Session - 29

Pipes in Inter-Process Communication (IPC)

Pipes are a fundamental IPC mechanism used in Unix/Linux systems to allow **data flow between processes**. They provide a **unidirectional communication channel**, where one process writes data to the pipe and another process reads from it.

- **Types of Pipes**

1. Unnamed Pipes (Anonymous Pipes)

- Used for communication **between parent and child processes**.
- Exist **only in memory** and **only during the lifetime** of the processes.
- Created using the pipe() system call.

Syntax:

int pipe(int fd[2]);

- fd[0] → read end
- fd[1] → write end

Example:

```
#include <unistd.h>

#include <stdio.h>
```

```
#include <string.h>

int main() {
    int fd[2];
    char buffer[20];

    pipe(fd); // create pipe

    if (fork() == 0) { // child process
        close(fd[1]); // close write end
        read(fd[0], buffer, sizeof(buffer));
        printf("Child read: %s\n", buffer);
    } else { // parent process
        close(fd[0]); // close read end
        write(fd[1], "Hello", 6);
    }
    return 0;
}
```

2. Named Pipes (FIFOs)

- Used for communication **between unrelated processes**.
- Exist as **a special file** in the filesystem.
- Created using mkfifo() or mknod().

Example:

mkfifo myfifo

In one terminal:

echo "Hello from writer" > myfifo

In another terminal:

cat < myfifo

🔸 Key Characteristics

Feature	Unnamed Pipe	Named Pipe (FIFO)
Direction	Unidirectional	Unidirectional
Persistence	Temporary	Persistent (as a file)
Scope	Parent-child only	Any unrelated process
Creation	pipe()	mkfifo()
File System	No	Yes

🔹 Limitations of Pipes

- **Unidirectional**: For two-way communication, you need **two pipes**.
- **Buffer size**: Limited (usually 4KB–64KB depending on the system).
- **Blocking I/O**: Reading/writing blocks until the other end is ready (unless using non-blocking I/O).

- **No message boundaries**: Pipes transmit raw byte streams, not messages.

- **Alternatives for More Complex IPC**
  - **Message Queues**
  - **Shared Memory**
  - **Semaphores**
  - **Sockets**