COMP 2401/2001 – Winter 2013

Today's agenda

- C programming language
  - Some history
  - Reserved words and Primitive data types
  - Operators
  - Control structures
  - Data structures
  - Functions
  - Start data representations

C

- Developed by Dennis Ritchie
- 1969-1973 at AT&T Bell Labs
- General-purpose, imperative, procedural progamming language

- "C is quirky, flawed, and an enormous success."
  — Dennis M. Ritchie (1941-2011)

C

- Allows for low-level programming (similar assembly language) but with high-level syntax
- Closer to the machine than most languages
- Generates fast code
- Widely used for low-level applications, including the Unix kernel
- Continues to evolve
  - Many standards exist: K&R C, ANSI C, C99, C11

C

- Allows for low-level programming (similar assembly language) but with high-level syntax
- Closer to the machine than most languages
- Generates fast code
- Widely used for low-level applications, including the Unix kernel
- Continues to evolve
  - Many standards exist: K&R C, ANSI C, **C99**, C11

Reserved words

- auto, _Bool, break,
  case, char,_Complex, const, continue,
  default, do, double,
  else, enum, extern,
  float, for, goto,
  if, _Imaginary, inline, int,
  long, register, restrict, return,
  short, signed, sizeof, static, struct,
  switch, typedef, union, unsigned,
  void, volatile, while

Reserved words

- auto, _Bool, break,
  case, char,_Complex, const, continue,
  default, do, double,
  else, enum, extern,
  float, for, goto,
  if, _Imaginary, inline, int,
  long, register, restrict, return,
  short, signed, sizeof, static, struct,
  switch, typedef, union, unsigned,
  void, volatile, while

Primitive data types

- Four basic arithmetic types
  - `char, int, float, double`
  - `lots more that are basically ints`
- Optional specifiers
  - `signed, unsigned, short, long`
- In C99
  - `_Bool, _Complex`

Operators

- Arithmetic
- Relational
- Logical
- Bitwise
- Assignment
- Member & Pointer
- Other

Arithmetic operators

- Binary operators
  - addition, subtraction, multiplication, division, modulo
  - `a+b, a-b, a*b, a/b, a%b`
- Unary operators
  - minus, plus
  - `-a, +a`
  - increment, decrement (prefix and suffix)
  - `++a, a++, --a, a--`

Relational / Comparison ops

- equality
  - equal to, not equal to
  - `a==b, a!=b`
- inequality
  - greater than, less than, greater than or equal, less than or equal
  - `a>b, a<b, a>=b, a<=b`

Logical operators

- Unary
  - negation
  - `!a`
- Binary
  - Logical AND, logical OR
  - `a && b, a || b`

Bitwise operators

- Unary
  - Bitwise NOT
  - `~a`
- Binary
  - Bitwise AND, bitwise OR, bitwise XOR
  - `a & b, a | b, a ^ b`
  - Bitwise left shift, bitwise right shift
  - `a << b, a >> b`

Assignment operators

- assignment
  - `a=b`
- compound assignment
  - `a += b, a -= b, a *= b, a /= b, a %= b`
  - `a &= b, a |= b, a ^= b`
  - `a <<= b, a >>= b`
  - `a ●= b` is almost the same as `a = a ● b`

Member and Pointer operators

- Array subscript `[ ]`
  - `a[b]`
- Indirection *, reference (address of) &
  - `*a , &a`
- Structure reference `.`, structure dereference `->`
  - `a.b, a->b`

Other operators

- Funtion call, comma
  - `f(a,b) , a,b`
- size-of
  - `sizeof(a) , sizeof a, sizeof(type)`
- align-of
  - `alignof(type), _Alignof(type)`

Other operators

- Casting operator
  - `(type)a`
- Ternary conditional
  - `a ? b : c`

Operator characteristics

- Arity
  - Number of operands in the operator
- Precedence
  - BEDMAS and then some...
  - Parentheses, (and), can help!
- Associativity
  - Reading left-to-right or right-to-left?

Precedence/Associativity

- [→] a++, a--, f(a,b), a[b], a.b, a->b
- [←] ++a, --a, +a, -a, !a, ~a, *a, &a, (type) a, sizeof
- [→] a*b, a/b, a%b
- [→] a+b, a-b
- [→] a<<b, a>>b
- [→] a<b, a<=b, a>b, a>=b
- [→] a==b, a!=b

Precedence/Associativity

- [→] a&b
- [→] a^b
- [→] a|b
- [→] a && b
- [→] a || b
- [←] a ? b : c
- [←] assignments a=b, a+=b, a>>=b, etc.
- [→] a,b

Control structures

- Two selection statements (branching)
  - `if, switch`
- Three iteration statements (looping)
  - `for, while, do-while`
- Four jump statements
  - `goto, continue, break, return`

Control structures

- Two selection statements (branching)
  - `if, switch`
- Three iteration statements (looping)
  - `for, while, do-while`
- Four jump statements
  - ~~`goto`~~, `continue, break, return`

Data structures

- Arrays
  - Container for similar "things"
- Structures
  - Container for different "things"
  - Like classes/objects but without any methods
  - There are no objects/classes in C

Functions

- C is a procedural language
  - You do things by calling functions
  - Functionality is broken down into separate procedures
- Characteristics of a good function
  - Encapsulates functionality
  - Must be reusable

Functions

- ```
  type name(function paramaters){
          body
          return
  }
  ```

- ```
  int add(int x, int y){
      int answer = x + y;
      return answer;
  }
  ```

Functions

```
int add(int x, int y){
    int answer = x + y;
    return answer;
}
```

- All functions in C pass their arguments by value
  - Function gets local copy of input arguments

Comments

- Same as Java

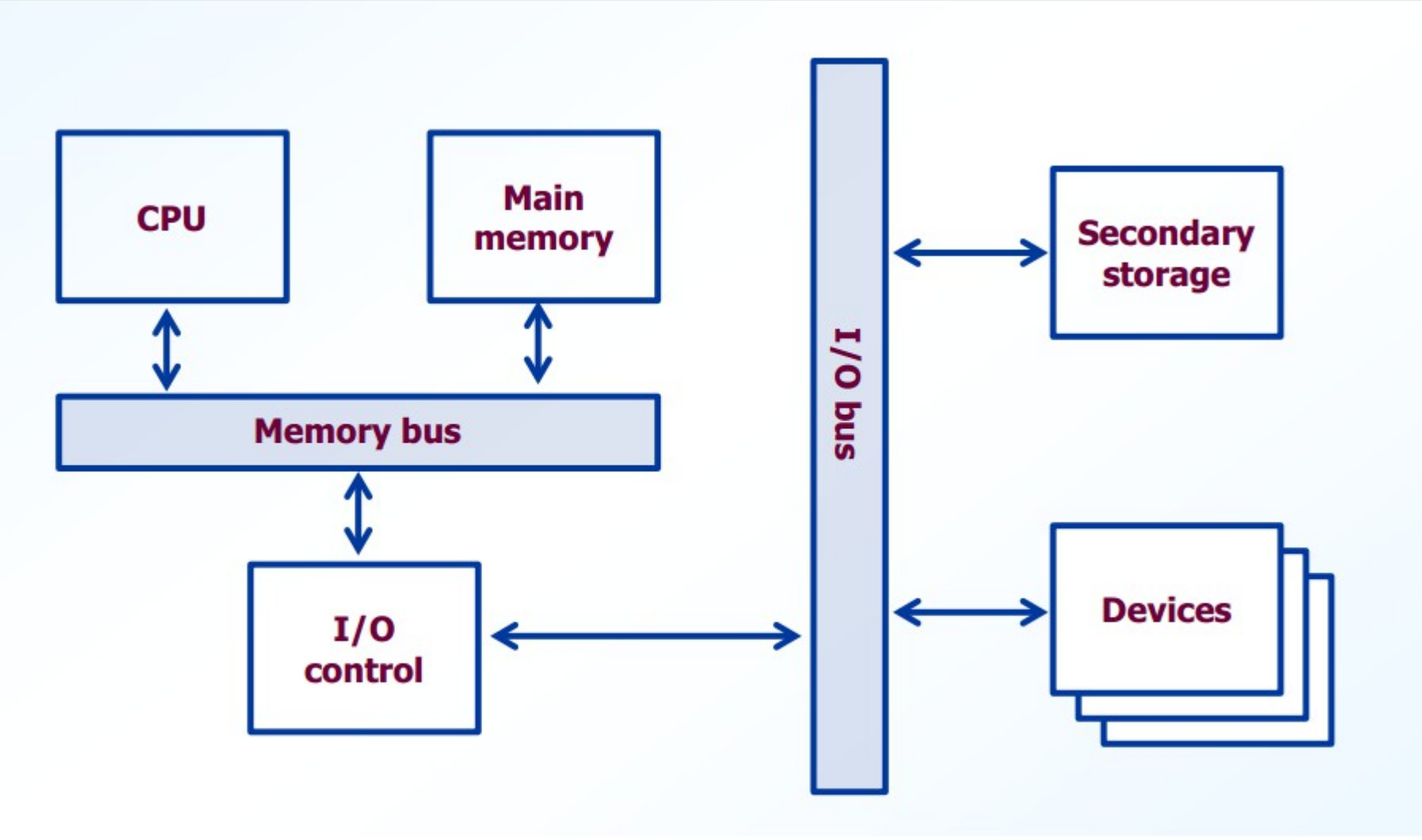
- ```
  /* ...
     multiline comments
     ...
  */

  /* single line or inline */
  ```

- ```
  // inline comment
  ```

Programming conventions

- Extremely important!
  - Naming conventions
  - Indentation
  - Commenting

Data representations



Data representations

- Bus
  - Bit highway... width is the number of lanes
  - Data is moved around via buses
  - Modern computers use
    - 32-bit width
    - 64-bit width
- Word
  - Atomic unit of data (not a bit)
  - Related to bus width

Data representations

- What does 10011100 mean?

Data representations

- What does 10011100 mean?
- Unfair question!

  it can mean whatever I want it to mean
- Need to have a proper context

Data representations

- What does 10011100 mean?
- 15th episode of season one of Star Trek Next Generation

Data representations

- Look at unsigned integers first
  - Magnitude only
- What about signed integers?
  - Sign-magnitude?
  - Two-comlement?
  - Other method?