Introduction to Computer Science and Programming in C

Session 8: September 25, 2008

Columbia University

Announcements

- Homework 2 out. Due 10/14 (correction)
- Midterm review 10/16, exam 10/21
- Submission procedure and deadline

Review

- Functions
- Variable Scope - when variables are valid
- Recursion - when a function calls itself

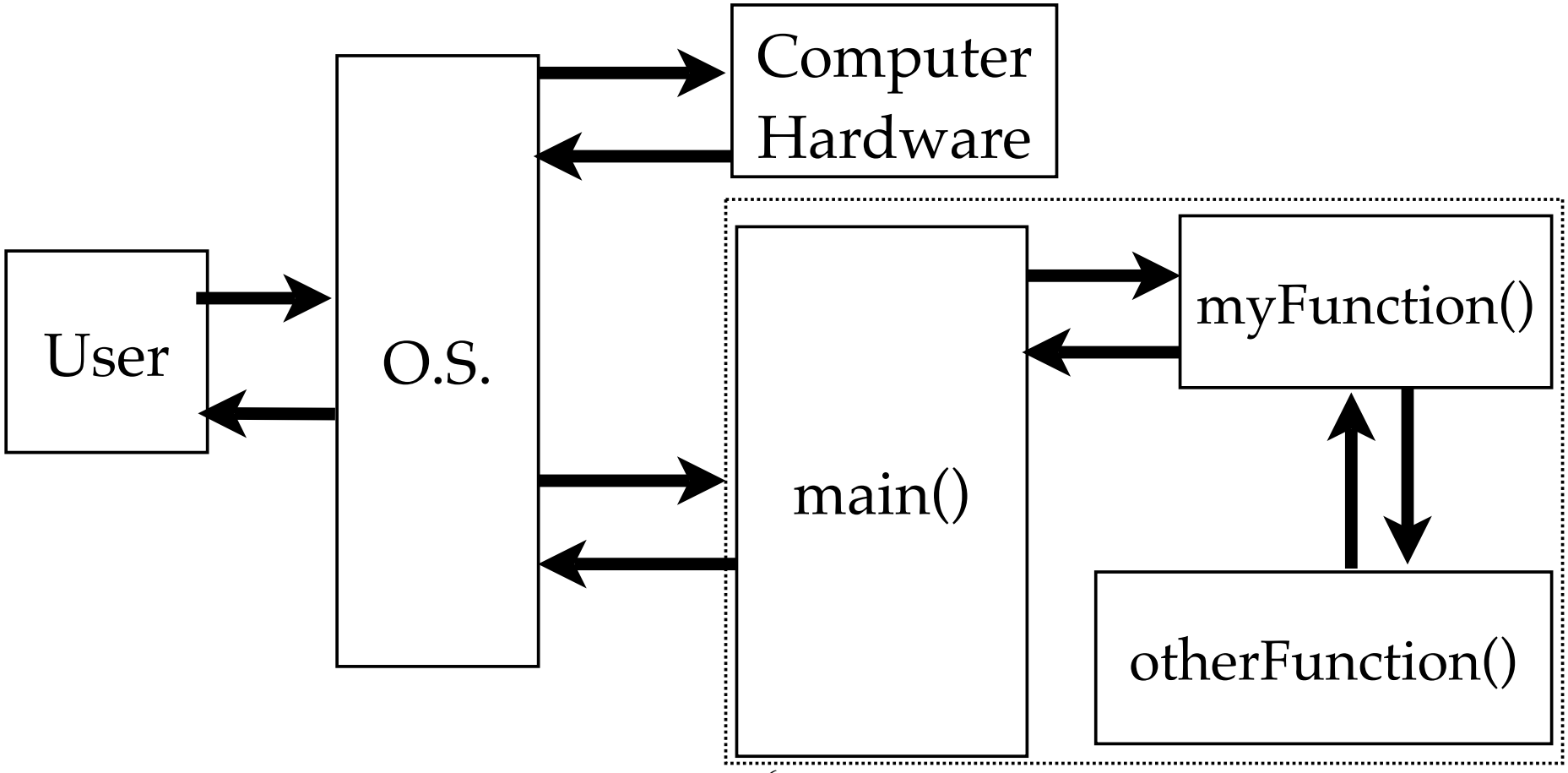
Today

- Go over HW1
- Recursion (continued)

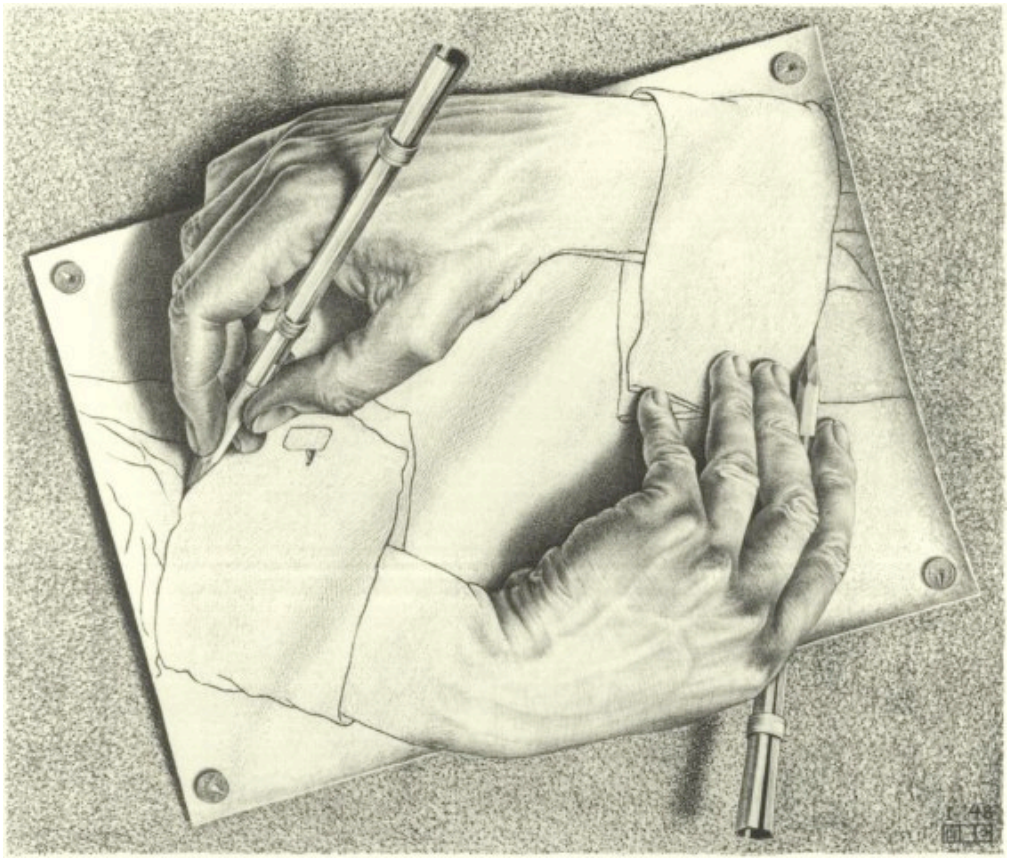
Homework 1

- http://www.cs.columbia.edu/~bert/courses/1003/homework1_soln.txt

Functions Illustrated



Recursion Examples



Painting by M.C. Escher
http://aixa.ugr.es/escher/640x480/Manos_dibujando.jpg

Silly Recursion Examples

- GNU (of gcc) stands for GNU is Not Unix
- The C Programming Language index:

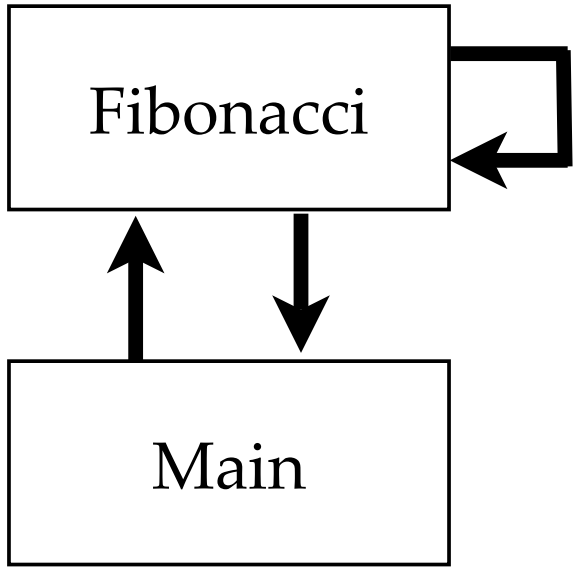
  `recursion 86, 139, 141, 182, 202, 269`

  is listed on page 269
- This sentence is not true.
- Every rule has exceptions.

Recursion Examples

- Fibonacci Sequence: 1, 1, 2, 3, 5, 8, 13, 21, 34 ...
  - Base case: fib(0) = 0, fib(1) = 1
  - fib(n) = fib(n-2)+fib(n-1)
    - fib(2) = fib(0)+fib(1) = 1
    - fib(3) = fib(1)+fib(2) = 2

Fibonacci Illustrated



Towers of Hanoi

Towers of Hanoi

- In English, target function:
  Move stack of N discs from peg A to peg B
- Base case: When N = 1, just move the disc
- What about N=2?
  - Move disc 1 from A to C,
    Move disc 2 from A to B,
    Move disc 1 from C to B

Towers of Hanoi

- N = 3?
- General rule: Move N discs from A to B
  - 1) Move stack of (N-1) discs from A to C
  - 2) Move Nth disc from A to B
  - 3) Move stack of (N-1) discs from C to B

Recursion Summary

- Simple, elegant algorithms with often complex results
- Always *possible* to use loops instead, but recursion can significantly simplify the algorithm
- Sometimes recursion can be less efficient; tradeoff between simplicity and efficiency

Reading

- Practical C Programming Ch. 9
- The C Programming Language, Ch. 4