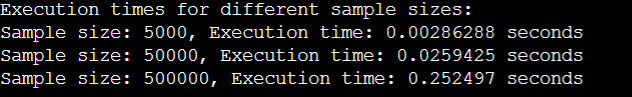
1. یک آرایه‌ی ۲۵ خانه‌ای ایجاد می‌کند که همه‌ی خانه‌ها به مقدار صفر مقداردهی اولیه می‌شوند.

از مولد اعداد تصادفی برای تولید اعداد بین ۰ تا ۱۰۰ استفاده می‌کند.

برای هر نمونه، ۱۲ عدد تصادفی تولید کرده و بسته به مقدار آن‌ها شمارنده را افزایش یا کاهش می‌دهد.

مقدار نهایی شمارنده را به عنوان ایندکس در آرایه‌ی اصلی استفاده می‌کند و مقدار آن خانه را افزایش می‌دهد.

زمان اجرای برنامه برای اندازه‌های مختلف نمونه را محاسبه کرده و نمایش می‌دهد.



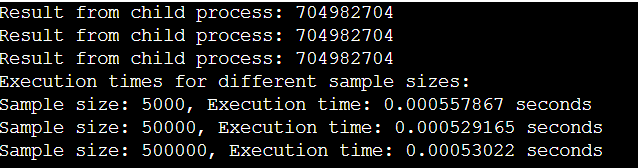
1. یک لوله (pipe) برای ارتباط بین فرآیندها ایجاد می‌کند.

از fork() برای ایجاد یک فرآیند فرزند استفاده می‌کند.

در فرآیند فرزند، محاسباتی انجام می‌دهد و نتیجه را از طریق لوله به فرآیند پدر ارسال می‌کند.

در فرآیند پدر، نتیجه را از لوله دریافت می‌کند و منتظر می‌ماند تا فرآیند فرزند به پایان برسد.

زمان اجرای برنامه را برای هر اندازه نمونه اندازه‌گیری و ذخیره می‌کند.



1. بله، این برنامه می‌تواند دچار شرایط رقابتی (Race Condition) شود. شرایط رقابتی زمانی رخ می‌دهد که دو یا چند فرآیند یا رشته همزمان به یک منبع مشترک دسترسی پیدا کنند و نتیجه نهایی به ترتیب دسترسی آنها بستگی داشته باشد.

در این مثال، اگر فرآیندهای فرزند به متغیری مشترک دسترسی داشته باشند یا سعی کنند به طور همزمان داده‌ای را به والد بفرستند، ممکن است نتایج غیرمنتظره‌ای رخ دهد. در مثال ارائه‌شده، از لوله (pipe) برای انتقال داده استفاده شده که تا حدودی شرایط رقابتی را کنترل می‌کند، اما اگر این انتقال داده همزمان باشد، مشکلاتی ممکن است بروز کند.

برای جلوگیری از شرایط رقابتی می‌توان از مکانیزم‌های هماهنگی مثل قفل‌ها (locks)، ممانعت‌ها (semaphores) یا mutex با کنترل دسترسی مناسب استفاده کرد. یکی از روش‌های ساده برای ارتباط ایمن بین فرآیندهای والد و فرزند استفاده از لوله‌ها به همراه یک سیستم مدیریت صف است.