General

Projects and its respective folders are listed below. Detailed information about each project can be found in their respective folders.

Problem 7.15

In Exercise 4.27, you developed a program to generate the Fibonacci sequence. The program required the parent thread to wait for the child thread to finish its execution before printing out the computed values.

If we modify the program to allow the parent thread to access the Fibonacci numbers as soon as they are computed by the child thread, without waiting for the child thread to terminate, what changes would be necessary to the solution? Implement this modified solution.

Solution

folder: `fibonacci_thread_sync`

7.17

Exercise 4.24 required you to design a multithreaded program to estimate π using the Monte Carlo technique. Initially, you were tasked with creating a single thread responsible for generating random points, with the result stored in a global variable. After this thread exited, the parent thread calculated the estimated value of π.

Now, modify the program so that you create multiple threads, each generating random points and determining if they fall within the circle. Each thread should update the global count of points falling within the circle. To safeguard against race conditions on updates to the shared global variable, utilize mutex locks.

Solution

folder: `monte_carlo_sync`

8.32

Implement the solution to Exercise 8.30 using POSIX synchronization.

In particular, represent northbound and southbound farmers as separate threads. Once a farmer is on the bridge, the associated thread will sleep for a random period of time to simulate traveling across the bridge.

Design your program to accommodate the creation of multiple threads representing both northbound and southbound farmers.

Solution

folder: `bridge`

9.28

Consider a system with a 32-bit virtual address and a 4-KB page size. Write a C program that receives a virtual address (in decimal) through the command line and outputs the corresponding page number and offset.

For instance, your program should execute like this:

```
./addresses 19986
```

And output:

```
The address 19986 contains:
page number=4
offset=3602
```

To accomplish this task, use the appropriate data type to handle 32 bits. It's recommended to utilize unsigned data types.

Solution

folder: `memory`