

Armazenamento de Dados



Universidade Federal do Ceará - Campus Quixadá

Roberto Cabral
rbcabral@ufc.br

26 de Novembro de 2020

Arquitetura e Organização de Computadores II

Representação numérica

Sintaxe **C/C++** para especificar números

	Prefix	Example	32-bit pattern (binary)
Decimal	<u>(none)</u>	123	0000 0000 0000 0000 0000 0000 0111 1011
Hexadecimal	<u>0x</u>	0x123	0000 0000 0000 0000 0000 0001 0010 0011
Octal	<u>0</u>	0123	00 000 000 000 000 000 000 000 001 010 011

Memória

- Existem dois tipos gerais de memória usados para armazenar instruções e dados do programa em um computador.
 - Memória de acesso aleatório (RAM) - Uma vez que um bit é definido como zero ou um, ele permanece nesse estado até que a unidade de controle o altere ativamente ou a energia seja desligada.
 - Memória de somente leitura (ROM) - Os bits são definidos permanentemente. A unidade de controle pode ler o estado de cada bit, mas não pode alterá-lo. Alguns tipos de ROM podem ser reprogramados com hardware especializado, mas os bits permanecem no novo estado quando a energia é desligada.

Endereço de Memória

- Cada byte na memória é numerado e chamado de **endereço de memória**.
- O endereço de memória de um determinado byte nunca muda, mas seu conteúdo pode ser modificado.
- Normalmente expressamos os endereços de memória em hexadecimal.

CHAR M[100]

M[73]

A handwritten diagram in green ink. The top part shows the text 'CHAR M[100]' followed by a closing square bracket. A horizontal line is drawn underneath. Below the line, the text 'M[73]' is written. The '73' is circled, and a green arrow points from the bottom of the circle to the right.

Endereço de Memória

Possível conteúdo dos primeiros 16 bytes da memória

<u>00000000:</u> 6a	00000008: f0
00000001: f0	00000009: 02
00000002: 5e	0000000a: 33
00000003: 00	0000000b: 3c
00000004: ff	0000000c: c3
00000005: 51	0000000d: 3c
00000006: cf	0000000e: 55
00000007: 18	0000000f: aa

Quantos bits são necessários para armazenar um determinado dado?

- Determinar se um aluno foi aprovado ou reprovado?
- Opções de uma questão de múltipla escolha?
- Armazenar os alunos da UEC?

A 00

B 01

C 10

D 11

E

2^x

2^n

Usando C para explorar o formato dos dados

```
/* intAndFloat.c
 * Using printf to display an integer and a float.
 * 2017-09-29: Bob Plantz
 */
#include <stdio.h>

int main(void)
{
    int anInt = 19088743;
    float aFloat = 19088.743;

    printf("The integer is %d and the float is %f\n", anInt, aFloat);

    return 0;
}
```

Usando C para explorar o formato dos dados

```
int main(void)
{
    unsigned int unsignedInteger;
    unsigned int bitPattern;

    printf("Enter an unsigned decimal integer: ");
    scanf("%u", &unsignedInteger);

    printf("Enter a bit pattern in hexadecimal: ");
    scanf("%x", &bitPattern);

    printf("%u is stored as %#010x, and\n", unsignedInteger, unsignedInteger);
    printf("%#010x represents the unsigned decimal integer %u\n",
           bitPattern, bitPattern);

    return 0;
```

Usando C para explorar o formato dos dados

- O programa mostra que hexadecimal é usado como uma conveniência humana para definir padrões de bits.
- Um número não é inerentemente binário, decimal ou hexadecimal.
- Um determinado valor pode ser expresso de maneira precisamente equivalente em cada uma dessas três bases numéricas.
- pode ser expresso de forma equivalente em qualquer base numérica, mas os números são armazenados em binários no computador.

Examinando a memória com o GDB

- O GDB (GNU Project Debugger) é uma ferramenta para:
 - observar um programa enquanto este executa
 - ver o estado no momento que a execução falha
- Permite:
 - iniciar a execução de um programa
 - executar linha-a-linha
 - especificar pontos de paragem
 - imprimir valores de variáveis

GDB

- O gdb possui um grande número de comandos.
- A seguir temos os comandos iniciais:
 - `br source-filename:line-number` - define um ponto de interrupção na linha do arquivo de origem especificada.
 - `cont` - Continuar a execução do programa a partir do local atual.
 - `help comando` — mostra como utilizar um comando.
 - `i r` - mostra o conteúdo dos registradores.
 - `li LineNumber` - lista dez linhas do código-fonte, centralizado no número da linha especificado por `LineNumber`.
 - `print expression` - calcula e imprime o valor da expressão.
 - `printf` - segue o mesmo formato do C.
 - `r` - comece a execução de um programa que foi carregado sob o controle do gdb.
 - `x/nfs MemoryAddress` - Exibir (examinar) *n* valores na memória no formato *f* de tamanho *s* começando em *MemoryAddress*.

Armazenamento de Dados



Universidade Federal do Ceará - Campus Quixadá

Roberto Cabral
rbcabral@ufc.br

26 de Novembro de 2020

Arquitetura e Organização de Computadores II