Topics: Multithreaded Programming

Approach: Study Two Problems

Today's New Ideas: Thread, Mutex, Condition Variable

Outline

Statement of Two Projects
- Animation in parallel - many animations on one web page
- Counting in parallel - counting votes, counting words in documents
- Big Idea: one program, but several things happen at once

Idea: A Thread of Execution
- A flow of control through a sequence of program steps
- Example: hello_simple.c
- Traditional programs have one thread of execution

New Idea: Multiple Threads (of Execution)
- A single program can run multiple, simultaneous threads
- A thread runs a function, which may call other functions
- Example: hello_multi.c
- Functions: pthread_create(), pthread_join()

Shared Variables: The Two Big Questions
- Good: All threads in a program share global variables
- Example: incprint.c - two threads share a counter
- Questions:
  - Can simultaneous access to memory cause problems?
  - How does the reading thread know when there is new data?

Answer 1: Cooperation - Mutual Exclusion Locks
- Bad: All threads in a program share global variables
- Example: twordcount1.c - Two threads update one counter
- Solution: twordcount2.c - Use a mutex to prevent simultaneous access
- Side issue: twordcount3.c - Multiple arguments for threads

Answer 2: Coordination - Condition Variables
- Question: How can a thread know when to read a shared variable?
- Answer: A thread waits for a signal from another thread
- Analogy: Receiving early voting returns from small towns
- Solution: twordcount4.c - Use a condition variable
- Condition variables - The dinner bell of programming
  - Threads can wait for variable to be signaled, and
  - Other threads can signal the variable
- Note: A condition variable is used with a locked, shared variable

Project 2: Multi-threaded Animation
- Example: tbounce1d.c - threads for use input and for animation
- Example: tanimate.c - Multiple animations, multiple threads