Architecture Overview

• Topics
  • Processors, cores, and threads
  • Hardware architectures

• Learning Objectives:
  • Explain the different ways in which parallelism emerges on today’s hardware platforms.
  • Explain the difference between a thread context, a core and a processor.
Uniprogramming vs Multiprogramming

- **Uniprogramming**: a system runs only one “thing” (process or program) at a time.
  - MS-DOS
  - Old-batch systems
- **Multiprogramming**: a system that appears to run multiple “things” at once.
  - Also called multitasking.
  - Multiple programs run concurrently, even if there is only one program is running at a given instant.
- **Multiprocessing**: True concurrency
  - The hardware is actually capable of running things simultaneously, because it has multiple processing elements (intentionally ambiguous, to be defined later).
- **Contrast**: Multiprogramming/multitasking refer to the number of programs running. Multiprocessing refers to there being more than one processing element in the system (historically multiple processors; today multiple cores or multiple [hardware] threads)
• Your basic processor:
  • 1 Chip
  • 1 Execution Core
  • 1 L1 Cache
  • 1 L2 Cache
  • 1 Memory
Architecture 101 (2)

Registers
- General purpose
- Program counter
- Stack Pointer
- Arithmetic Logic Unit (ALU)
- Adder
- Multiplier
- Floating Point Unit FPU
  (optional)

Execution Core

L1 Cache

L2 Cache

Memory
Architecture 101 (3)

Really fast memory. 
~4 CPU cycles 
Typically a few tens of KB.
Fast memory.
~10 CPU cycles.
Typically a few tens of MB.
Architecture 101 (5)

Main memory.
~40-60 cycles
Typically several to many GB.
**Multicore:**
- 1 chip
- Multiple execution cores
- Multiple L1 caches
- Single L2 Cache
- 1 Memory
Architecture 101 (7)

- **Multithreading**
  - Hyperthreading
  - 1 Chip
  - Multiple execution cores
  - 1 L1 Cache
  - 1 L2 Cache
  - 1 Memory
Architecture 101 (8)

- **Multithread/Multicore**
  - 1 Chip
  - Multiple cores
  - Multiple L1 caches (1 per core)
  - Multiple execution contexts per core
  - 1 L2 Cache
  - 1 Memory
Architecture 101 (9)

- **Modern Multiprocessor**
  - Multiple chips
  - Multiple cores per chip
  - Multiple threads per core
  - L1 Cache/core
  - L2 Cache/chip
  - Shared memory

- **Sys/161**
  - Not multithreaded
  - Does not distinguish cores from processors
  - Think of as N-way single-core, multiprocessors

---

2/4/16

CS161 Spring 2016 11