Linux Internals Interview Questions & Answers for Freshers

What is the difference between a process and a thread?

Process is an independent program in execution with its own memory space. Thread is a lightweight process that shares memory with other threads of the same process.

1. What is a system call?

It is a function used by user programs to request a service from the kernel (e.g., read(), write(), fork()).

2. What is the purpose of fork()?

It creates a new process by duplicating the calling process. The new process is called the child process.

3. What is a zombie process?

A process that has completed execution but still has an entry in the process table.

4. How to identify a zombie process?

Using 'ps aux | grep Z', or check for processes with STAT as Z.

5. What is an orphan process?

A process whose parent has terminated, but it's still running. It's adopted by init (PID 1).

6. What is the use of exec() family functions?

These replace the current process image with a new process image (e.g., execl(), execp(), execv()).

7. Difference between wait() and waitpid()?

wait() waits for any child. waitpid() can wait for a specific child.

8. What is a context switch?

The process of storing and restoring the state of a CPU so that multiple processes can share a single CPU.

9. What is the purpose of the init process?

It's the first user-space process started by the kernel. It manages system startup and services.

10. What is the role of the kill command?

Sends a signal to a process. By default, it sends SIGTERM.

11. What are common signals in Linux?

SIGINT, SIGKILL, SIGTERM, SIGSEGV, SIGSTOP, SIGHUP.

12. Can SIGKILL be caught or ignored?

No, SIGKILL and SIGSTOP cannot be caught, blocked, or ignored.

13. What is signal masking?

Temporarily blocking a signal using sigprocmask().

14. What is a real-time signal?

Signals in the range of SIGRTMIN to SIGRTMAX with queuing support and delivery in order.

15. What is virtual memory?

An abstraction where each process gets its own address space. It maps to physical memory via page tables.

16. What is a page fault?

When a program accesses a page not currently in physical memory, the kernel brings it in.

17. What is segmentation fault?

An invalid memory access, usually due to accessing restricted memory areas.

18. What is mmap()?

Maps files or devices into memory for faster access.

19. What is the role of brk() and sbrk()?

Used for dynamic memory allocation in older systems (controls the end of the data segment).

20. What are file descriptors?

Integers that uniquely identify an open file in a process (0: stdin, 1: stdout, 2: stderr).

21. What is the difference between read() and fread()?

read() is a system call; fread() is a library function.

22. What is inode?

A data structure that stores metadata about a file.

23. What is the purpose of lseek()?

Changes the file offset within an open file.

24. Difference between hard link and soft link?

Hard link points directly to the inode; Soft link (symbolic) points to the filename.

Linux Internals Interview Questions & Answers for Freshers

25. What is pthread?

POSIX threads - a standard for implementing multithreading.

26. What is mutex?

A locking mechanism to prevent simultaneous access to a resource.

27. Difference between mutex and semaphore?

Mutex is for mutual exclusion (1 lock at a time). Semaphore can allow multiple access (counting).

28. What is race condition?

A condition where the output depends on the sequence/timing of threads.

29. How to avoid race conditions?

Use synchronization tools like mutexes, semaphores, or atomic operations.

30. What is kernel space vs user space?

Kernel space has full system access; user space is limited for protection.

31. What is a system call interface?

The gateway between user space and kernel space.

32. How does context switch happen between kernel and user space?

Through system calls or interrupts.

33. What is the function of the scheduler?

Decides which process to run next based on scheduling algorithms.

34. What is the difference between monolithic and microkernel?

Monolithic kernel has all services in one large block. Microkernel runs minimal services in kernel space.

35. What is the Linux boot sequence?

BIOS/UEFI -> Bootloader -> Kernel -> Init/Systemd -> User Processes

36. What is the role of GRUB?

Loads the kernel into memory and transfers control.

37. What is initrd or initramfs?

Temporary root file system used during boot before the real root FS mounts.

38. What is the purpose of /proc?

Virtual file system showing process and kernel info.

39. What is udev?

Device manager for the Linux kernel, handles device events.

40. How to check running processes?

ps, top, htop, pidof

41. How to check memory usage?

free -m, top, vmstat

42. How to check CPU load?

uptime, top, mpstat

43. How to monitor system logs?

dmesg, /var/log/syslog, /var/log/messages

44. How to view open files?

lsof, fuser

45. What is a device driver?

A software module that allows the kernel to communicate with hardware.

46. What are character and block devices?

Character devices are accessed byte-by-byte, block devices in blocks.

47. What is major and minor number in device files?

Major identifies driver, minor identifies device.

48. How to create a device file?

Using mknod /dev/mydev c 240 0

49. What are types of IPC?

Pipes, Message Queues, Shared Memory, Semaphores, Sockets

50. What is a loadable kernel module?

A piece of code that can be added to the kernel at runtime without rebooting.

51. How do you insert and remove kernel modules?

Use insmod to insert and rmmod to remove kernel modules.

52. What is the use of lsmod command?

Lists currently loaded kernel modules.

53. What is the role of modprobe?

Loads a module along with its dependencies.

54. What is dmesg used for?

Displays kernel-related messages, especially during boot or device events.

55. What is the function of sysfs?

A virtual filesystem that exposes kernel device structures.

56. What is the /etc/fstab file?

Configuration file for auto-mounting filesystems at boot.

57. What is the cron daemon used for?

Schedules recurring tasks using crontab files.

58. What is the purpose of the nice and renice commands?

Used to change process priority.

59. What is strace?

A debugging tool to trace system calls made by a process.

60. What is the purpose of the watchdog timer?

Monitors system health and resets the system if it hangs.

61. What is a race condition in device drivers?

When multiple threads access shared data without proper synchronization.

62. What is a kernel panic?

A fatal error from which the system cannot recover.

63. How do you debug kernel code?

Using printk, KGDB, or dynamic debug features.

64. What is the use of O_NONBLOCK flag in open()?

Allows non-blocking I/O on files or devices.

65. What is the difference between blocking and non-blocking I/O?

Blocking waits until operation completes; non-blocking returns immediately.

66. What are watchdog devices?

Hardware/software timers that reset the system if not reset periodically.

67. What is latency in real-time systems?

The time delay between event occurrence and its handling.

68. What is preemption in the Linux kernel?

Ability to interrupt a task in kernel mode to run a higher priority task.

69. How can you reduce latency in embedded Linux systems?

Use real-time kernels, minimize interrupts, optimize code paths.