

# MC102 - Algoritmos e Progração de Computador

Prof. Alexandre Xavier Falcão

23º Aula: Arquivos

Até o momento, a entrada padrão (teclado) e a saída padrão (tela) foram utilizadas para a comunicação de dados. Muitas situações, porém, envolvem uma grande quantidade de dados, difícil de ser digitada e exibida na tela. Esses dados são armazenados em **arquivos** que ficam em dispositivos secundários de memória (e.g. disco rígido, CDROM, diskete). Portanto, um arquivo é uma seqüência de bytes com uma identificação que permite o sistema operacional exibir na tela seu nome, tamanho, e data de criação.

Existem dois tipos de arquivo: **arquivo texto** e **arquivo binário**. Um arquivo texto armazena os dados em ASCII, na forma de seqüência de caracteres, os quais podem ser exibidos na tela com o comando **more**, se o sistema operacional for unix/linux, ou com o comando **type**, se o sistema for dos. Um arquivo binário armazena os dados na forma binária (seqüência de bits), e normalmente ocupa bem menos espaço em memória do que um arquivo texto para armazenar a mesma informação.

Os arquivos são manipulados com variáveis do tipo apontador para arquivo, as quais são declaradas da seguinte forma:

```
int main()
{
    FILE *aparq;
}
```

Antes de ler ou escrever dados em um arquivo, precisamos endereçá-lo na variável aparq. Este endereçamento é realizado com o comando **fopen**, que pode abrir o arquivo para leitura e/ou escrita. Após leitura e/ou escrita, o comando **fclose** é usado para fechar o arquivo, liberando o apontador aparq.

## 1 Arquivo texto

A leitura de um arquivo texto é feita de forma similar aos casos de leitura da entrada padrão, usando agora comandos **fscanf**, **fgets**, e **fgetc** em vez de **scanf**, **gets**, e **getc**, respectivamente. A mesma observação vale para a gravação, agora usando os comandos **fprintf**, **fputs**, e **fputc**, em vez de **printf**, **puts**, e **putc**, respectivamente.

Por exemplo, suponha o arquivo **arq.txt** abaixo com número de amigos, nome e telefone de cada um, representando uma agenda pessoal.

```
11
Alexandre Falcao # 30
Paulo Miranda # 20
```

Felipe Bergo # 90  
Fabio Capabianco # 25  
Eduardo Picado # 15  
Celso Suzuki # 19  
Joao Paulo # 23  
Jancarlo Gomes # 28  
Danilo Lacerda # 18  
Andre Goncalves # 35  
Carlos Oliveira # 50

Um programa para ler este arquivo, ordenar os registros por nome, e gravar um arquivo ordenado é apresentado abaixo.

```
#include <stdio.h>

typedef struct _amigo {
    char nome[100];
    int telefone;
} Amigo;

typedef struct _agenda{
    Amigo *amigo;
    int num_amigos;
} Agenda;

Agenda LeAgenda(char *nomearq);
void OrdenaAgenda(Agenda a);
void GravaAgenda(Agenda a, char *nomearq);
void DestroiAgenda(Agenda a);

Agenda LeAgenda(char *nomearq)
{
    Agenda a;
    FILE *aparq;
    int i;
    char linha[200],*pos;

    aparq = fopen(nomearq,"r"); /* abre arquivo texto para leitura */
    fgets(linha,199,aparq); /* le cadeia de caracteres do arquivo atÃ© \n */
    sscanf(linha,"%d",&a.num_amigos);
    a.amigo = (Amigo *) calloc(a.num_amigos,sizeof(Amigo));
    if (a.amigo != NULL){
        for (i=0; i < a.num_amigos; i++) {
            fgets(linha,199,aparq);
            pos = strchr(linha,'#');
            strncpy(a.amigo[i].nome,linha,pos-linha-1);
            sscanf(pos+1,"%d",&a.amigo[i].telefone);
        }
    }
}
```

```

    }
}

fclose(aparq); /* fecha o arquivo */
return(a);
}

void OrdenaAgenda(Agenda a)
{
    int i,j,jm;
    Amigo aux;

    for (j=0; j < a.num_amigos-1; j++){
        jm = j;
        for (i=j+1; i < a.num_amigos; i++){
            if (strcmp(a.amigo[i].nome,a.amigo[jm].nome) < 0){
                jm = i;
            }
        }
        if (j != jm){
            aux = a.amigo[j];
            a.amigo[j] = a.amigo[jm];
            a.amigo[jm] = aux;
        }
    }
}

void GravaAgenda(Agenda a, char *nomearq)
{
    int i;
    FILE *aparq;

    aparq = fopen(nomearq,"w"); /* cria novo arquivo texto aberto para escrita */
    fprintf(aparq,"%d\n",a.num_amigos); /* escreve no arquivo */
    for (i=0; i < a.num_amigos; i++)
        fprintf(aparq,"%s # %d\n",a.amigo[i].nome,a.amigo[i].telefone);
    fclose(aparq); /* fecha arquivo */
}

void DestroiAgenda(Agenda a)
{
    if (a.amigo != NULL)
        free(a.amigo);
}

int main()

```

```
{
  Agenda a;

  a = LeAgenda("arq.txt");
  OrdenaAgenda(a);
  GravaAgenda(a,"arq-ord.txt");
  DestroiAgenda(a);
  return(0);
}
```