

Modos de abertura para arquivos binários

Modo de abertura	Arquivo Existe	Arquivo não existe	Permissão de leitura	Permissão de escrita
rb	ok	erro	x	
wb	cria novo, antigo eliminado	cria		x
r+b ou rb+	ok	erro	x	x
w+b ou wb+	cria novo, antigo eliminado	cria	x	x
ab	novos dados acrescentados ao final	cria		x
a+b ou ab+	novos dados acrescentados ao final	cria	x	x

Obs: os modos r+b, w+b e a+b também podem ser escritos rb+, wb+ e ab+.

Slide 1

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Modos de abertura para arquivos texto

Modo de abertura	Arquivo Existe	Arquivo não existe	Permissão de leitura	Permissão de escrita
r	ok	erro	x	
w	cria novo, antigo eliminado	cria		x
r+	ok	erro	x	x
w+	cria novo, antigo eliminado	cria	x	x
a	novos dados acrescentados ao final	cria		x
a+	novos dados acrescentados ao final	cria	x	x

Slide 2

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Exemplo 1

Desenvolver um programa em C que:

- crie um arquivo binário, através de uma função, contendo as seguintes informações de produtos de uma quitanda:
nome (string)
valor (float)
comentário (string)
- a seguir, baseado no conteúdo do arquivo acima e através de uma função, gere um arquivo texto onde cada linha apresenta o número de ordem sequencial do produto correspondente no arquivo binário, seguido do conteúdo do campo <nome>, hífen e conteúdo do campo <comentário>.
- antes de encerrar, novamente através de função, imprima o arquivo texto gerado.

Slide 3

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

A execução do programa deve gerar o conteúdo da tela abaixo:

```
Nome do produto com ate 31 caracteres: batata
Valor do produto: 2.32
Comentario com ate 51 caracteres: por kilo, fornecedor Antonio
1 - NovoProduto: 0 - EncerrarInclusao
1
Nome do produto com ate 31 caracteres: alface
Valor do produto: .07
Comentario com ate 51 caracteres: por unidade, COBOL
1 - NovoProduto: 0 - EncerrarInclusao
1
Nome do produto com ate 31 caracteres: feijao
Valor do produto: 2.32
Comentario com ate 51 caracteres: por kilo, Sogenalda
1 - NovoProduto: 0 - EncerrarInclusao
0
Produtos gravados = 3
Linhas escritas no arquivo texto = 3
produto 1 : batata - por kilo, fornecedor Antonio
produto 2 : alface - por unidade, COBOL
produto 3 : feijao - por kilo, Sogenalda
Programa concluido com sucesso!
Pressione qualquer tecla para continuar. . .
```

Slide 4

4

UFRGS Informática

Exemplo 1

```
/* Gera um arquivo binário, com estruturas com informações de produtos de
uma quitanda: nome do produto (string)
valor (float)
comentario (string)
Gera um arquivo texto em que cada linha inicia pelo número correspondente à
sequência do produto no arquivo, seguido do conteúdo do campo nome do
produto, hífen, e conteúdo do campo comentário. */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXNOME 31
#define MAXCOMENT 51
FILE *arqbin;
FILE *arqtxt;
struct prod
{
    char nome[MAXNOME];
    float valor;
    char coment[MAXCOMENT];
}
int gravarqbin (FILE *, struct prod, int, int); //protótipo, arquivo como parâmetro
int gerarqtexto (FILE *, FILE *, struct prod);
void listarqtexto(FILE *);
...
```

Slide 5

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
int main ( )
{
    system("color 70");
    int cont_reg; //retornado pela função, informa nro de registros gravados
    struct prod produto; // var buffer global
    if ((arqbin = fopen("c:\\prod.bin", "wb")) == NULL)
        printf("\nErro abertura arquivo binario - criacao\n\n");
    else
    {
        printf("\nProdutos gravados = %d\n\n",
            cont_reg = gravarqbin(arqbin, produto, MAXNOME, MAXCOMENT));
        fclose(arqbin);
        if (cont_reg < 1)
            printf("\nArquivo binário está vazio! ");
        else
        {
            if ((arqbin = fopen("c:\\prod.bin", "rb")) == NULL)
                printf("\nErro abertura arquivo binario - leitura\n\n");
            else
            {
                if ((arqtxt = fopen("c:\\prod.txt", "w")) == NULL)
                    printf("\nErro abertura arquivo texto - criacao\n\n");
                else
                {
                    // próximo slide
```

Gravação do
arquivo binário

Slide 6

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```

...
else
{
    printf("\nLinhas escritas no arquivo texto = %d\n",
    gerarqtexto(arqbin, arqtxt, produto));
    fclose(arqtxt);
    if ((arqtxt = fopen("c:\\prod.txt", "r")) == NULL)
    {
        printf("\nErro abertura arquivo texto (leitura)\n");
    }
    else
    {
        listarqtexto(arqtxt);
        printf("\nPrograma concluído com sucesso!\n\n");
        fclose(arqtxt);
    }
}
}
system("pause");
return 0;
}
// funções nos próximos slides

```

Gravação do
arquivo texto

Listagem do
arquivo texto

Slide 7

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```

// criação do arquivo binário, retornando número de registros gravados:
int gravarqbin(FILE * arqb, struct prod produt, int maxnom, int maxcoment)
{
    int contgrav = 0, op;
    do
    {
        // leitura dos campos, com consistência
        {
            printf("\nNome do produto com ate %d caracteres: ", maxnom);
            fflush(stdin);
            gets(produt.nome); // usa gets, para não incluir \n...
            fflush(stdin);
        }
        while (strlen(produt.nome) >= maxnom );
        do
        {
            printf("\nValor do produto: ");
            scanf("%f", &produt.valor);
            while (produt.valor <= 0);
        }
        do
        {
            printf("\nComentario com ate %d caracteres: ", maxcoment);
            fflush(stdin);
            gets(produt.coment); // idem
            fflush(stdin);
        }
        while (strlen(produt.coment) >= maxcoment );
        fwrite(&produt, sizeof(produt), 1, arqb);
        contgrav++; // conta registros gravados
        printf("\n1 - NovoProduto; 0 - EncerrarInclusao\n");
        scanf("%d", &op);
    } while (op);
    return contgrav; // encerra função, retornando número de registros gravados
}
...

```

Gravação no
arquivo binário

Slide 8

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```

// conversão para arquivo texto
int gerarqtexto(FILE * arqbin, FILE * arqt, struct prod produt)
{
    int cont = 0;
    while (!feof(arqbin))
    {
        if (fread(&produt, sizeof(produt), 1, arqbin) == 1)
        {
            cont++;
            // imprime arquivo txt, inserindo caracteres adicionais no fprintf:
            fprintf(arqt, "produto %d : %s - %s\n",
            cont, produt.nome, produt.coment);
        }
    }
    return cont; // retorna número de linhas gravadas
}
void listarqtexto(FILE * arqt) // impressão caractere à caractere
{
    char caract;
    while (!feof(arqt))
    {
        printf("%c", getc(arqt));
    }
}

```

Gravação em
arquivo texto
com fprintf

Slide 9

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Exemplo 2

Desenvolver um programa em C contendo uma função void **ConvertePontuacao**(FILE *p_out, int vet_ptos[]), que abre um arquivo texto "out.txt", composto de seqüências de até 3 dígitos numéricos sucedidos por um asterisco, sucessivamente (até encontrar EOF), e converte cada seqüência em um número inteiro, o qual é armazenado no vetor parâmetro (inserir -1 no elemento do vetor que sucede o último valor convertido, como sinalizador de elemento final), fecha o arquivo e devolve o vetor preenchido.

Slide 10

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```

// converte txt em vetor de inteiros
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
FILE *p_out;
void ConvertePontuacao( FILE *p_out, int vet_ptos[])
{
    char caractere = ' ', str_aux[4]; // armazena dígitos para conversão
    int i_vet=0, i_aux=0;
    if ((p_out = fopen("out.txt", "r")) == NULL)
    {
        while (caractere != EOF) // foi inicializado com branco!
        {
            caractere = getc(p_out); // lê caractere
            switch (caractere)
            {
                case '*':
                case EOF: break;
                default:
            }
            vet_ptos[i_vet] = -1; // sinalizador de fim de valores convertidos
            fclose(p_out);
        }
    }
    else
    {
        printf("\nNao encontrou o arquivo!\n");
    }
}
// fim da função

```

próximo
slide

```

.....
while (caractere != EOF) // foi inicializado com branco!
{
    caractere = getc(p_out); // lê caractere
    switch (caractere)
    {
        case '*': // indica fim de seqüência de números
            str_aux[i_aux] = '\0'; // insere finalizador de string
            printf("\nString gerado: %s", str_aux);
            vet_ptos[i_vet] = atoi(str_aux); // converte string para inteiro
            printf("\nNro convertido = %d", vet_ptos[i_vet]);
            i_aux = 0; // prepara para reiniciar o preenchimento
            i_vet++; // posiciona no próximo elemento do vetor
            break;
        case EOF: break;
        default: str_aux[i_aux] = caractere;
            i_aux++;
    }
    vet_ptos[i_vet] = -1; // sinalizador de fim de valores convertidos
    fclose(p_out);
}
else
{
    printf("\nNao encontrou o arquivo!\n");
}
}

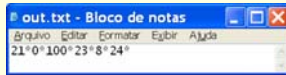
```

Slide 12

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

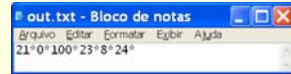
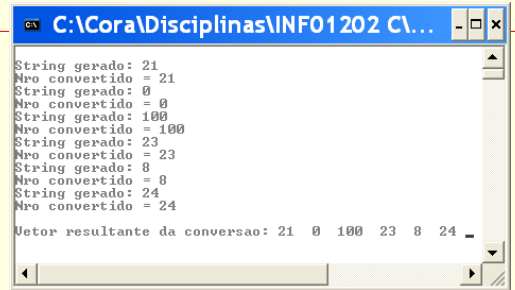
```
int main()
{
    int ptos[50], i_p=0; // vetor de pontuação e índice
    system("color F1");
    ConvertePontuacao(p_out, ptos);
    printf("\n\nVetor resultante da conversao:\n");
    while (ptos[i_p] != -1)
    {
        printf(" %d ", ptos[i_p]);
        i_p++;
    }
    system("pause>>null");
    return 0;
}
```



Slide 13

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática



Slide 14

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Exemplo 3

Desenvolver um programa em C contendo uma função void `ConvertePontuacao(FILE *p_out, int vet_ptos[])`, que abre um arquivo texto "textoout.txt", composto de linhas com seqüências de até 3 dígitos numéricos sucedidos por um asterisco, sucessivamente (até encontrar `\n` e EOF), e converte cada seqüência em um número inteiro, o qual é armazenado no vetor parâmetro (inserir -1 no elemento do vetor que sucede o último valor convertido, como sinalizador de elemento final), fecha o arquivo e devolve o vetor preenchido.

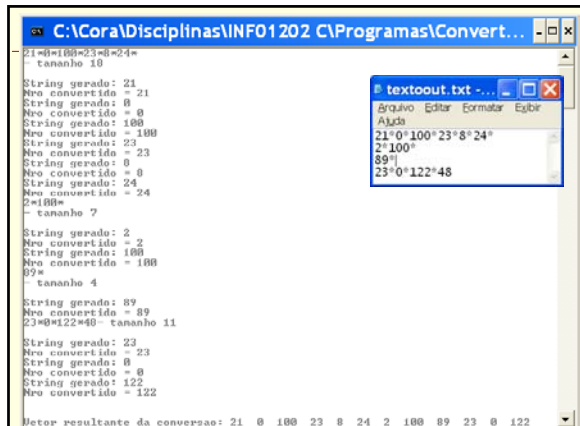
```
// converte.txt em vetor de inteiros
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
FILE *p_out;
void ConvertePontuacao( FILE *p_out, int vet_ptos[])
{
    char caractere = ' ', str_aux[4]; // armazena dígitos para conversão
    char buf_txt[256];
    int i_vet=0, i_aux=0, i_t;
    if ((p_out = fopen("textoout.txt", "a")) != NULL)
    {
        while(!feof(p_out))
        {
            if (fgetc(buf_txt) == '\n')
            {
                printf("Chegou ao fim do arquivo");
                break;
            }
            else
            {
                printf("%s- tamanho %d\n", buf_txt, strlen(buf_txt)); // imprime na tela
                for (i_t=0; i_t<strlen(buf_txt); i_t++)
                {
                    switch (buf_txt[i_t])
                    {
                        // ... próximo slide
                    }
                }
                i_aux = 0; // prepara para reiniciar o preenchimento
                printf("\n");
            }
        }
        vet_ptos[i_vet] = -1; // sinalizador de fim de valores convertidos
        fclose(p_out);
    }
    else
        printf("\nNao encontrou o arquivo\n");
} // fim da função
```

Slide 15

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
...
int main()
{
    int ptos[50], i_p=0; // vetor de pontuação e índice
    system("color F8");
    ConvertePontuacao(p_out, ptos);
    printf("\n\nVetor resultante da conversao:");
    system("pause>>null");
    while (ptos[i_p] != -1)
    {
        printf(" %d ", ptos[i_p]);
        i_p++;
    }
    system("pause>>null");
    return 0;
}
```



Slide 17

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Slide 18

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Função fgets

- Permite a leitura de uma linha do arquivo como string.
- Formato:

```
fgets(vetorchar, sizevetor, point_arq );
```

vetorchar : onde o conteúdo da linha será armazenada
sizevetor : tamanho do vetor **vetorchar**
point_arq : arquivo de onde será feita a leitura

- Exemplo:

```
fgets(buffertxt,MAX,arq); // string lida de arq é armazenada no vetor
```

Slide 19

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Função strtok

- Retorna a *string* correspondente ao primeiro conjunto de caracteres que antecede o separador informado.

- Formato:

```
strtok(vetorchar, charseparador );
```

vetorchar : onde ocorre a busca
charseparador : símbolo separador

- Exemplo:

```
strtok(buffertxt, ','); // extrai token (substring) até vírgula
```

Slide 20

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Função strtok

- Exemplo: extrai substrings sucessivos

```
// especifica o string inicial só na primeira extração:
```

```
sub1= strtok(stringinicial, ",");  
sub2= strtok(NULL, ",");  
sub3= strtok(NULL, ",");
```

- O nome do vetor é usado na função **strtok** apenas para extrair a primeira *substring*. Para a extração das demais *substrings* de uma mesma *string*, deve-se usar a constante **NULL** como primeiro parâmetro das chamadas subsequentes de **strtok**, uma vez que enquanto a *string* base for a mesma, **strtok** segue avançando pelas posições da *string* base.

Slide 21

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Função atoi

- Transforma o conteúdo de um vetor de caracteres em inteiro.

- Formato:

```
atoi(vetorchar);
```

vetorchar : conteúdo a ser convertido

- Exemplo:

```
num1 = atoi(strtok(buf, ","));  
substring = strtok(NULL, ",");  
num2 = atoi(substring1);
```

Slide 22

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Função atof

- Transforma o conteúdo de um vetor de caracteres em real.
- Formato:

```
atof(vetorchar);
```

vetorchar : conteúdo a ser convertido

- Exemplo:

```
real1 = atof(strtok(buf, ","));  
substring = strtok(NULL, ",");  
real2 = atof(substring1);
```

Slide 23

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Os dados no arquivo devem estar separados uns dos outros por delimitadores (caracteres) diferentes do branco.

Por exemplo: , (vírgula) # /

Sejam os seguintes exemplos de possíveis linhas de arquivos:

Exemplo1:

```
2,Maria da Silva,8,4,5,7,8,8,  
5,José Ferreira,9,0,5,8,7,5,  
14,Agnes Lindemann,8,5,7,0,6,
```

Exemplo2:

```
20#Ricardo Leão#8,4,5,7,8,8#  
55#João Pedreira#9,0,5,8,7,5#  
67#Dagmar Scherer#8,5,7,0,6,0#
```

Slide 24

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Extração de elementos variados de linhas de um arquivo texto

Passos:

1. Ler toda a linha do arquivo como uma *string* (função **fgets**).
2. Extrair da *string* lida *substrings* que correspondam aos dados (função **strtok**).
3. Converter cada *substring* para o tipo de dado correspondente (funções **atoi** e **atof**).

Slide 25

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Exemplo

Desenvolver um programa em *C* que leia um arquivo texto contendo nomes e notas, listando os conteúdos formatados e calculando a média das notas lidas.

O programa ao executar lê o arquivo dados,

dados - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

C:\Magda\Programas\Prog\Arquivos\p34c.exe

C:\>listar_notas

C:\>in

>L

NUM	NOME	N1	N2
004	Mariana Bastos	8.5	4.8
013	Joao da Silva	7.5	6.1
015	Peter Johannes	5.0	4.4
021	Vilmar Carvalho	7.2	7.1
030	Sonja Sperb	5.9	6.2

Media das notas = 6.3

Pressione qualquer tecla para continuar. . .

produzindo a saída:

Slide 26

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h> // necessário para strtok (
#define MAX 80
char opcao_menu( ) // menu com as operações
{
    system("CLS");
    system("color, 70");
    printf("(L)istar notas\n");
    printf("(F)im \n");
    printf(">");
    return((getchar()));
}

void listar_notas( )
{
    int num, notas;
    float n1,n2,media;
    char *nome;
    char buf[MAX];
    FILE *arq;
    arq = fopen("dados.txt","r"); // abrir arq com opção só de leitura: r
    if (arq==NULL)
        printf("Erro ao abrir arquivo \n");
    else
    { // imprime cabeçalho
        printf("\n");
        printf("NUM | NOME | N1 | N2 \n");
        printf("-----+-----+-----+-----\n");
        .... // próximo slide
    }
```

tica

```
...
notas=0;
media=0;
fgets(buf,MAX,arq); // lê arq string de Max caracteres e armazena em buf
while ((feof(arq)) // enquanto não for fim de arquivo.....
{
    num= atoi(strtok(buf," ")); // transforma string de buf em int
    nome= strtok(NULL," ");
    n1=atof( strtok(NULL," ")); // transforma string de buf em float
    n2=atof( strtok(NULL," "));
    printf("%03d | %20s | %4.1f | %4.1f \n",num, nome,n1,n2);
    notas=notas +2;
    media =media + n1 + n2;
    fgets(buf,MAX,arq); // lê próxima linha
}
printf("-----+-----+-----+-----\n");
media=media/notas;
printf("Media das notas = %4.1f\n", media);
fclose(arq);
}

int main()
{
    char op;
    op=opcao_menu( ); // função retorna a opção
    if (op == 'L')
        listar_notas(); // chama listagem de notas
    printf("\nPrograma sera encerrado!\n");
    system("pause");
    return 0;
}
```

ica

Execução:

C:\backcupcida\LinguagemCPagina20081\ARQUIVOSTEXTO\codig

<L>istar notas

<F>im

>L

NUM	NOME	N1	N2
004	Mariana Bastos	8.5	4.8
013	Joao da Silva	7.5	6.1
015	Peter Johannes	5.0	4.4
021	Vilmar Carvalho	7.2	7.1

Media das notas = 6.3

Pressione qualquer tecla para continuar. . .

Slide 29

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática