[Revisão de Representaçao Númerica 2](#_Toc158804773)

[Tipos de dados 2](#_Toc1382224434)

[Condicoes 3](#_Toc284282600)

[Interacoes 3](#_Toc1273068601)

[Funcoes 3](#_Toc1106968876)

[Ponteiros 3](#_Toc1332567358)

[Matrizes 4](#_Toc1038094079)

[Listas encadeada 4](#_Toc837899788)

Revisão de Representaçao Númerica

Um número pode ser representado usando uma coleçao finita de símbolos através da técnica de bases: definice uma ordenaçao crescente dos simbolos e cada vez que um numero supera a representaçao atual adiciona-se uma “casa” a esquerda com valor multiplicado pela “base”

Atualmente usamos os algarismos (0,1,...,9) para representação dos números, dessa forma um numero fica representado por uma cadeia de algarismos.

Obs: com temos 10 algarismos nossa base é 10.

Ex: 11 = 1x10+1

21 = 2x10+1

Os computadores usam uma representação em base binaria (1,0), assim o numero 11 m base binaria ficaria

(11)10 =(1011)2 = 1x8+1x2+1 = 1x23+0x22+1x21+1x20

Tipos de dados

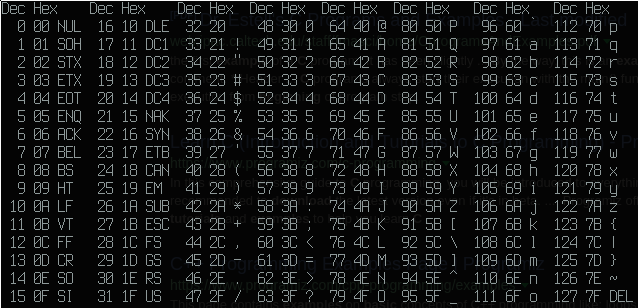
* Int - 4bytes
* Char - 1byte
* String - n \* 1byte
* Float - 4~6 bytes
* Void - 0bytes
* Array - ?bytes
* Struct - ?bytes
* Union - ?bytes
* Enum - ?bytes

INT - O int representa um inteiro, ou seja, um numero decimal que pode ser negativo

É representado por quatro bytes, sendo o ultimo bit (mais significativo) o indicador de sinal, porém os valores sao deslocados de 1, ou seja, o -0 na verdade é o -2^32

Int = |s|0000000|00000000|00000000|00000000|

CHAR - O char por sua vez representa so caracteres, ele segue a tabela ascii



Onde os valores seguem a representação binaria

STRING - A string é um ARRAY de caracteres terminada em ‘\0’ (zero)

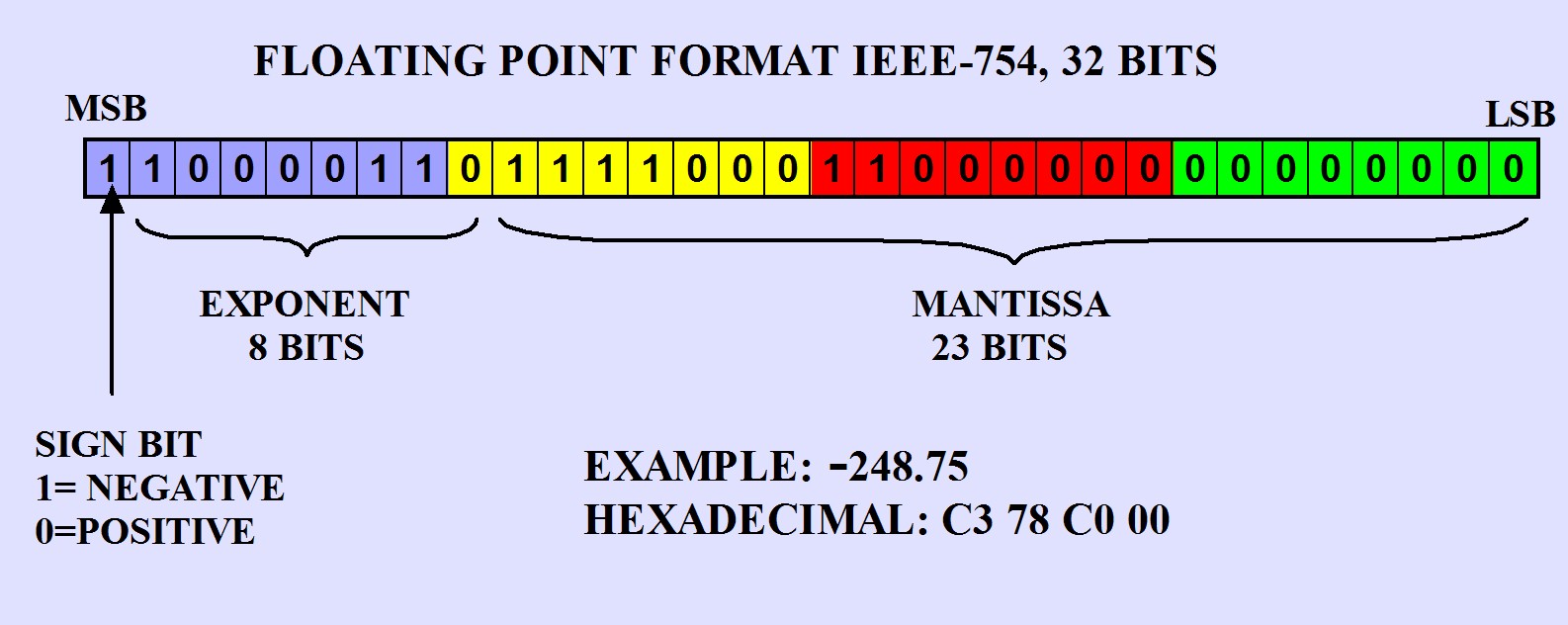
FLOAT - O float é o ponto flutuante, ele usa uma representação um pouco diferente: é separado em sinal, mantissa e expoente

Matematicamente fica assim

Numero = (sinal)x(mantissa)expoente

Onde a mantissa nao pode superar 1

Em termos de armazenamento de memoria fica asssim



ARRAY - O array é um vetor de elementos, podendo ser de qualquer tipo. Os elementos do array pode ser acessador pelo indice começando de 0:

Elemento 1 = nome\_do\_array[0]

Elemento 10 = nome\_do\_array[9]

O nome do array representa o endereço para seu primeiro elemento (na parte de ponteiros isso vai ficar mais claro)

STRUCT - O struct é uma estrutura, ele é usado para agregar varios tipos e criar um novo

Exemplo:

Criando

struct pessoa {

char nome[50];

int idade;

};

Declarando

struct pessoa ana;

Usando

ana.nome=”ana crisna ferreira borges”’

ana.idade = 20;

UNION - O union é uma união de elementos, ele é usado quando nao se tem certeza do tipo de dado que vai se alocado uma variável (depois dou um exemplo, usei pouco isso)

ENUM - O enum é uma enumeração, é usado para representar uma lista de possibilidades sem precisar definir um valor para elas,

Enum opcoes {

Opcao\_1,

Opcao\_2

}

enum op;

Op = Opcao\_1;

Op = Opcao\_2;

Condicoes

IF else

Para se controlar o fluxo de execuçao de um programa pode se usar alguns condicionantes logicos, assim para determinados valores o programa segue um caminho ou outro

Exemplo:

if (ana.idade > 30 )

printf (“Ana que ta morta\n”);

else

printf(“Ana ta deboas\n);

No exemplo acima o programa pode ir para uma parte ou para outra dependendo do valor de ana.idade

Alguns comparadores lógicos de C são:

A > B , A maior que B (retorna verdadeiro caso A seja maior que B)

A >= B, maior ou igual que

A < B, menor que

A <= B, menor ou igual que

A == B, igual que

Obs: as expreçoes logicas tem um valor numerico na memoria, assim podesse usar um valor no lugar de uma expressao

if (numero)

printf(“numero e diferente de zero\n”);

else

printf(“numero é igual a zero\n”);

faz a mesma coisa que o codigo anterior

Switch case

Interacoes

For

While

Goto

Funcoes

Argumentos e formas de declaracao

Ponteiros

Simples

Matrizes

Examplos de calculo com matrizes

Passando matrizes para funcoes

Listas encadeada