

## Interprocess Communication

Three techniques are used to perform communication between two programs P1 (Sender) and P2 (Receiver):

### **1. Unix domain sockets**

Sender program

- `socket()` : creates a new socket and returns a file descriptor for communication
- `bind()` : binds the file descriptor returned by `socket()` to a local address
- `listen()` : shows that created socket is willing to accept incoming request and sets the upper bound of incoming requests
- `accept()` : waits to receive a connection request and accepts request using first in - first out policy

Receiver program

- `socket()` : creates a new socket and returns a file descriptor for communication
- `connect()` : initiates a connection request to connect the socket to the specified address

The usual file handling functions of `read()`, `write()` and `close()` are used to communicate the strings, their indexes and acknowledge the success of communication.

### **2. FIFO**

Two FIFOs are used, one (`myfifow`) for sending the strings and their respective indexes in groups of 5, and the other (`myfifor`) for sending acknowledgement as the highest index received to indicate successful communication.

Sender program

- `mkfifo()` : creates a named pipe with specified permissions
- `open()` : opens the named pipe in specified mode and returns a file descriptor
- `unlink()` : deletes the name from the filesystem and also the file if no processes have the file open

Receiver program

- `open()` : opens the named pipe in specified mode and returns a file descriptor
- `unlink()` : deletes the name from the filesystem and also the file if no processes have the file open

### 3. Shared memory

#### Sender program

- `shm_open()` : creates a shared memory segment in specified mode and returns a file descriptor
- `ftruncate()` : truncate the size of the shared memory segment
- `mmap()` : maps the specified file descriptor to the address space of the calling process, returning the starting address of the mapping
- `munmap()` : deletes the mapping from the address space
- `shm_unlink()` : removes the shared memory name and also destroys the memory space if no processes are using it

#### Receiver program

- `shm_open()` : opens the shared memory segment created in specified mode by sender and returns a file descriptor
- `mmap()` : maps the specified file descriptor to the address space of the calling process, returning the starting address of the mapping
- `munmap()` : deletes the mapping from the address space
- `shm_unlink()` : removes the shared memory name and also destroys the memory space if no processes are using it

The return address from `mmap()` is used to write/read data to/from the shared memory segment by sender and receiver programs.

#### General points

1. The `clock_gettime()` function is used to clock the timing for execution of interprocess communication using each of the three techniques. The time observed for communication of 50 strings of 128 characters with these techniques is as follows –

Technique	Time observed in seconds
Socket	0.002773
FIFO	0.001414
Shared Memory	0.509203

2. The `srandom()` function is used to change the seed for generating random numbers on each distinct execution of the programs.
3. The random numbers generated using `random()` are mapped to the ASCII value of visible characters to generate random strings of 128 characters