Concurrent Programming

Sequential vs. Concurrent

- Sequential Program
  - Sequence of actions that produce a result (statements + variables)
  - Called a process, task, or thread (of control)
- Concurrent Program
  - Two or more processes that work together

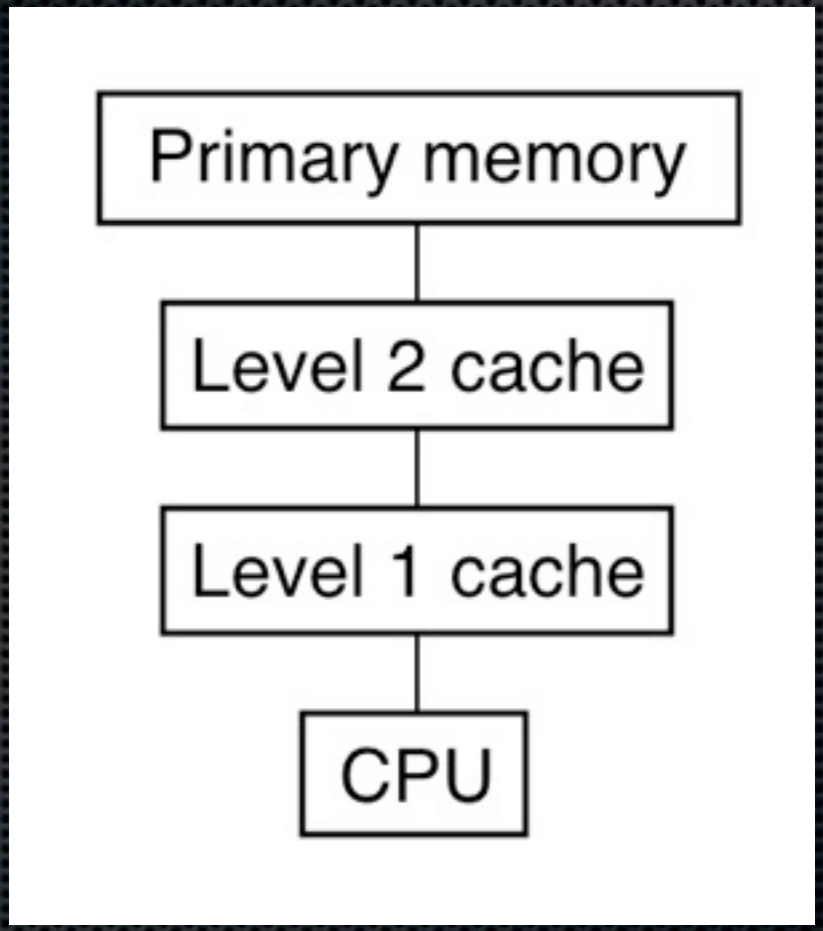
communication and synchronization

shared variables or message passing

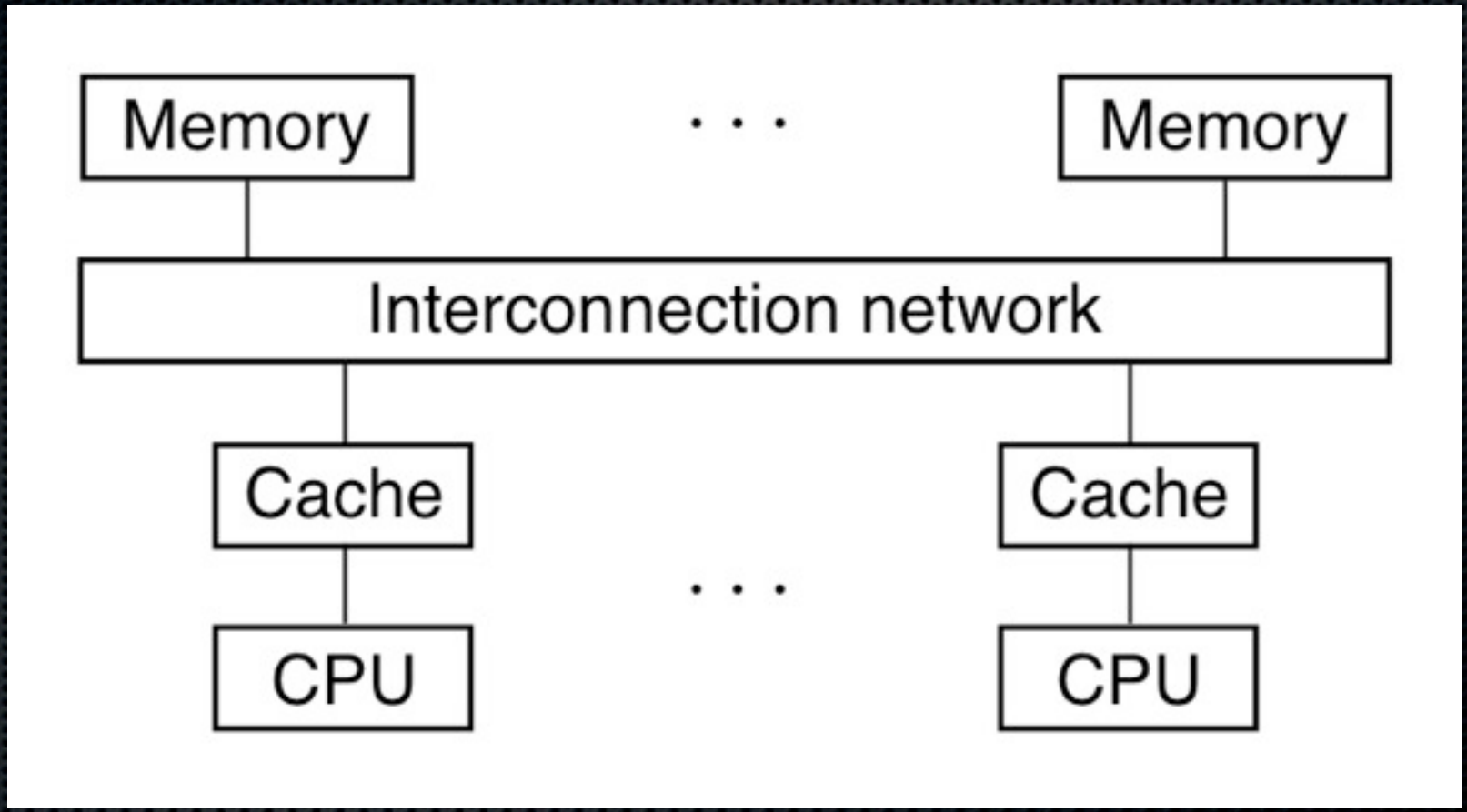
Hardware

- Single processor
- Multiprocessor -- shared memory
- Multicomputer -- separate memories
- Network -- slower communication

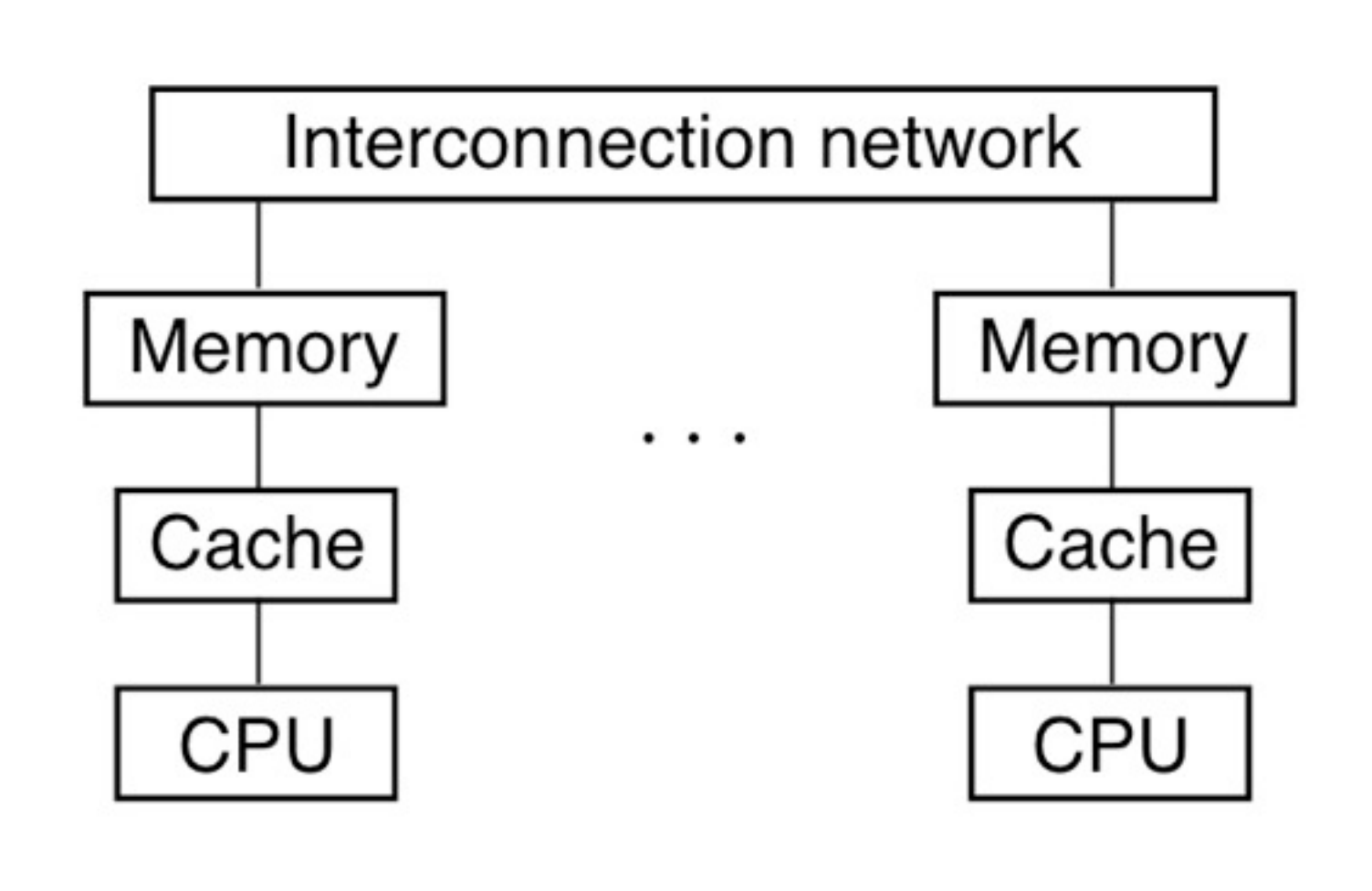
Single Processor



Multiprocessor: Shared Memory



Multicomputer: Separate Memories



Multithreaded Applications

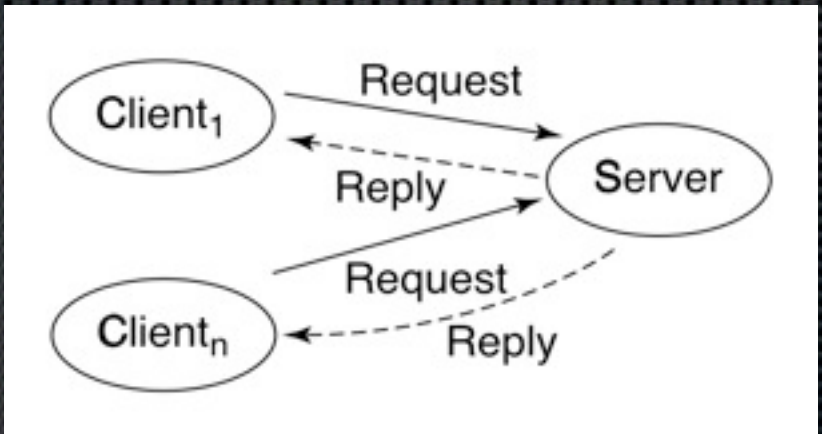
- What? More than 1 thread (usually share CPU time)
- Why? good way to organize modern software systems
  - OS -- timesharing, servers
  - PC -- windows
  - browser -- applets
  - user -- Unix pipes (this book "sed | eqn | groff")

Parallel Applications

- What? Processes execute on their own processor
- Why? Solve a problem faster -- or solve a larger problem
- Two main algorithm/programming styles:
  - Iterative -- loops, divide them up
  - Recursive -- divide and conquer, with calls in parallel

Distributed Applications

- What? Processes communicate over a network
- Why? Offload work -- servers



  - ➡Connect to remote data -- Internet, airlines, banks
  - ➡Scalable parallel computing on multicomputers and networks

Programming Paradigms

- Iterative Parallelism
- Recursive Parallelism
- Producers and Consumers
- Clients and Servers
- Interacting Peers