Course Introduction

Lecture Objectives:

1) Draw the relationship between hardware, systems software, and applications software as a hierarchy
2) Explain the concept of a compiler
3) Draw a flowchart showing the basic steps of compilation for code.
4) Explain the relationship between high level languages, assembly language, and machine language

Course Description

- This course provides students with an introduction to the structure of computer hardware, including the components of a modern computer system as well as the tradeoffs necessary to construct such a system. Specific course topics include numeric systems, the role of performance in designing computer systems, Amdahl's Law, instruction formats, addressing modes, computer arithmetic with both fixed and floating point numbers, single cycle and multi-cycle data-path design, pipelining, the memory hierarchy, caching, and parallel processing using SIMD and MIMD formats. Students will develop small, assembly language programs on a simulator as a means of exploring instruction formats and data-path operation. (prereq: CE-1900)

Graphics Processing

Course Outcomes

- 1. (Comprehension) Understand the relationship between input, output, memory, the processor, the data path, and the control path within a computer.
- 2. (Comprehension) Explain how signed and unsigned numbers and floating point numbers are represented internally within a computer and perform these operations.
- 3. (Comprehension) Explain the memory hierarchy within a computer and quantify its impact on computer performance.
- 4. (Application) Explain how variables are allocated in memory and the relationship between variables and pointers.
- 5. (Application) Compute performance related metrics for a computer based system or implemented program.
- 6. (Analysis) Critique the design and implementation of a processor based upon design parameters.
- 7. (Synthesis) Write simple assembly language routines using MIPS assembly language.

Calculator may be necessary

MSOE

About the Instructor

- **Instructor:** Dr. Walter W. Schilling, Jr.
- **Office:** Walter Schroeder Library 335
- **Office Hours:**
  - ~~MTWR 10:00 10:45~~ 5:00 – 10:00
  - While I post office hours, I keep an open door policy. If I am in my office and the door is open, please feel free to stop in.
- **Telephone:** 414 277 7370
- **E-mail:** schilling@msoe.edu
  - Best method to contact me during non-class days
  - Please prefix subject with ~~CS2854~~ CS2710
- **Course Web Page:**
  - ~~https://myweb.msoe.edu/~schilling/msoe/cs3841/cs3841.shtml~~ Syllabus

About the Instructor (Continued)

- Ohio Northern University graduate in Electrical Engineering
  - Computer Science Minor
- Masters and PhD. from University of Toledo
  - Specialized in Computer Systems Design and Software Reliability
- Worked in Automotive Industry for approximately 6 years
  - Audio Software Engineer – Embedded Systems Design
    - US Patent 6,707,768
    - "Randomized Playback of Tracks in a Multimedia Player"

- Personal Website: http://www.walterschilling.org

Computer Classification

- How would you classify computing applications?

Utility => Does a specific task

Numbers

Embedded

Desktop { PC / Mac

Laptop { Mac / PC

Netbooks Subset of Laptops

Mobile Phone

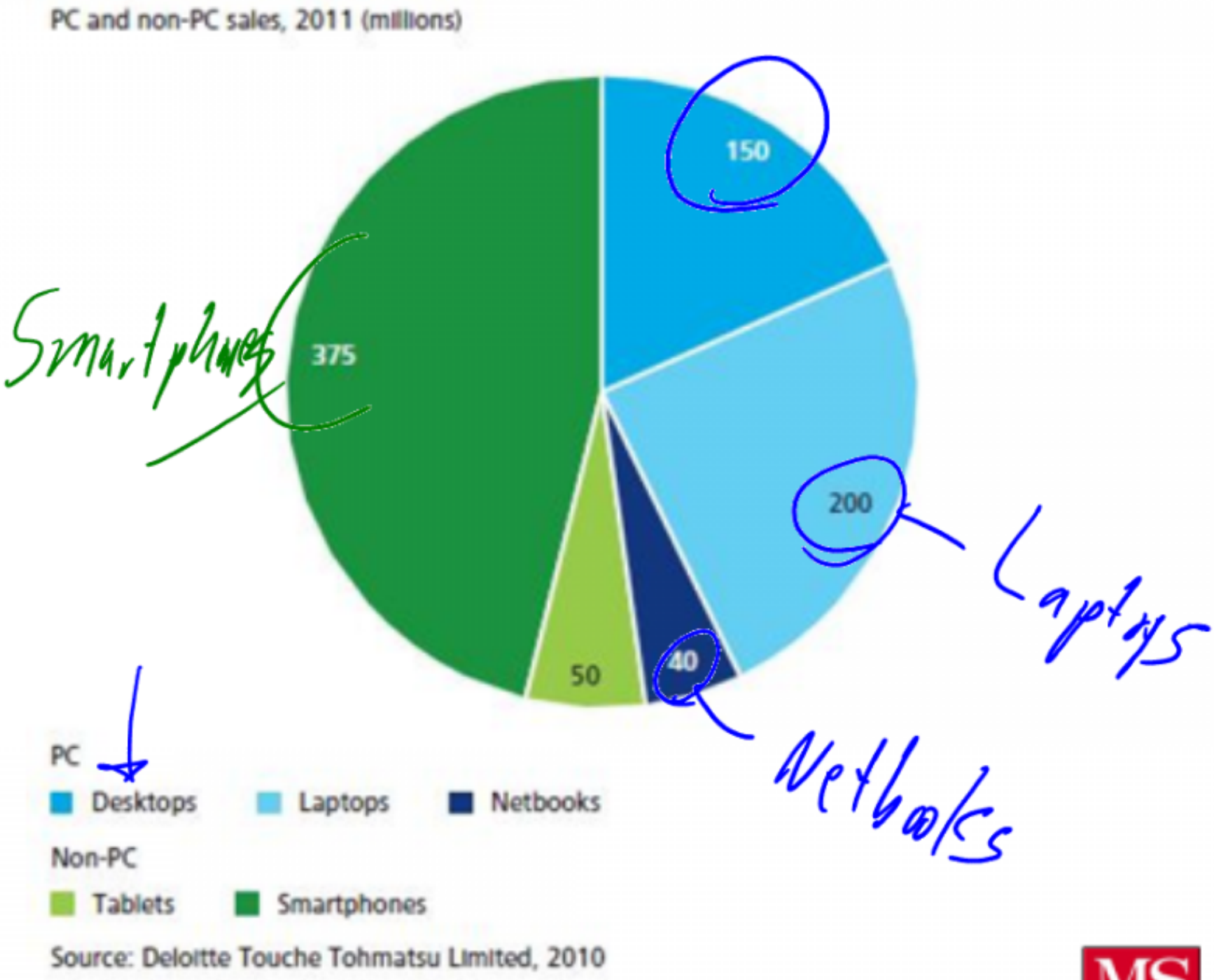
Classes of Computers

- Desktop computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

Classes of Computers

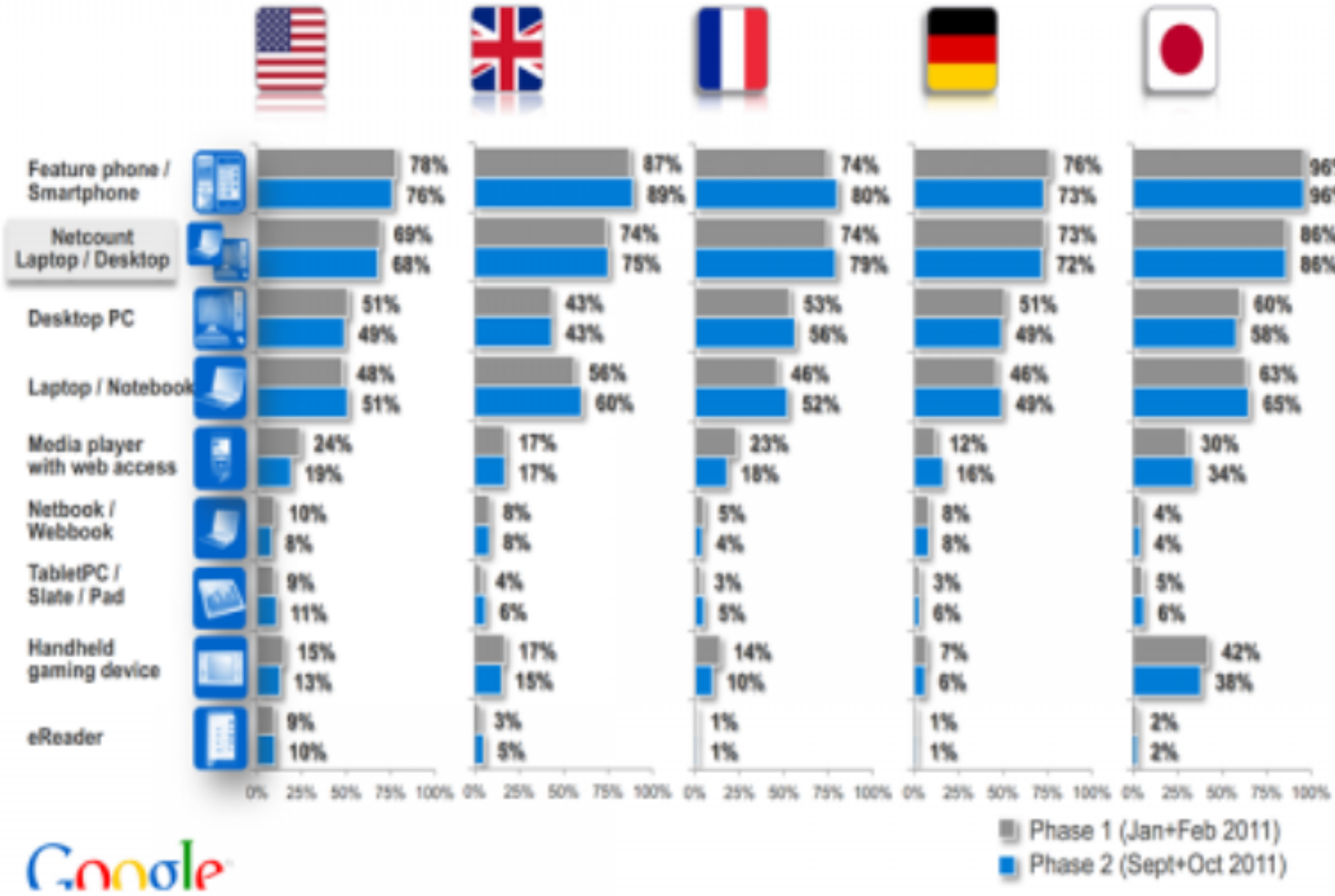
- Desktop computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

Computer Sales, 2011



Consumer Usage by device

http://services.google.com/fh/files/blogs/Google_Ipsos_Mobile_Internet_Smartphone_Adoption_Insights_2011.pdf



- I want to drive between Milwaukee and Chicago. How long will it take you?

Performance

Performance

2.5 hrs => Driving time in standard Car.

Model 4 =>

Rate Car => 200 mph => Limited

What impacts computer performance?

What impacts computer performance

- Algorithm ← How we do things?
  - Determines number of operations executed
- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation

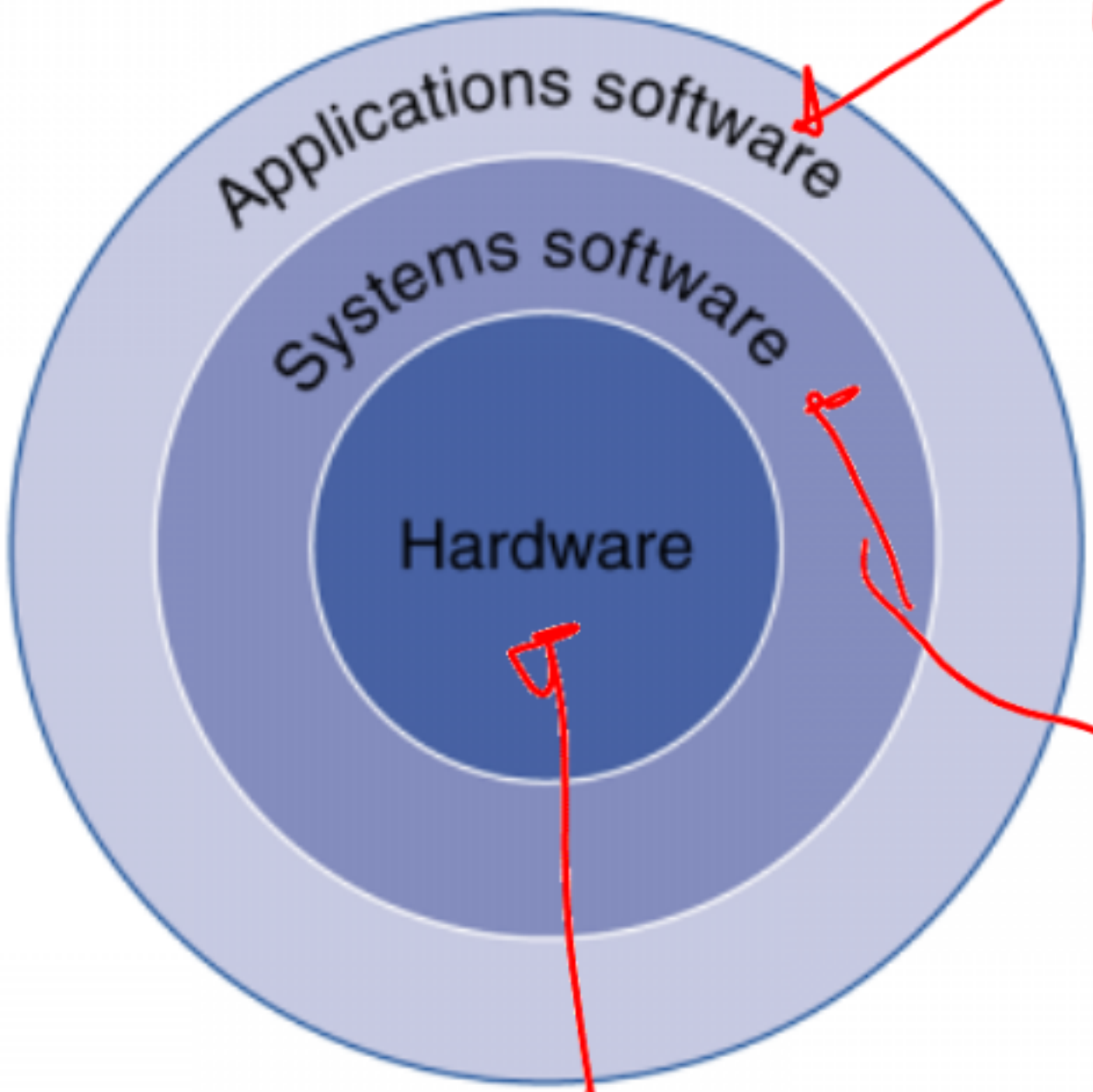
How efficient is our CPU

- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS) input / output
  - Determines how fast I/O operations are executed

How quickly can we get data in and out of the CPU?

MSOE

Your program



Highest Level program

Operating system

Hardware architecture

Levels of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data

- Talk about Java and C

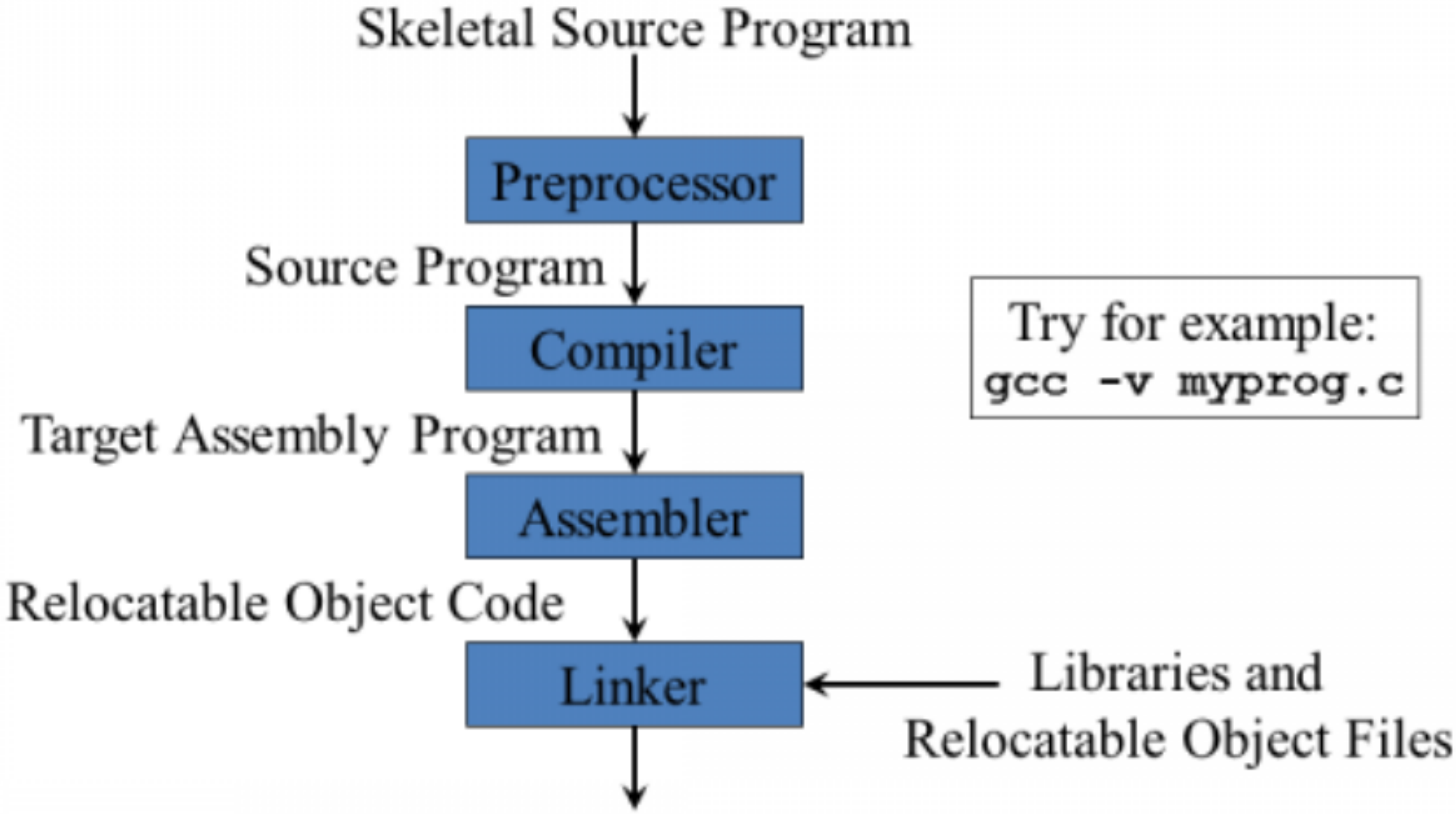
How does a program go from source code to executable?

- How does source code go from source code to an executable program?

Discussion

Possible demo

Preprocessors, Compilers, Assemblers, and Linkers



Try for example:
`gcc -v myprog.c`