

# Question 1 : Prime Number Parallel Processing

---

This C program calculates the sum of prime numbers up to 100 million using parallel processing and pipe as communication path. Here's how it works:

## 1. Prime Number Checking

The `check_prime()` function implements a basic primality test. It checks if a number is prime by:

- Quickly filtering out even numbers and 1
- Using trial division up to the square root of the number

## 2. Parallel Processing Strategy

The program splits the work across multiple CPU cores by:

- Determining available CPU cores using `sysconf(_SC_NPROCESSORS_ONLN)`
- Dividing the range (0 to 100 million) equally among cores
- Using `fork()` to create child processes for each range
- Using pipes for communication between parent and child processes

## 3. Process Communication

- Each child process finds primes in its assigned range
- Found primes are sent through a pipe to the parent process
- The parent process reads these primes and maintains a running sum

## 4. Pipe Communication Details

The program uses Unix pipes for inter-process communication:

```
int pipe_fd[2]; // pipe_fd[0] for reading, pipe_fd[1] for writing
```

### Parent Process:

- Creates the pipe before forking
- Closes write end (`pipe_fd[1]`)
- Reads from pipe using:

```
while (read(pipe_fd[0], &prime, sizeof(prime)) > 0)
{
    total_sum += prime;
}
```

### Child Processes:

- Close read end (`pipe_fd[0]`)
- Write prime numbers through pipe:

```
write(pipe_write, &i, sizeof(i));
```

- Close write end before exiting

## Main Flow

1. Creates pipe for communication
2. Splits work among available CPU cores
3. Child processes find primes in their ranges
4. Parent process collects results and sums them
5. Program prints the final sum

This design allows for efficient parallel processing of a computationally intensive task, significantly reducing the time needed to find all primes up to 100 million compared to a single-threaded approach.

## Key Points About Pipe Communication

- Pipes provide unidirectional data flow
- Parent reads from multiple children through the same pipe
- Data is transmitted in binary format (integers)
- Pipe automatically handles synchronization between processes
- System manages pipe buffer, blocking when full