Lecture 25: Concurrency

- Why concurrency?
- Threads
- Race conditions

Announcements:
- Discussions are review this week
- Practice exams posted; study guide up soon
- P6 up soon.
- Course evaluations
- Processor does one thing at a time.
- Computers seem to do many things at once.
- Operating system keeps track of many tasks, switches back and forth between them (rapidly). Doesn’t make computer faster!
- Adds context switching overhead.

Modern computers may have multiple processors, which may have multiple cores. Each core is doing one thing at a time.
Clock speed: how often processor does this
- cycles per second
  \( \text{cycles per second} = \frac{1}{\text{Hz}} \)

Clock speeds increased exponentially for many decades.

"Moore's law": clock speeds double every 2 years or so.
- Transistors per chip also double every 2 years or so.

Barriers caused by heat dissipation become less useful because processors can only be so complicated!
Thread represents a task, like a separate running program. Current line of code executed.

Thread 2: stack

heap

new Thread()

h() g(...)= f(...)

g() f() main(...) static area

class C

static init x