1.Child process-fork()

fork() is a system call in Linux used to create a new process.

It creates a child process which is a copy of the parent process.

Both parent and child execute independently after the fork.

The child gets a return value of 0; the parent gets the PID of the child.

2.Handing common signals

Signals are notifications sent to a process by the OS or another process.we can use signal() or sigaction() to handle or ignore these signals.

SIGINT – Ctrl+C (interrupt)

SIGTERM – request to terminate

SIGKILL – force kill (cannot be caught)

SIGSEGV – segmentation fault (invalid memory access)

3.Exploring different Kernel crashes

Kernel crashes occur due to severe bugs in kernel-level code, often causing a system freeze or panic.

Common causes:

Null pointer dereference

Buffer overflows

Divide by zero

Invalid memory access

4.Time complexity

Describes how long an algorithm takes based on input size. Expressed using Big O notation (O(n), O(log n)).

Linear search: O(n)

Binary search: O(log n)

Bubble sort: $O(n^2)$

Helps in comparing algorithm efficiency.

5.Locking mechanism-mutex/spinlock

Mutex: Used to prevent multiple threads from entering a critical section. If already locked, the thread sleeps/waits. Good for long operations.

Splinlock: Lock used where threads spin (busy wait) until the lock is free. Better for very short tasks in kernel or low-latency code.