A System Memory Management Unit (SMMU) is an essential component in modern SoCs, particularly in ARM-based architectures. It is responsible for managing virtual-to-physical memory translations for I/O devices.

For a Senior Engineer role in a Linux environment, you can expect interview questions spanning across SMMU architecture, Linux IOMMU subsystem, device drivers, debugging, and performance tuning. Below is a detailed list categorized by topics:

1. Fundamentals of SMMU & IOMMU

* What is an SMMU, and why is it used in modern SoCs?
* How does SMMU help in secure memory isolation?
* What are the key differences between ARM SMMU v1, v2, and v3?
* How does an IOMMU differ from a CPU MMU?
* Explain translation contexts, Stream IDs, and ASIDs in SMMU.
* What is device passthrough mode, and when would you use it?
* What is a page table walk, and how does an SMMU handle it?
* How does ATS (Address Translation Service) work in SMMU?
* What is stall mode, and how does it differ from bypass mode?
* What are TLB shootdowns, and how do they affect performance?

2. Linux IOMMU Subsystem

* How does Linux support SMMU/IOMMU?
* What are the key kernel configuration options for enabling SMMU?
* How do you enable and configure IOMMU groups in Linux?
* Explain the purpose of /sys/kernel/iommu\_groups/.
* How does Linux handle IOMMU faults?
* What is VFIO (Virtual Function I/O), and how does it work with SMMU?
* How does the Linux DMA mapping API interact with IOMMU?
* What is the difference between iommu\_map() and iommu\_map\_sg()?
* Explain iommu\_ops and its role in Linux.
* What is default domain type, and what are its different modes (identity, DMA, etc.)?

3. Device Driver Integration

* How do you enable IOMMU support in a device driver?
* What changes are required in a PCIe or platform device driver for SMMU?
* How does the iommu\_attach\_device() function work?
* What is the role of iommu\_domain and iommu\_fwspec?
* How do you handle IOMMU fault recovery in a device driver?
* Explain iommu\_unmap() and its implications on device operations.

4. Debugging & Performance Optimization

* How do you debug IOMMU translation failures?
* What tools do you use to debug IOMMU faults (dmesg, iommu\_dump)?
* How do you analyze IOMMU latency and performance bottlenecks?
* How does Linux perf help in debugging SMMU performance issues?
* What kernel logs should you check when a device is failing due to IOMMU?
* How do you profile IOMMU TLB misses?
* Explain the role of stall events and PRI (Page Request Interface) in debugging.

5. Security & Virtualization

* How does IOMMU protect against DMA attacks?
* What are DMA fences, and how do they relate to IOMMU security?
* How does SMMU work in a virtualized environment (KVM, Xen)?
* How does SR-IOV (Single Root I/O Virtualization) interact with SMMU?
* Explain Nested IOMMU Translation and its use cases.
* What are IOMMU context banks, and how are they used in virtualization?

6. ARM-Specific SMMU Features

* What is Stage 1 vs. Stage 2 translation in ARM SMMU?
* How does SMMU interact with SMMU-500 and MMU-600?
* What are the key differences between SMMU v2 and SMMU v3?
* How do you configure SMMU in UEFI/ACPI vs. device tree (DTB)?
* Explain Command Queue (CQ) and Event Queue (EQ) in SMMU v3.

7. Real-World Scenarios

* How do you debug IOMMU errors when adding a new PCIe device?
* A device is experiencing high latency when using IOMMU—how would you debug and optimize?
* A faulting DMA transaction is causing system instability—how do you isolate the issue?
* Your system has an SMMU page table corruption issue—how do you investigate?
* How would you analyse a performance degradation after enabling IOMMU?

8. Advanced Topics

* What is IOMMU batching, and how does it improve performance?
* How does PCIe ATS (Address Translation Service) interact with IOMMU?
* Explain SVA (Shared Virtual Addressing) in IOMMU.
* What is IOMMU bounce buffering, and how does it impact performance?
* How does IOMMU interact with coherent vs. non-coherent DMA?