

# The Programming Process

15-110 Summer 2010

Margaret Reid-Miller

# Hardware Components

- **Central Processing Unit (CPU)**
  - Program control
  - Arithmetic/logical operations
  - Coordinates data movement between memory and registers (small amount of storage)
- **Memory**
  - Main (RAM): fast, expensive, volatile
  - Secondary (Hard disk, CD, DVD): slow, nonvolatile
- **Input/Output (I/O) Devices**
  - Keyboard, mouse, camera, microphone, ...
  - Monitor, speakers, printers, ...

# Software

- **Algorithm** - A sequence of steps (strategy) to solve a problem (not necessarily computational).
- **Program** - A sequence of instructions and decisions executed by a computer to achieve a task.
  - E.g., web browser, mailer, editor, operating system, compiler, scheduler
- **Programming** helps to explain the approach computer scientists use to solve complex problems.

# Digital and Analog

- **Analog** information is continuous
  - e.g., mercury in a thermometer, sound waves, light waves
- **Digital** information is discrete and can be represented by numbers.
  - e.g., music on a CD is made by sampling the sound briefly at a high rate.
- Computers stores information in **binary** numbers, which has only two digits (0, 1)

# Why Binary?

- Simple; easy to build, reliable
  - two stable states: (on, off), (high, low), (magnetized, demagnetized),
- Unambiguous signals
  - threshold between high and low voltage
  - unaffected by slight noise
  - as it degrades the two extremes can be reinforced
- Flawless copying
- Can represent anything

# Binary Numbers

decimal	binary	decimal	binary
0	0	8	1000
1	1	9	1001
2	10	10	1010
3	11	11	1011
4	100	12	1100
5	101	13	1101
6	110	14	1110
7	111	15	1111

# Bits 'n Bytes 'n Memory

- A **bit** is a one binary digit. A **byte** is 8 bits.
- Main memory is a long list of bytes.
  - A byte is just large enough to hold a single keyboard character
  - Each byte in memory is conceptually numbered
  - The number is called its **address**.
- Terms for the size of memory are
  - Kilobyte (**KB**) is  $2^{10} = 1024$  bytes
  - Megabyte (**MB**) is  $2^{20}$  (approx. million) bytes
  - Gigabyte (**GB**) is  $2^{30}$  (approx. billion) bytes

# Machine Instructions

- Each type of computer processor has a set of **machine operations** it can perform. Each operation does one very simple thing.
  - e.g., get a value from memory and put it in a register, add the values in two registers, test if the value in a register is zero, jump to some instruction ...
- A **machine instruction** consists of several bytes of memory that tell what machine operation to perform and, if needed, what data to use.



# Program Execution

- To **execute** (run) a program the operating system does the following:
  - loads the program instructions from the hard drive into memory;
  - finds some memory for the program to use;
  - tells the CPU at what address the first instruction is;
- Then the CPU fetches one instruction at a time from memory into the CPU and carries out the operation.

# Programming Language Types

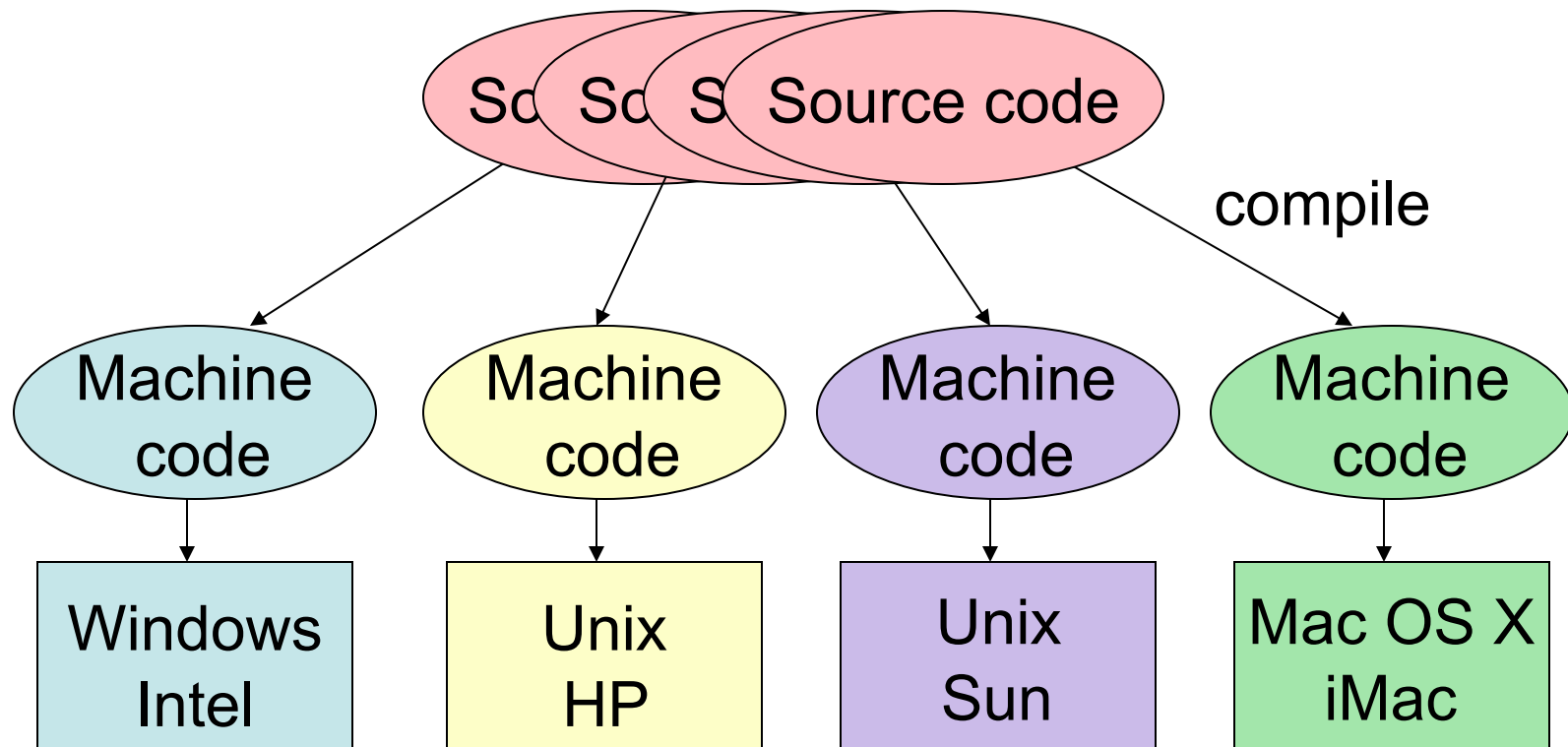
- **Machine** - Uses only operations that the specific hardware can execute directly (e.g., `1101 0000 0000 0111`)
- **Assembly** - Uses *mnemonics* to represent operations or data values (e.g., `add %o0 %o1 %o0`)
- **High-level** - Supports structured programming and use English-like phrases (e.g., C++, C, Java, Python, Lisp)  
(e.g., `x = 1 + 2;`)
- **Fourth-generation** - Describes *what* needs to be done, not *how* to do it. (e.g., SQL, SAS, Mathematica)  
(e.g., `SELECT * FROM books WHERE price < 10.0`)

# Program Translation

- **Compiler** - program that translates code (a program) in the *source* language, all at once, into another language call the *object* or *target* language.
  - E.g., translates Java to machine instructions.
  - The code is translated only once and executed many times .
- **Interpreter** - program that translates and then executes only one (or several) statement(s) at a time.
  - This translation needs to be done each time a statement is executed.

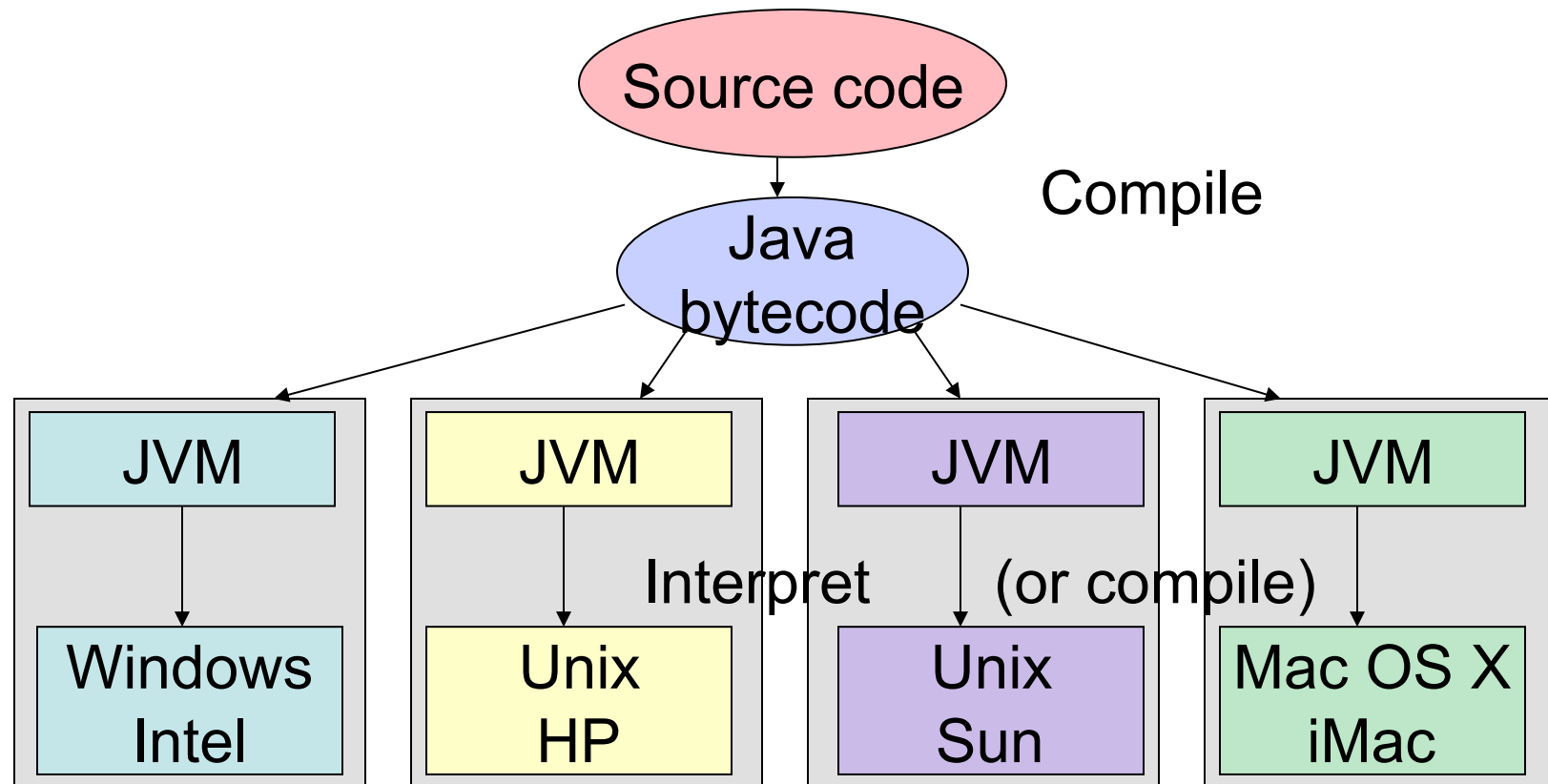
# Traditional Compilation

Different hardware/operating systems need slight variants of the program and separate compilers.



# Java Translation

Java compiles to a single “virtual” machine.



JVM - Java Virtual Machine

Summer 2010

15-110 (Reid-Miller)

13

# Programming Languages

- **Syntax** – Rules of the programming language to form valid statements (punctuation, statement structure, order of operations,...).
- **Semantics** – The meaning of the statements. That is, what will happen when the computer executes statements.

# Types of Programming Errors

- **Compile-time error**- Syntax errors found by the compiler or syntax-aware editors.
  - e.g.,  $x - * y =$
- **Run-time error** - Error causes program to stop abnormally; often called an *exception*.
  - e.g.,  $x = y / z$  (What if  $z$  is zero?)
- **Logical error** - Program produces incorrect results without complaint.
  - e.g., Sort program does not always sort the data

# Goals of the Course

- Understand basic programming concepts.
- Translate an algorithm specification into correct program code.
- Write programs to solve simple problems.
  - Writing correct programs requires “debugging” your programs: Find and correct errors that causes your program not to do what you intend.



# What is a computer system?

- **Hardware** is the electronic and mechanical parts. It is the tangible, physical device.
  - But a computer must be programmed to be useful (other than as a doorstop or sculpture).
- **Software** is the sets of instructions and data necessary to solve problems. It is intangible, conceptual.