Bare Machine and Resident Monitor

In the early days of computing, before the development of modern operating systems, the execution of programs was done directly on the hardware. This was made possible by two fundamental components: Bare Machine and Resident Monitor. These components are although outdated in today's computing environment but once they were important for memory management and task execution.

Bare Machine

A Bare Machine is a computer system that operates directly with its hardware. It executes programs in machine language without the need for an operating system. Before the development of operating system, the only drawback was that it accepted instructions in only machine language due to which only people with sufficient knowledge about Computer field are able to operate a computer. Hence, after the development of the operating system, Bare machine is considered as inefficient.

Working of Bare Machine

- The processor executes machine language instructions provided by the user.
- There is no abstraction between the user and the machine. Hence, the system is highly dependent on the knowledge of machine code.
- Each instruction is processed by the hardware without the help of an operating system which results in a basic, raw computing environment.

Advantages of Bare Machine

1. **Direct Control**: Users can directly control the hardware without any interference from an operating system which can be beneficial in low-level tasks or specialized applications.
2. **Efficiency in Simple Tasks**: For basic, single-task operations, Bare Machines can be faster than using an operating system as there is no overhead.
3. **Low Resource Requirement**: Bare Machines do not require resources or memory needed by an operating system which makes them useful in environments with limited resources.

Disadvantages of Bare Machine

1. **Lack of User-Friendliness**: Bare Machines only accept machine language. Users must have a understanding of low-level programming.
2. **No Multitasking**: Unlike systems with operating systems, Bare Machines cannot handle multiple tasks simultaneously which makes them inefficient for most general-purpose use cases.
3. **Limited Scope**: Bare Machines are inefficient for complex applications as they lack the abstraction and multitasking features provided by modern operating systems.

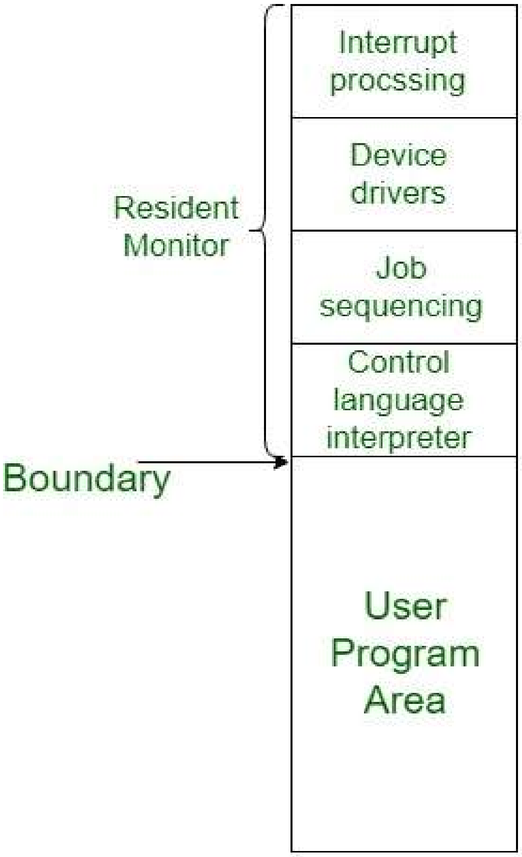
Resident Monitor

The Resident Monitor is program that runs on Bare Machines. The resident monitor works like a primitive operating system that controls the instructions and performs all necessary functions such as job scheduling, memory management, and interrupt processing.

Parts of Resident Monitor

It is divided into 4 parts as:

- **Control Language Interpreter**: The first part of the Resident monitor is control language interpreter which is used to read and carry out the instruction from one level to the next level.
- **Loader:** The second part of the Resident monitor which is the main part of the Resident Monitor is Loader which Loads all the necessary system and application programs into the main memory.
- **Device Driver:** The third part of the Resident monitor is Device Driver which is used to manage the connecting input-output devices to the system. So basically it is the interface between the user and the system. it works as an interface between the request and response. request which user made, Device driver responds that the system produces to fulfill these requests.
- **Interrupt Processing:** The fourth part as the name suggests, it processes the all occurred interrupt to the system.



Memory Layout of Resident Monitor

Working of Resident Monitor

- It works like job sequencer as it sequences job and sends them to the processor.
- After scheduling the job Resident monitors loads the programs one by one into the main memory according to their sequences.
- It controls the execution of instructions and ensures that programs are executed without any interruption in between.
- The Interrupt Processing feature allows the system to handle signals or requests from the hardware during execution, ensuring smooth operations.

Advantages of Resident Monitor

1. **Job Sequencing**: The Resident Monitor handles job sequencing and memory management.
2. **Faster Execution**: There is no time gap between job scheduling and execution, making the system faster compared to manual job handling.
3. **No OS Overhead**: The Resident Monitor allows program execution without the overhead of a operating system, making it lighter in terms of resource consumption.
4. **Basic System Control**: It provides a minimal, yet effective level of control over hardware resources.

Disadvantages of Resident Monitor

1. **Limited Functionality**: The Resident Monitor provides only basic control over the system, and lacks the extensive features of a modern operating system.
2. **No Multitasking**: Like the Bare Machine, Resident Monitors cannot handle multiple programs running simultaneously, which limits their use in more complex environments.
3. **Not Scalable**: The Resident Monitor is not scalable for larger applications or systems that require more advanced management.
4. **Error Handling**: As with Bare Machines, error handling is more difficult because there is no higher-level operating system to catch and handle errors gracefully.