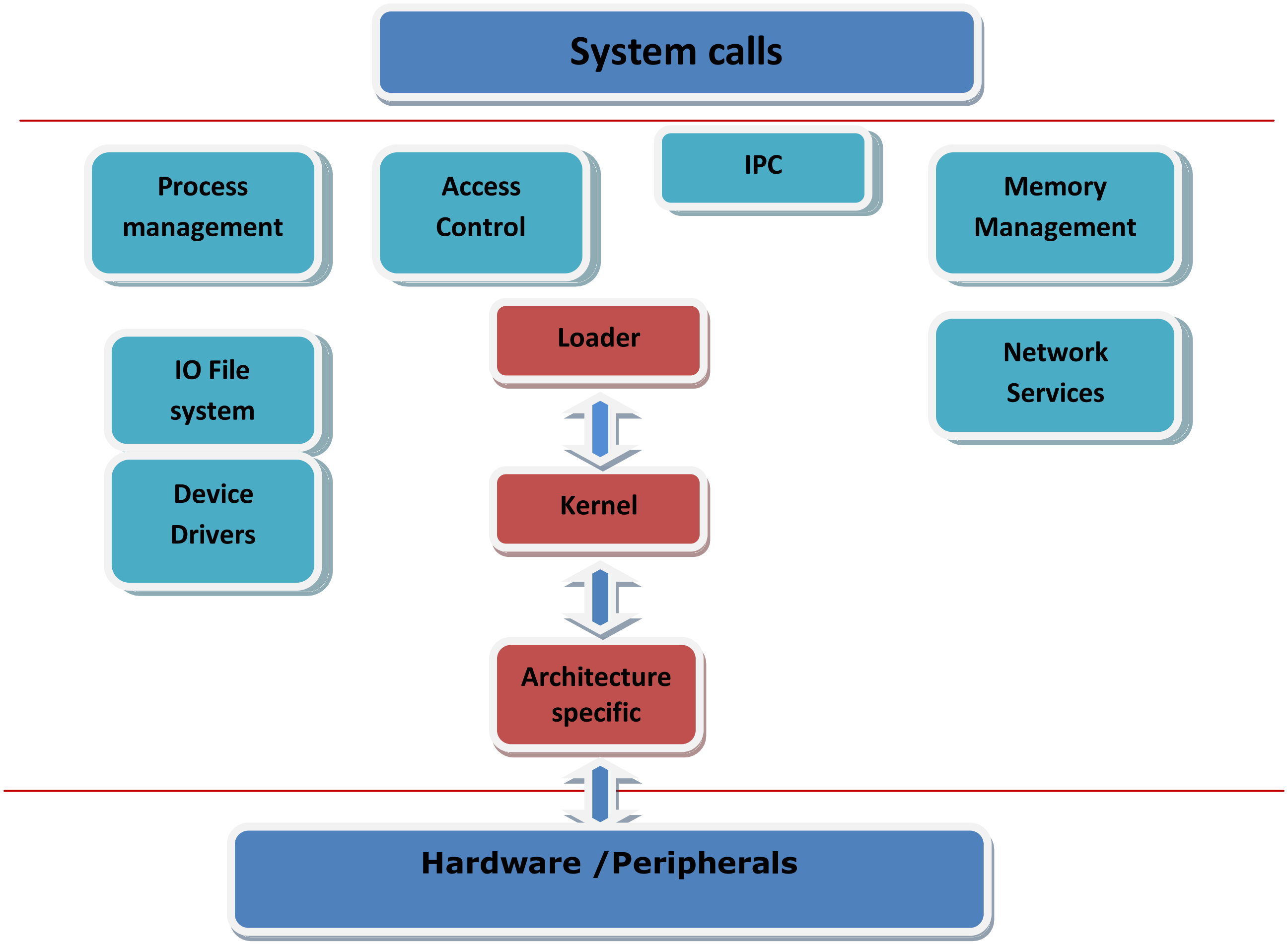
OS Designs –Class Notes

Operating System Designs

Operating system serves as a platform to run all kinds of applications. Basically OS is integration of two modules.

a) User Interface
b) Kernel

Generic Kernel



Kernel is a collection of various subsystems that provides services to an application.

- Loader, Kernel and Architecture specific together called as **Core Subsystem.** They are responsible for system initialization and resource management.

- Remaining all are called **service subsystems**.

Architecture specific (HAL / BSP): This layer is responsible for platform initialization which includes processor and I/O controllers.

In-Kernel: This layer is responsible for data initialization and initializing other kernel subsystems.

Loader: This layer is responsible for loading an application by allocating required memory.

Service Subsystems:

These subsystems are responsible for providing resources for the applications at the runtime.

TYPES OF OPERATING SYSTEMS

1) **General purpose OS:**

   This type of OS is designed for desktop computers. It provides lots of services and supports any applications. Here the Kernel will be very heavy.

2) **Embedded OS:**

   This type of OS is designed for closed devices usually targeted for RISC (Reduced Instruction Set computing) architecture, and services are fine tuned for specific set of applications. The kernel will be light.

3) **Real Time OS:**

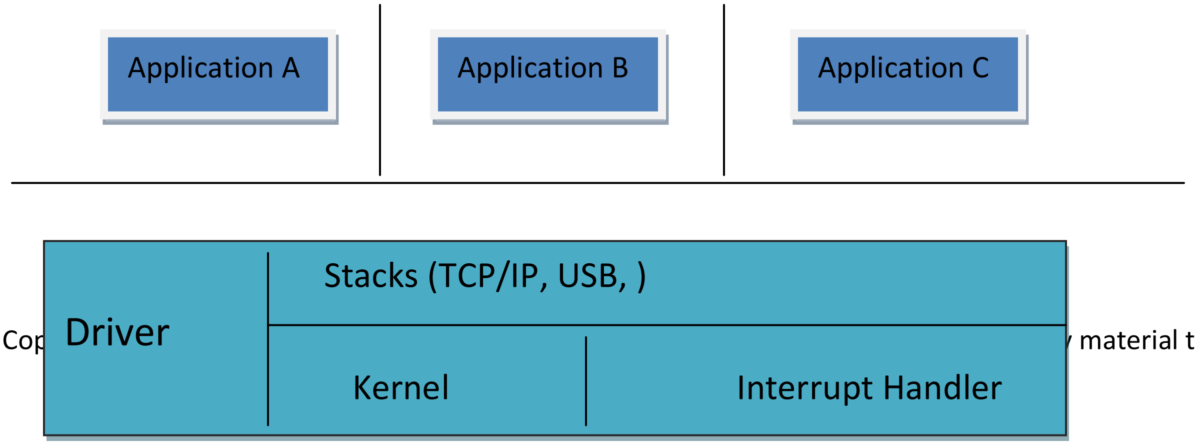
   This OS has fixed time response.

Operating System Design:

In earlier days, user applications and Kernel were aligned together. There by any bug in that application crashed the entire OS.

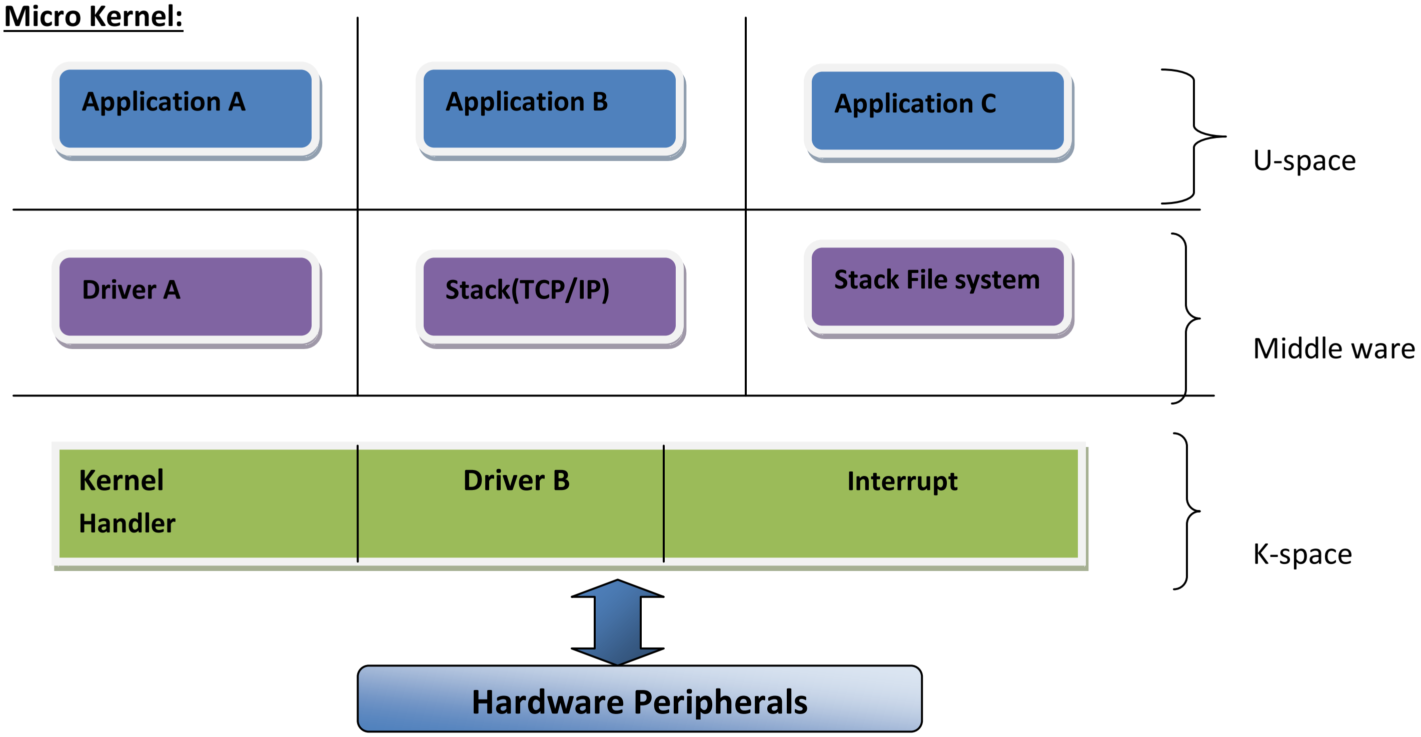
To ensure that exceptions caused by an application do not result in kernel crash, memory (RAM) is logically partitioned into kernel space and User space. Applications are loaded in user space and Kernel is loaded in kernel space.

Monolithic Kernel:



- Most of the Desktop OS are monolithic. The applications are completely abstracted from kernel.
- The main drawback of Monolithic kernel is the buggy drivers installed from un-trusted sources can crash the kernel.

Micro Kernel:



- Here the third party services are put on the middle layer.
- Example: Android.
