Signals are a fundamental, albeit limited, form of Inter-Process Communication (IPC) in Unix-like operating systems. They function as asynchronous notifications sent to a process or a specific thread within a process to indicate the occurrence of an event.

Key characteristics and uses of signals in IPC:

- **Asynchronous Notification:**

  Signals interrupt the normal flow of execution of a target process, delivering an immediate notification of an event without requiring the process to be actively waiting for it.

- **Event-Driven:**

  They are primarily used to signal various events, including:

- User-initiated actions (e.g., pressing Ctrl+C generating `SIGINT`).
- System-generated events (e.g., a process attempting to access invalid memory triggering `SIGSEGV`).
- Notifications from other processes (e.g., a parent process receiving `SIGCHLD` when a child process terminates).
- **Limited Data Content:**

  Signals themselves carry no data beyond their type (e.g., `SIGTERM`, `SIGKILL`). While some advanced signal mechanisms like `sigqueue` can deliver a small integer value, signals are not designed for general data transfer.

- **Signal Handlers:**

  Processes can register custom signal handlers to define specific actions to take upon receiving a particular signal, overriding the default system behavior (e.g., graceful termination instead of immediate exit).

- **Non-ignorable Signals:**

  Certain signals, notably `SIGKILL` (immediate termination) and `SIGSTOP` (pause process), cannot be caught, ignored, or handled by a custom handler, ensuring system control.

- **Simplicity and Efficiency:**

  Signals are a lightweight IPC mechanism with minimal overhead, making them suitable for basic synchronization and event notification.

- **Limitations:**
  Due to their asynchronous nature and lack of data content, signals are not ideal for complex data exchange or reliable message passing where order and content are crucial. Other IPC mechanisms like pipes, message queues, or shared memory are more appropriate for such scenarios.