Fundamentals of Computer Systems
Reading for this lecture

- **Primary**
  - Roosta, Chapter 1

- **Secondary**
  - Miller and Boxer, Chapter 5
  - Fountain, Chapters 1 and 2
  - Cosnard and Trystram, Chapters 1 to 3
Our View of the World
The Computational Universe

Computer Systems

Hardware
- Memory
- Processors
- Interconnections
- Storage

Software
- Operating Environment
- Shared Libraries
- Application Code

OS
- Programming Language
- Compiler
- Data Structures
- Algorithms
- Implementation
Pseudo-code notation

- We will often need to write pseudo-code
- Our notation will be loosely based on C with some parallel constructs
- Declarations, etc. can be left out when the context makes it clear
- Basic functions which are not the focus of the exercise can simply be called

```c
for (i = 0; i < 10; i++)
    parallel for (j = 0; j < 10; j++)
        find the minimum element of x[i][10*j, 10*(j+1)];
```
Computer Architecture
Flynn's Taxonomy

Single Instruction Stream, Single Data Stream
(Serial Computer)
Computer Architectures
Flynn's Taxonomy

Single Instruction Stream, Multiple Data Stream (SIMD)
Computer Architectures
Flynn's Taxonomy

Multiple Instruction Stream, Single Data Stream (MISD)
Computer Architectures
Flynn's Taxonomy

Multiple Instruction Stream, Multiple Data Stream (MIMD)
Memory Configurations

Shared Memory

Memory

Interconnection Network

Processor 1  Processor 2  ...  Processor n
Memory Configurations

Distributed Memory

Interconnection Network

P1 M1

P2 M2

... ...

Pn Mn
Memory Hierarchies

Each level in this hierarchy is orders of magnitude slower than the previous one.
Importance of the memory hierarchy

- Processor-Memory performance gap grows 50% per year.
- Cache memory tries to overcome this performance gap.
- However, it is easy to defeat.
- Example:

```
for (i = 0; i < 100; i++)
    for (j = 0; j < 5000; j++)
        x[i][j] = 2*x[i][j];
```

- Declaring a variable "register" in C