The C Language

Rick Matthews

History

- Von Neumann architecture implemented in Mainframe & Minicomputer hardware
- Pure Binary > Simple Assemblers > Macro-assemblers > sub-routines & functions > high-level languages
- Languages included PL/1, COBOL, ALGOL, FORTRAN, BASIC, etc.
- 1969-73 Ken Thompson & Dennis Ritchie at Bell Labs develop Unix & C Language for PDP7 & PDP11
- 1978 Brian Kernighan & Dennis Ritchie write "The C Programming Language" handbook (K&R) - now ANSI.
- 1993-96 Linus Torvalds develops kernel and console similar to Unix that runs on a PC that he calls "Linux" using C and assembler.

C

- world's 1st (or 2nd?) most popular computer language & has led to languages such as C++, Java, C#, etc.
- a procedural (imperative) language
- compiled language – you iteratively edit, compile, clear any compiler errors, run, clear any run-time errors, distribute.
- one-pass compiler so must define all "names" prior to their use
- produces very fast, efficient code
- nearly all computer languages, including C, and many applications are written in C.
- “make” files for bigger programs or projects
- IDE, Integrated Development Environment, to make development easier, faster.

C Language Features

- allows the inclusion of assembler instructions
- can access hardware and memory directly.
- C introduced pointers, arrays, structures, strong typing
- can link to other languages
- all code is a function or part of a function (ie. no sub-routines), & therefore can't return more than 1 value.
- functions' parameters are always passed by value
- uses standard or user-written includes, libraries (encourages code re-use)
- programs start by passing control to a main() function
- flow control: if, else, while, do while, for(; ;), switch
- types: int, char, long, float, double, pointer, unsigned, void, complex
- strings are character arrays
- zero numbering used

Format

- free form coding
- statements end in ";" (C ignores end-of-lines)
- curly brackets - position are matter of choice
- indenting is for readability only

Symbols

- arithmetic (+, -, *, /, %)
- equality testing (==, !=)
- relations (<, <=, >, >=)
- boolean logic (!, &&, ||)
- bitwise logic (~, &, |, ^)
- bitwise shifts (<<, >>)
- assignment (=, +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=)
- increment and decrement (++, --)
- reference and dereference (&, *, [ ])
- conditional evaluation (? :)
- member selection (. , ->)
- cast or type conversion (( ))
- object size (sizeof)
- sequencing (,)

Some C Features

- naming - be careful of keywords!
- prototypes
- struct, union, enum
- some standard headers - stdio.h, string.h, stdlib.h, math.h
- maths library - needs linking during compilation
- passing parameters from command line - argc, *argv[ ]

Some common standard library functions

- printf()
- scanf()
- fopen()
- fprintf()
- fclose()
- return
- error handling - detection, correction
- goto

Example

```
#include <stdio.h>

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