CMPE 312- OPERATING SYSTEMS

Introduction to C Programming Language

C is a general-purpose programming language used for wide range of applications from Operating systems like Windows and iOS to software that is used for creating 3D movies.

C programming is highly efficient. That's the main reason why it is very popular despite being more than 40 years old.

Newer languages like Python and Java offer more features (garbage collection, dynamic typing) than C programming. However, the performance lowers due to additional processing.

C is a procedural programming language. In procedural languages like C, a list of predefined instructions are carried out step by step. A typical C program may contain one or more procedures (functions) to perform a task.

C is a statically typed language. This means that the type of a variable is checked during the compile time but not in the run-time. This helps in detection of errors during the software development cycle. Also, the statically typed languages are faster than dynamically typed language in general.

C's type system and error checks exist only at compile-time. The compiled code runs in a stripped down run-time model with no safety checks for bad type casts, bad array indices, or bad pointers. There is no garbage collector to manage memory. Instead the programmer manages heap memory manually. All this makes C fast but fragile.

C programming language is not based on OOP like Java, so it is not necessary to solve a given task with use of classes and objects.

Value Types in C

char ASCII character -- at least 8 bits. Pronounced "car". As a practical matter char is basically always a byte which is 8 bits which is enough to store a single ASCII character. 8 bits provides a signed range of -128..127 or an unsigned range is 0..255. char is also required to be the "smallest addressable unit" for the machine -- each byte in memory has its own address.

short Small integer -- at least 16 bits which provides a signed range of -32768..32767. Typical size is 16 bits. Not used so much.

int Default integer -- at least 16 bits, with 32 bits being typical. Defined to be the "most comfortable" size for the computer. If you do not really care about the range for an integer variable, declare it int since that is likely to be an appropriate size (16 or 32 bit) which works well for that machine.

long Large integer -- at least 32 bits. Typical size is 32 bits which gives a signed range of about -2 billion ..+2 billion. Some compilers support "long long" for 64 bit ints.

The integer types can be preceded by the qualifier unsigned which disallows representing negative numbers, but doubles the largest positive number representable. For example, a 16 bit implementation of short can store numbers in the range -32768..32767, while unsigned short can store 0..65535.

float Single precision floating point number typical size: 32 bits

double Double precision floating point number typical size: 64 bits

Constants in the source code such as 3.14 default to type double unless the are suffixed with an 'f' (float) or 'l' (long double).

Compiling & Running C Programs

Example 1: A simple Hello World program in C

```c
/*
* hello.c
* This program prints a welcome message
* to the user.
*/
#include <stdio.h>//for printf

/*
* Import Statements
* C libraries are written with angle brackets.
* Local libraries have quotes:
* #include "lib.h"
*/

int main(){
/*
* Main Function
* entry point for the program
* , should always return an integer (0=success.)
*/
  printf("Hello World!\n");//prints output to the screen.
  return 0;
}
```

In order to compile & run this program, firstly you need a C compiler. Using Terminal on macOS/ Ubuntu you can check by typing **gcc - - version**.

After that, by typing **gcc hello.c -o hello** you may compile and create the executable file, then you just need to type **./hello** to run.

Example 2: A program to handle input/output operations.

```c
#include <stdio.h>

int main(){
  int number;
  printf("Enter a number");
  scanf("%d",&number);
  printf("Number is %d \n",number);

  char name[20];
  printf("Enter name: ");
  scanf("%s", name);
  printf("Your name is %s.", name);

  return 0;
}
```

Exercise

You are requested to create a C program that takes two integer numbers from user, and check the divisibility by 2 of the sum of these two numbers by using conditional statements(if-else statements).