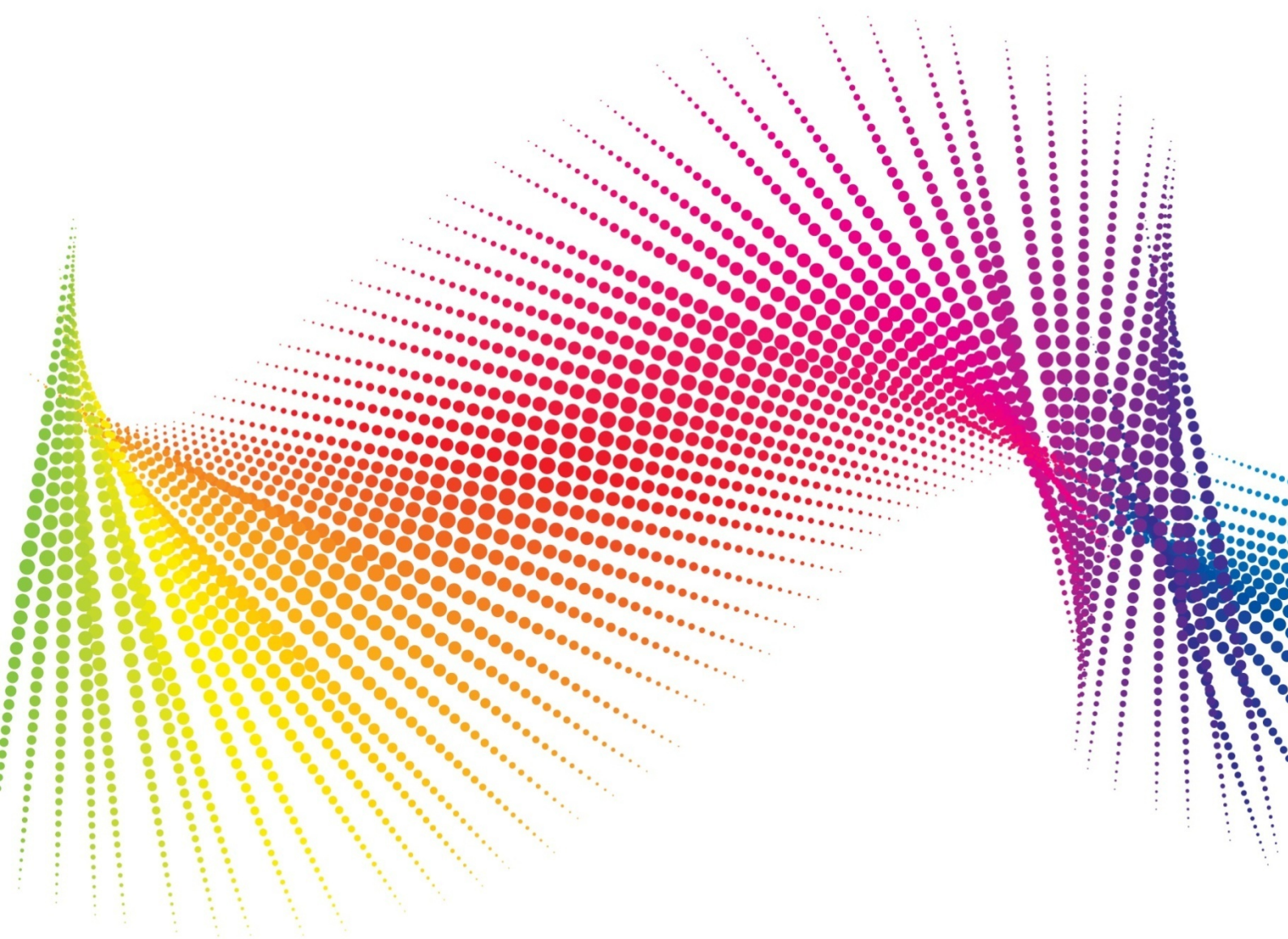


Estrutura de Dados

Aula 16



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

Aula 16: Funções para manipulação de arquivos

Objetivo: Continuar estudando os arquivos, apresentando mais funções para sua manipulação.

Caro aluno, a linguagem C oferece diversas funções para manipulação de arquivos: `fopen()`, `fclose()`, `feof()`, `ferror()`, `rewind()`, `remove()`, `fflush` e `fseek()`. Além destas, existem as funções específicas para leitura e escrita em arquivos: `fgetc()`, `fputc()`, `fputs()`, `fgets()`, `fwrite()`, `fread()`, `fprintf()` e `fscanf()`. Na nossa disciplina não temos o objetivo de estudar todas elas, dessa forma, vamos apresentar as que são mais úteis ao nosso curso, ok? Recomendo o livro do Schildt (1997) para estudar aquelas que não forem tratadas no nosso curso, caso você queira ampliar seus estudos.

A função `feof()`

Geralmente não sabemos quantos dados estão armazenados num arquivo existente. Assim, para lermos todos os seus dados, podemos contar com a ajuda da função `feof()` que sinaliza o fim do arquivo ou *end of file*. Vejamos as características da função `feof()`:

- Arquivo cabeçalho a ser incluído: **<stdio.h>**.
- Determina quando o final de arquivo foi atingido na leitura de dados de arquivos texto e binários. Na leitura de um arquivo binário, caso um valor inteiro igual ao de **EOF** seja lido, indicaria uma condição de fim de arquivo, apesar do final físico não ter sido alcançado, por isso devemos ter cuidado com esta situação indesejada.
 - Recebe como parâmetro um ponteiro para arquivo.
 - Devolve verdadeiro se o final de arquivo tiver sido atingido, e 0, caso contrário.

A função fprintf()

A função fprintf() permite a gravação de dados num arquivo. Vejamos as características da função fprintf():

- Arquivo cabeçalho a ser incluído: **<stdio.h>**.
- Comporta-se da mesma forma que **printf()**, mas escrevendo não na tela, mas em arquivos.
- Seu primeiro parâmetro é um ponteiro para arquivo, sendo os demais argumentos iguais aos de **printf()**.

A função fscanf()

A função fscanf() permite a leitura de dados de um arquivo. Vejamos as características da função fscanf():

- Arquivo cabeçalho a ser incluído: **<stdio.h>**.
- Comporta-se da mesma forma que **scanf()**, mas lendo não do teclado, mas de arquivos.
- Seu primeiro parâmetro é um ponteiro para arquivo, sendo os demais argumentos como os de **scanf()**.

Exemplo de programa que cria e lê dados de um arquivo texto usando feof(), fprintf() e fscanf()

```
/* File2.c - Ilustra o uso de feof, fprintf e fscanf */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
FILE *arq;
char nome[30];
int idade, cont=0;
float altura;
```

```
int main()
{   arq = fopen("dados.dat", "w"); // cria o arquivo
    if (arq == NULL)
    {   printf("\nErro na criacao do arquivo dados.dat");
        getch();
        exit(1);
    }
    printf("\nTodos os dados serao gravados no arquivo dados.dat\n");
    for (;;)
    {   printf("\n Digite um nome (. para finalizar):  ");
        scanf("%s",nome);
        if (strcmp(nome, ".")==0)
            break;
        printf("\n Idade:  ");
        scanf("%i",&idade);
        printf("\n Altura:  ");
        scanf("%f",&altura);
        fprintf(arq,"%s %i %.2f \n",nome,idade,altura);
    }
    fclose(arq);

    printf("\n\nVamos abrir o arquivo e ler os dados gravados\n");
    arq = fopen("dados.dat", "r");
    if (arq==NULL)
    {   printf("\nErro na abertura do arquivo dados.dat");
        getch();
        exit(1);
    }

    while (!feof(arq))
    {   fscanf(arq,"%s %i %f \n",nome,&idade,&altura);
        printf("\n Nome=%s Idade=%i Altura=%.2f \n",
                nome,idade,altura);
        ++cont;
    }

    fclose(arq);

    printf("\n\nArquivo com %i registros. Fechando o arquivo.\n",cont);

    puts("\nGoodbye!");
    getch();
    return 0;
}
```

Caro aluno, um exercício interessante agora é implementar este programa para, em seguida, ir ao diretório corrente e verificar se o arquivo criado está realmente lá com os dados que você forneceu.

A função fputs()

A função fputs() permite que se grave strings em um arquivo.

Exemplo:

```
char str[80]="Todos somos iguais perante a lei.";
FILE *fp;
...
fputs(str, fp);
/*Escreve a string str no arquivo apontado por fp*/
```

A função fgets()

A função fgets() permite que se leia strings de um arquivo.

Exemplo:

```
#define COMPRIMENTO 80
char str[COMPRIMENTO];
FILE *fp;
...
while(!feof(fp))
{ fgets(str, COMPRIMENTO, fp);
}
```

Exemplo de programa que usa as funções fgets() e fputs()

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    FILE * fp;
    char nome[40],arq[]="teste.txt";
    int i=0;
    //abrindo o arquivo no modo gravação
    if((fp=fopen(arq,"w"))==NULL){
        printf("\n Nao pode abrir %s\n",arq);
        exit(1);
    }
```

```
}  
printf("\n Entre com alguns nomes. tecla apenas enter sem digitar nada para  
sair:");  
for(; ;){  
    printf("\nNome %d:",i+1);  
    fflush(stdin);  
    gets(nome);  
    if(!nome[0])  
        break;  
    fputs(nome,fp);  
    fputs("\n",fp);  
}//fim for  
fclose(fp);  
//agora abrindo de novo o arquivo para leitura  
if((fp=fopen(arq,"r"))==NULL){  
    printf("\n Nao pode abrir %s\n",arq);  
    exit(1);  
}  
i=1;  
while(fgets(nome,40,fp) != NULL){  
    printf("\n%d) %s", i,nome);  
    i++;  
}  
fclose(fp);  
system("pause >> log");  
}
```

Caro aluno, um exercício interessante agora é implementar este programa para, em seguida, ir ao diretório corrente e verificar se o arquivo criado está realmente lá com os dados que você forneceu.

Acesse o AVA e teste seus conhecimentos entretendo-se com o jogo de forca sobre arquivos e suas funções.

Agora vamos praticar resolvendo os exercícios propostos. **Leia a lista, resolva os exercícios e verifique seu conhecimento.** Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

REFERÊNCIAS

SCHILDT, H. *C Completo e total*. São Paulo: Makron Books, 1997.

TENEMBAUM, Aaron M., et al. *Estruturas de dados usando C*. São Paulo: Pearson Makron Books, 1995.