|  |  |
| --- | --- |
| Semester | T.E. Semester VI – Computer Engineering |
| Subject | SBL Cloud Computing |
| Subject Professor In-charge | Prof. Divya Nimbalkar |
| Assisting Teachers | Prof. Divya Nimbalkar |
| Laboratory | M313A |

|  |  |
| --- | --- |
| Student Name | Deep Salunkhe |
| Roll Number | 21102A0014 |
| TE Division | A |

**Title: Virtualization**



**Explanation:**

**Hypervisor:**

A hypervisor, also known as a virtual machine monitor (VMM), is a software layer that enables multiple operating systems to run concurrently on a single physical host machine. It abstracts the underlying hardware resources and provides virtualized environments called virtual machines (VMs) in which guest operating systems can operate as if they were running on dedicated physical hardware.

There are two types of hypervisors:

1. **Type 1 Hypervisor (Bare Metal Hypervisor)**: This type of hypervisor runs directly on the physical hardware without the need for an underlying operating system. It manages the hardware resources and provides virtualization services directly to the guest operating systems. Examples include VMware vSphere/ESXi, Microsoft Hyper-V, and Xen.
2. **Type 2 Hypervisor (Hosted Hypervisor)**: This type of hypervisor runs on top of a conventional operating system (known as the host OS). It relies on the host OS to manage hardware resources and provides virtualization services through the host OS to the guest operating systems. Examples include Oracle VirtualBox, VMware Workstation, and Parallels Desktop.

Hypervisors enable efficient resource utilization, isolation between VMs, and flexibility in managing virtualized environments.

**KVM (Kernel-based Virtual Machine):**

KVM is an open-source virtualization technology built into the Linux kernel. It allows the Linux kernel to act as a hypervisor, transforming the host operating system into a bare-metal hypervisor. KVM provides hardware-assisted virtualization capabilities by leveraging hardware virtualization extensions (such as Intel VT-x and AMD-V) present in modern CPUs.

Key features of KVM include:

1. **Full Virtualization**: KVM allows running unmodified guest operating systems (including Windows, Linux, BSD, etc.) directly on the virtualized hardware, providing near-native performance.
2. **Para-virtualization Support**: KVM also supports para-virtualization, where guest operating systems are modified to be aware of the virtualized environment, improving performance and efficiency.
3. **Memory Management**: KVM manages memory resources efficiently by leveraging the Linux kernel's memory management capabilities, such as memory overcommitment and dynamic memory allocation.
4. **Device Virtualization**: KVM supports various virtualized devices, including virtual CPUs, memory, disks, network interfaces, and graphics adapters.
5. **Management Tools**: KVM can be managed using various tools, including command-line utilities like **virsh** and graphical user interfaces like **virt-manager**.



**Implementation:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



**End Result:**

A screenshot of a computer

Description automatically generated

A computer screen shot

Description automatically generated



**Conclusion:**

A hypervisor is a critical software layer facilitating the concurrent operation of multiple operating systems on a single physical machine. KVM, a Type 1 hypervisor, is integrated into the Linux kernel, effectively converting the host OS into a bare-metal hypervisor. Leveraging hardware virtualization extensions, KVM offers full virtualization support for unmodified guest operating systems and para-virtualization options for enhanced performance. With efficient memory management, device virtualization capabilities, and robust management tools like **virsh**, KVM is widely adopted in enterprise environments and cloud platforms for its scalability, performance, and seamless integration with the Linux ecosystem.