

## Course Review

Kai Shen

1

## Introduction to Computer Systems

- Abstraction and theory are helpful:
  - Abstract data types, asymptotic analysis
- The study of computer systems confronts reality beyond simple abstractions:
  - Real implementation, real hardware artifacts, real security vulnerability, real performance anomalies, ...
- Acquire a broad understanding of computer systems, its components, and how they work together.
- Become more effective programmers
  - Able to find and eliminate bugs efficiently
  - Able to understand and tune for program performance

2

## Bits and Arithmetic Computation

- Are integer additions associative?
  - $(x + y) + z = x + (y + z)?$
- How about floating point additions?
  - $(1e20 + -1e20) + 3.14 = 3.14$
  - $1e20 + (-1e20 + 3.14) = ??$

3

## Integer C Puzzles

Initialization

```
int x = foo();
int y = bar();
unsigned ux = x;
unsigned uy = y;
```

- $ux \geq 0$
- $ux > -1$
- $x > 0 \ \&\& \ y > 0 \Rightarrow x + y > 0$
- $ux \gg 3 == ux/8$
- $x \gg 3 == x/8$

4

## You've Got to Know Assembly

- Chances are, you'll never write programs in assembly again
  - Compilers are much better & more patient than you are
- But: understanding assembly is useful and important
  - Implementing system software
    - Compiler has machine code as target
    - Operating systems do weird things (save/restore process state)
  - Access special hardware features
    - Processor model-specific registers
  - Tuning program performance
    - Understand optimizations done / not done by the compiler
    - Understanding sources of program inefficiency
  - Creating / fighting malware
    - x86 assembly is the language of choice!

5

## Example: Read Timestamp Counter

- Timestamp Cycle Counter (useful for fine-grained time tracking)
  - Special 64-bit register on many processors
  - Incremented every clock cycle
  - Read with rdtsc instruction
- Write small amount of assembly code using GCC's asm facility

```
static unsigned cyc_hi = 0;
static unsigned cyc_lo = 0;

/* Set *hi and *lo to the high and low order bits
   of the cycle counter.
*/
void access_counter(unsigned *hi, unsigned *lo)
{
    asm("rdtsc; movl %%edx,%0; movl %%eax,%1"
        : "=r" (*hi), "=r" (*lo)
        : "%edx", "%eax");
}
```

6

## Memory Referencing Errors

- C and C++ do not provide any memory protection
  - Out of bounds array references
  - Invalid pointer values
  - Abuses of malloc/free
- Can lead to nasty bugs
  - Whether bug has any effect depends on system/compiler
  - Consequence at a distance
    - Corrupted object logically unrelated to one being accessed
    - Effect of bug may only be observed long after it is generated
- How can I deal with this?
  - Program in language with stronger memory protection (Java)
  - Use tools to detect referencing errors (e.g., gdb, Valgrind)
  - Understand and write better code!

7

## Locality Matters

```
void copyij(int src[2048][2048],
            int dst[2048][2048])
{
    int i,j;
    for (i = 0; i < 2048; i++)
        for (j = 0; j < 2048; j++)
            dst[i][j] = src[i][j];
}

void copyji(int src[2048][2048],
            int dst[2048][2048])
{
    int i,j;
    for (j = 0; j < 2048; j++)
        for (i = 0; i < 2048; i++)
            dst[i][j] = src[i][j];
}
```

21 times slower  
(Pentium 4)

- Layout of multi-dimensional array

A[0][0]      A[0][2047]   A[1][0]      A[1][2047]



8



## Locality Matters

- Temporal and spatial locality
- Locality helps performance because computer systems employ caching and prefetching
  - Computer systems employ caching and prefetching because many programs exhibit locality ⇒ chicken & egg problem!
- Locality for the following?
  - Execution code region
  - Stack region
  - Hash table
  - Sequential search
  - Binary search

9



## Computers Do More Than Computing

- Applications need to get data in and out
  - I/O system critical to program reliability and performance
- Disks are slow
  - 10 million times slower than executing an instruction in CPU
  - 100 thousand times slower than accessing the memory
  - 10 thousand times slower than a call to operating system
- Newer I/O devices: Flash-based solid-state disks
- File system abstraction
  - Files aren't all on disks
  - Disks contain more than files

10



## Computers Do More Than Computing

- Web programming
  - HTML, JavaScripts, PHP, Ajax, ...
- Computers communicate with each other over networks
  - TCP/IP socket programming
  - A good systems perspective helps understand performance, robustness, security, ...
  - A good protocol design is important for client/server programming

11



## Develop A Systems Perspective

- Things are nuanced through a systems perspective
  - A simple abstraction misses important details
  - Many parts of the systems interact in complex ways beyond simple mathematical models
- Sometimes clean principles may be derived
  - Many factors are at play but X is the key thing because ...
  - The whole system is best understood through a chain of dependencies with the root cause at ...

12

### Security In A Systems Perspective

- Security problem (like the buffer overflow problem) involves various parts of the computer systems
- Processor architecture
  - Executable segment
- Programming language and compilers
  - Stack layout
- Operating systems
  - Flow and taint tracking, protection of sensitive system operations
- Computer networks:
  - Interconnection exacerbates security threats: the problem would not have caused as much damage if it wasn't in a network server (fingerd)

13

### Performance In A Systems Perspective

- Google keyword searching
  - Each keyword has a prepared list of matching web pages
  - Merge multiple lists of matching web pages for multi-keyword queries
- Solutions (two-keyword query)
  - Coordinated scanning of two sorted lists
  - Hashtable-based matching

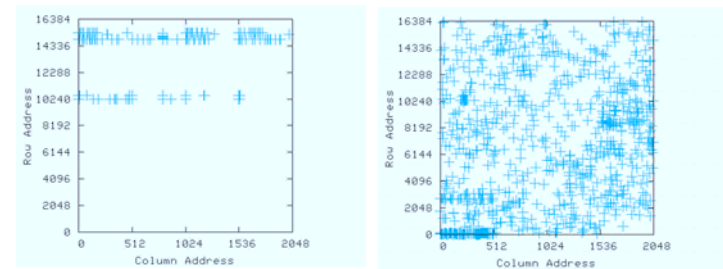
14

### Reliability In A Systems Perspective

- You build/manage a computer system but it crashes once in a while.
- Problem may be in
  - Application software
  - Systems software
  - Hardware
- Memory hardware errors
  - Measurement over 9 months on 300 machines (over 800GB memory)
  - At 99% confidence: transient error rate lower than 0.6 failures in  $10^9$  hours per Mb of memory

15

### Permanent Errors on Two Chips



16



## Computer Power/Energy Usage In A Systems Perspective

- High power/energy is expensive economically and environmentally
  - Small personal computers/devices and large data centers
- How to do better?
  - Use the right hardware
  - Write more efficient code
  - Compiler use more power-efficient instructions (speculate less?)
  - OS induce bursty activities to prolong sleep time
  - Data center energy conservation through server consolidation
  - Build your data center next to a hydro power plant

17



## Introduction to Computer Systems

- Prepare and motivate for later “systems” courses
  - Computer Architecture
  - Compilers and Programming Languages
  - Operating Systems
  - Computer Networks
  - Parallel and Distributed Systems

18