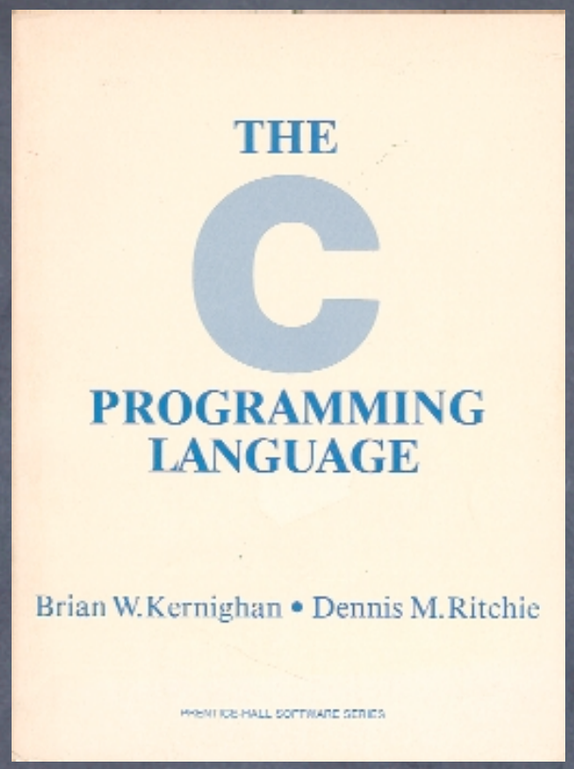
Getting Started with C Programming

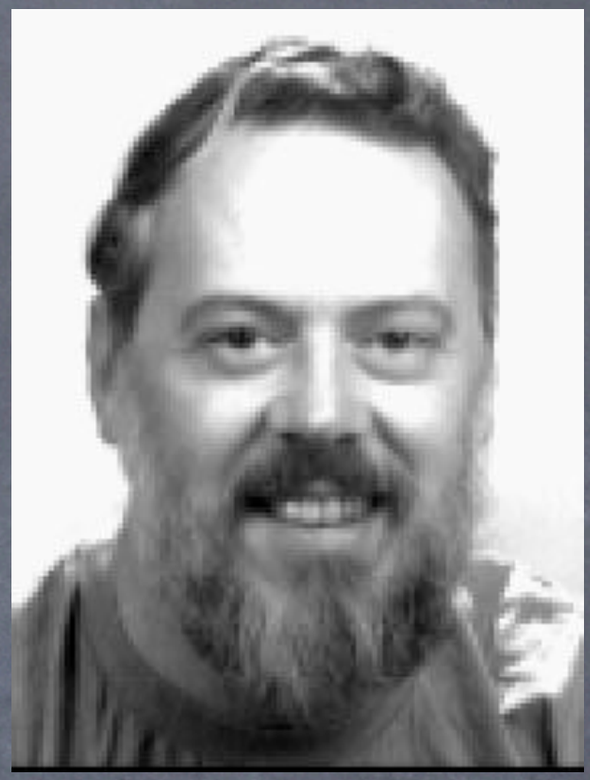
for people who have prior programming experience

About C

- Widely used for systems software, e.g. embedded systems, device drivers, operating systems. Also used for applications.
- An imperative programming language (vs. object oriented, functional/logical)
- Like a "high level assembly language"
- A simple language, but not an easy one!
- http://en.wikipedia.org/wiki/C_programming

Some History





- 1969-1973: Dennis Ritchie creates C at AT&T Bell Labs
- 1973: unix kernel (by Ritchie, Thompson) rewritten in C
- Briefly:

  Fortran > Algol > B > C > C++ > Java > C#
- Thoroughly:

  http://www.levenez.com/lang/history.html

Three C Programs

```
/* Every C program must have a main() method, where execution begins.
 * This is a valid program. It compiles. But when run, it does nothing.
 */
int main() {

}
```

```
/* This program prints "hello, world" */
int main() {
  printf("hello, world");
}
```

```
/* This program also prints "hello, world".
 * But unlike the one above, it compiles without warnings because:
 *     - it includes stdio.h before printf() is called
 *     - the main method properly returns an int
 * Also, its output looks nicer because it prints a newline char.
 */
#include <stdio.h>

int main() {
  printf("hello, world\n");
  return 0;
}
```

Create, Compile, Run

- From the unix shell command line:

```
emacs hello.c&
gcc hello.c
./a.out
```

- Run emacs "in the background" (using the ampersand) so your terminal window is free for compiling. C source file names end with ".c".

- Compile with gcc, the Gnu C compiler. If compilation succeeds, an executable (machine code) file called "a.out" is created.
(Read about Richard Stallman's Free Software Foundation, and Gnu)

- Run the program by typing the executable file name. Depending on your system, you may not need the "./" which means "look in the current directory".