Linux Ques.

C Language:

1. Memory Layout of a C program.
2. Different Storage classes (storage of variables section in Memory Layout)
3. Where is uninitialised variable stored? Explain BSS section.
4. What is static variable? Difference between Local static vs Global static.
5. Difference between static variable and global variable.
6. const char *ptr **vs** char *const ptr
7. Size of **int** in different architectures.
8. What is structure padding and why is it necessary? How do we remove structure padding(A: #pragma pack(1))?
9. What is Volatile keyword? (A: To avoid compiler optimization). Explain
10. What is register keyword? What happens if number register keywords used in any C code are more than the number of CPU registers?
11. Difference between Register keyword and Volatile keyword.
12. What is dangling pointer, NULL pointer and void pointer?
13. How to free a pointer without using free() system call?
14. Zombie process and Orphan process.
15. Memcpy vs strcpy.
16. Difference between inline and #define macro.

Linux Internals:

1. Linux Booting process (Explain in detail from ROM code till runlevel selection and login prompt)
2. What is Bootloader? What are the core peripherals initialised in this stage?
3. RISC vs CISC architecture difference.
4. IO mapped IO **vs** Memory mapped IO
5. Process **vs** Thread. Explain (considering stack section)
6. fopen() **vs** open()
7. IPC mechanism. Types of IPC?
8. What is preemption?
9. What are atomic operations?
10. Synchronisation Techniques. Difference between Mutex, Semaphore and Binary Semaphore.
11. Explain Mutex and spinlocks usage.
12. Where spinlocks are used outside of Interrupt conext?
13. Explain Deadlock. How to avoid deadlock? How to come out of deadlock?
14. What is Race Condition?
15. How system call works. Explain
16. Types of interrupts. Difference between exceptions, interrupts and signals.
17. Different types of bottom halves. Explain
18. What is memory leak and how to avoid it?
19. What is DMA and How it works.
20. What is page fault and demand paging?
21. What is the advantage of Virtual Memory?
22. What is memory leak?

Linux Kernel:

1. How to write a Basic device driver.
2. How do you debug a Device Driver?
3. Linux Platform driver vs Device driver.

4. Device tree binding vs Device tree overlay
5. ATAGs vs DTB
6. About **okay**, **disable** and **compatible** usage in DTS.
7. How malloc works internally?
8. Difference between kmalloc and vmalloc.
9. Memory allocation. Buddy allocator vs Slab allocator
10. How much is actual Kernel source size.(Excluding device drivers).
11. What are different types of Kernels? Explain
12. RTOS vs GPOS
13. How a User Mode is transerred to Kernel Mode. (Change of processor state)
14. How Context Switching works?
15. How interrupt works (Processor transition, do_irq)
16. spin_lock_irqsave() **vs** spin_lock()
17. How ioctl works? (With reference to command number)
18. IRQ_HANDLED vs IRQ_NONE
19. How interrupt sharing works(IRQF_SHARED)?

Logical Ques:

1. What is Endianess. How to find Endianess of a machine
2. Implement XOR without using XOR
3. Check whether two strings are Anagram or not.
4. Check whether number is palindrome or not.
5. How to find given number is power of 2
6. Toggle alternate Bits.
7. How to delete **n**th node from the last in singly linked list?
8. How to find a loop in Linked List?
9. How to delete a node in singly linked list?
10. How to delete a node a in singly linked list without head pointer?

Misc:

1. I2C protocol, I2C bus arbitration and Clock stretching. Explain
2. What is Bit binding in I2C protocol?
3. JTAG, GDB or any debugging tools
4. Linux Kernel crash dump analysis.

References:

1. https://www.geeksforgeeks.org/c-language-2-gq/
2. http://linuxdevicedrivercinterviewqs.blogspot.com
3. https://doc.lagout.org/programmation/unix/Linux%20System%20Programming%20Talking%20Directly%20to%20the%20Kernel%20and%20C%20Library.pdf
4. https://www.doc-developpement-durable.org/file/Projets-informatiques/cours-&-manuels-informatiques/Linux/Linux%20Kernel%20Development,%203rd%20Edition.pdf