Introduction to C language for CS2100

WXY

August 17, 2024

Contents

1 von Neumann Architecture (Prinstone Architecture)

Von Neumann Architecture describes a computer consisting of:

1. Central Proessing Unit (CPU)
    - Register: special, fast but expensive memory within processor. Used to store temproray result of computation.
    - A control unit containing an instruction register and prgroam counter.
    - An arithmetic/logic unit (ALU)
2. Memory
    - Store both program and data in random-access memory (RAM)
3. I/O Devices

2 Variables in C

A variable is identified by a name (identifier), has a data type and contains a value which could be modified. It also has an physical address in the memory. A variable is declare with a data type. Such as int, float etc. Variables can be initialised at declaration, otherwise would hold an unkonw value (which cannot be assumed to be 0).

Variables in C MUST be declared BEFORE usage. Initialisation may be redundant at times. When initialsation is required but not done so warning will be given by the compiler.

3 Data types in C

C is a stringly typed language, which means data must has data type specified. Type casting in C is allowed, for example:

```
float i = 1.5;
int j = i;
printf("\%d", j);
```

j will be converted to integer data type by truncating the decimal part.

4 Program Structure in C

A C program has 4 main parts:

- Preprocessor directives: libraries used, #define etc.
- Input: through stdin (using scanf), or file input.
- Compute: through arithmetic operations and assignment statements.
- Output: through stdout (using printf), or file output.

Compile C program files using command

```
gcc programFileExample.c -o programFileExample
```

programFileExample will be compiled to binary code in file programFileExample. Without specifying the output file, the code will be compiled by default into a.out.

4.1 Preprocessor Directives

C Preprocessor consists of:

1. Inclusion of header files: libraries used;
2. Macro expansions: defining constants;
3. Conditional compilation

4.1.1 Header Files

To use standard input and output functions such as scanf and printf, <stdio.h> must be included.

```
#include <stdio.h>
```

To use functions in a library, respective header file must be included.

```
#include <math.h>     // Library for mathematical functions
    // must be compiled with -lm to compile
#include <string.h>   // Library for string functions
# include <pthreads.h>
    // must be compiled with -pthreads
```

4.1.2 Marco Expansion

Compiler will do text substitution to replace all marco names in the program with the value declared.

```
#define PI 3.142      // use all CAP for marco
// NOTE: No unintentional semicolon in defining constant, otherwise
#define PI 3.142;
// PI will be substituted by '3.142;' instead.
```

4.2 Input/Output

scanf(format string, input list); printf(format string, print list);

```
int i;
double j;             // declare before use
printf("Enter a number: ");
scanf("%d", &i);      // &: address of operator
                      // the following input will be stored at the
                      // address of i
printf("Enter a decimal number: ");
scanf("%lf", &j);
printf("You entered %d and %f", i, j);
                      // listing order of variables corresponds to
                      // output order

// Note: scanf uses %lf to take floating point input,
//       while printf uses %f to print floating point output

%5d    // an integer with width of 5, right justified
%8.3f  // an float/double with width of 8, with 3 decimal places, right
    justified
```

Placeholder	Variable Type	Function Use
%c	char	printf/scanf
%d	int	printf/scanf
%f	float/double	printf
%f	float	scanf
%lf	double	scanf
%e	float/double	printf(for scientific notation)

Table 1: Format specifiers

Escape sequences	Meaning	Result
\n	New line	subsequent output will be on the next line
\t	Horizontal tab	Move to the next tab position within the same line
\"	Double quote	literal double quote character
##	Percent	literal charater '%'

Table 2: Escape sequences

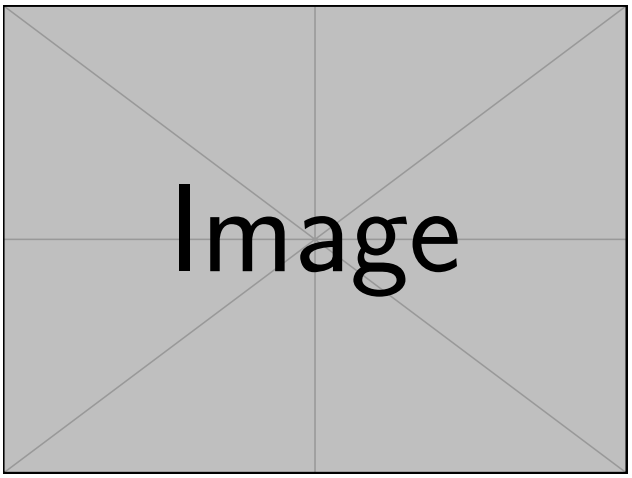

Figure 1: This is a sample caption.