

3. Know about the following topics and explore them (Write a note on your understandings)

Areas for exploration,

Child process - fork()

Handling common signals

Exploring different Kernel crashes

Time complexity

Locking mechanism - mutex/spinlock

### **1. Child Process - fork()**

- fork() is a system call in Unix/Linux used to create a new process, known as a child process.
- The child process is a duplicate of the parent process, but with a unique process ID.
- After a fork(), both processes execute independently from the point of the fork.
- Return values:
  - 0 for the child process,
  - Positive PID of the child for the parent,
  - -1 if the creation fails.

### **2. Handling Common Signals**

- Signals are software interrupts delivered to a process to notify it of events (e.g., SIGINT, SIGKILL, SIGTERM, SIGSEGV).
- Common signal handling functions:
  - signal() – assigns a handler to a signal.
  - sigaction() – more robust signal handling.
- Example: handling SIGINT (Ctrl+C) to safely close files or clean resources before exit.

### **3. Exploring Different Kernel Crashes**

- Kernel crashes occur due to bugs in kernel code, invalid memory access, hardware failures, or corrupted modules.
- Tools to explore:
  - dmesg – shows kernel logs.
  - kdump – kernel crash dumping mechanism.
  - gdb – debug kernel crash dumps.
- Common crash types: Null pointer dereference, stack overflow, use-after-free, kernel panic.

#### 4. Time Complexity

- Time complexity measures the computational time an algorithm takes relative to input size ( $n$ ).
- Common notations:
  - Constant:  $O(1)$
  - Linear:  $O(n)$
  - Logarithmic:  $O(\log n)$
  - Quadratic:  $O(n^2)$
- It helps compare algorithm efficiency and scalability.

#### 5. Locking Mechanism - Mutex / Spinlock

- **Mutex (Mutual Exclusion):**
  - Used to prevent concurrent access to shared resources in multithreaded programs.
  - Blocks the thread until the lock is released.
  - Suitable for user-space synchronization.
- **Spinlock:**
  - A busy-wait lock where the thread waits in a loop ("spins") checking the lock.
  - Suitable for short, low-contention critical sections in kernel space.
  - Doesn't cause context switching but can waste CPU cycles.