Estruturas de Dados

Algoritmos e Programação de Computadores

Guilherme N. Ramos
gnramos@unb.br

2015/2

Registros

Registro

Estrutura que armazena diferentes tipos de dados em uma única variável.

```
1  Algoritmo LeFuncionários
2  Definições
3      funcionario : registro (nome, endereço : string;
4                                        sexo : caractere;
5                                      código : inteiro;
6                                     salário : real)
7  Variáveis
8      funcionários : vetor[1000] de funcionario
9  Início
10     /* ... */
11     Para i de 0 a 999 Faça
12         Leia(funcionários[i])
13     FimPara
14     /* ... */
15 Fim
```


Registros

Na linguagem `C`, o registro é definido pela palavra-chave `struct`, e o acesso a seus componentes pelo identificador e o caractere '`.`'.

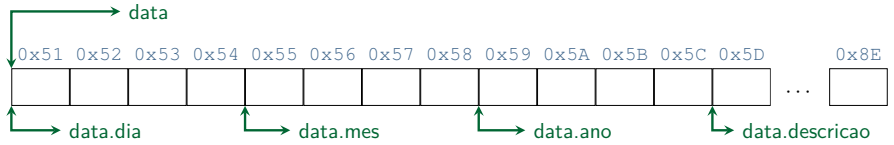
0-data.c

```
1      struct {
2          int dia, mes, ano;
3          char descricao[50];
4      } data;
5
6      leia_string("Digite a descrição: ", data.descricao);
7      data.ano = leia_int("Digite o ano: ");
8      data.mes = leia_int("Digite o mes: ");
9      data.dia = leia_int("Digite o dia: ");
10
11     printf("%s:\n%02d/%02d/%04d\n", data.descricao,
12                                     data.dia,
13                                     data.mes,
14                                     data.ano);
15
```


Registros

0-data.c

```
1      struct {
2          int dia, mes, ano;
3          char descricao[50];
4      } data;
```



Registros

3-mp3.c

```c
/**      @file: 3-mp3.c
 *     @author: Guilherme N. Ramos (gnramos@unb.br)
 * @disciplina: Algoritmos e Programação de Computadores
 *
 * Exemplo de uso de registro (ID3v1) para armazenar as
 * informações de um arquivo no formato MP3. Veja mais em:
 * http://en.wikipedia.org/wiki/ID3#ID3v1 */

typedef struct{
   char header[3];
   char titulo[30];
   char artista[30];
   char album[30];
   char ano[4];
   char comentario[30];
   unsigned char genero;
} mp3_ID3v1;
```


Binários

É muito fácil manipular arquivos binários, mas os procedimentos de leitura não podem ser dissociados dos de escrita (e vice-versa).

Binários

O computador trabalhar apenas com bit e bytes, portanto todos os arquivos são conjuntos binários.

A manipulação é extremamente simples, tem-se o endereço do arquivo, basta ler/escrever a quantidade de bytes desejada.

Pseudo-código

```
Função Void Leia(arquivo origem, tipo destino)
Função Void Escreva(arquivo destino, tipo origem)
```

Linguagem C

```c
size_t fread(void *destino, size_t tam, size_t qte, FILE *origem);
size_t fwrite(void *origem, size_t tam, size_t qte, FILE *destino);
```


Texto

Humanos não se comunicam por bytes...

```c
int fprintf(FILE *fp, const char *formato, ... );
int fscanf(FILE *fp, const char *formato, ... );
int fputc(int caractere, FILE *fp );
int fgetc(FILE *fp);
int fputs(const char *string, FILE *fp );
char *fgets(char *string, int num_caracteres, FILE *fp );
```

Cor

Um padrão comum de representação de cor é o sistema RGB, em que cada cor é composta pelos três componentes (*Red* - *Green* - *Blue*).

Cada componente tem um valor definido por 1 byte indicando a intensidade: `0xRRGGBB`

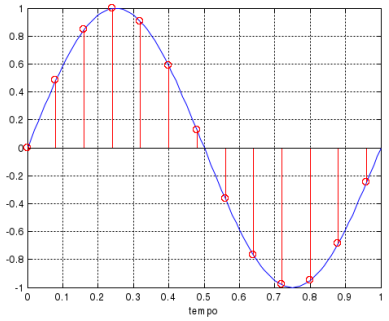
(ausência da cor) `00` $\Leftrightarrow$ `FF` (intensidade máxima)

São, portanto, $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16,777,216$ cores possíves.

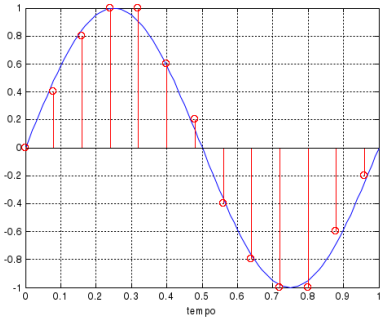
`0xFF0000`	vermelho	`0x000000`	preto
`0x00FF00`	verde	`0xFFFFFF`	branco
`0x0000FF`	azul	`0xFFFF00`	amarelo

Áudio

- 44.1*kHz*
- 16 bits
- Estéreo (2 canais)



Amostragem



Quantização