Aula – Programação II (INF 09330)

UFES
Informática

Manipulação de Arquivos

Prof. Thiago Oliveira dos Santos
Departamento de Informática
Universidade Federal do Espírito Santo

2015

Visão Geral da Aula

UFES
Informática

- Introdução
- Stream
- Arquivos
- Arquivos texto
- Arquivos binários

Visão Geral da Aula

UFES
Informática

- **Introdução**
- Stream
- Arquivos
- Arquivos texto
- Arquivos binários

Introdução

UFES
Informática

Arquivo

- É uma abstração de grupos de dados em um dispositivo
- Permite armazenar dados permanentemente

Arquivos em C

- C permite leitura e escrita em arquivos
- Arquivos são novamente abstraídos com o uso de *stream*
  - *Streams* fornecem os dados para as funções auxiliares
  - Ex. Similar ao uso do teclado e tela

Visão Geral da Aula

UFES
Informática

- Introdução
- **Stream**
- Arquivos
- Arquivos texto
- Arquivos binários

Stream (Fluxo)

UFES
Informática

Conceito

- Seqüência de bytes
- Facilita a interação com outros tipos de dados
  - Meio de conversão de e para seqüência de bytes

Stream (Fluxo)

UFES
Informática

Exemplo de Funções Baseadas em *Stream*

- scanf
  - Lê uma seqüência de bytes da entrada padrão
  - Converte para o formato desejado
  - Coloca em uma variável apropriada
- printf
  - Converte uma variável em uma seqüência de caracteres
    - Com formato apropriado
  - Escreve uma seqüência de bytes na saída padrão

Stream (Fluxo)

UFES
Informática

Strings como *Stream*

- Strings (Cadeia de caracteres) são seqüências de bytes
- Quando vistas como *streams*
  - Podem ser utilizadas para
    - Conversão de dados numéricos para string
    - Conversão de string para dados numéricos

Stream (Fluxo)

UFES
Informática

Funções de String como *Stream*

- int sscanf (const char * s, const char * format, ...)
  - Lê uma seqüência de bytes de uma variável string
  - Converte para o formato desejado
  - Coloca em uma variável apropriada
- int sprintf (char * str, const char * format, ...)
  - Converte uma variável em uma seqüência de caracteres
    - Com formato apropriado
  - Escreve uma seqüência de bytes na variável string

Visão Geral da Aula

UFES
Informática

- Introdução
- Stream
- **Arquivos**
- Arquivos texto
- Arquivos binários

Arquivos

UFES
Informática

Funcionamento

- Também é baseado em *stream*
- Uma *stream* fornece os dados
  - Da leitura de arquivos
  - Para escrita em arquivos
- Segue três operações básicas
  - Abrir o arquivo
    - Obtém a *stream* de um respectivo arquivo
  - Manipular o arquivo (será mais visto adiante)
    - Utiliza a *stream* de um arquivo aberto para ler e escrever dados
  - Fechar o arquivo
    - Utiliza a *stream* de um arquivo aberto para liberar o arquivo

Arquivos

UFES
Informática

Abrir o arquivo

- FILE * fopen (const char * filename, const char * mode)
  - Abre o arquivo com nome *filename*
  - O arquivo é aberto no modo *mode*
  - Retorna a stream usada para manipulação do arquivo aberto

Modo	Resultado
`"r"`	Somente leitura. O arquivo deve já existir.
`"w"`	Somente escrita. O arquivo é criado ou sobrescrito.
`"a"`	Somente escrita. Se o arquivo já existe, adiciona ao final. Do contrário, cria um novo arquivo.
`"r+"`	Para leitura e gravação. O arquivo deve já existir.
`"w+"`	Para leitura e gravação. O arquivo é criado ou sobrescrito.
`"a+"`	Para leitura e gravação. Se o arquivo já existe, adiciona ao final. Do contrário, cria um novo arquivo.

Arquivos

UFES
Informática

Fechar o arquivo

- int fclose (FILE * stream)
  - Fecha o arquivo representado por *stream*
  - Os buffers são esvaziados
  - Retorna NULL para sucesso

Visão Geral da Aula

UFES
Informática

- Introdução
- Stream
- Arquivos
- **Arquivos texto**
- Arquivos binários

Arquivos Texto

UFES
Informática

Manipulação de Arquivos Texto

- É feita por funções específicas
- Interpreta cada caractere (incluindo os dígitos) como um byte
- Dados numéricos
  - São convertidos para string antes de serem escritos
  - São convertidos para números após serem lidos

Arquivos Texto

UFES
Informática

Funções para Manipulação de Arquivos

- int fscanf (FILE * stream, const char * format, ...)
  - Lê uma seqüência de bytes de uma variável stream
  - Converte para o formato desejado
  - Coloca em uma variável apropriada
- int fprintf (FILE * stream, const char * format, ...)
  - Converte uma variável em uma seqüência de caracteres
    - Com formato apropriado
  - Escreve uma seqüência de bytes na variável stream
- int feof (FILE * stream)
  - Verifica se é o final do arquivo
  - Retorna diferente de zero quando o final do arquivo é atingido

Arquivos Texto

UFES
Informática

Exemplo 1

```
char linha[1000];
FILE * pFile, * pFileC;
int n = 0;

pFile = fopen ("test.txt","r");
pFileC = fopen ("copia.txt","w");

if ( !pFile || !pFileC )
        printf("Error opening file\n");
else {
        while ( !feof(pFile) ) {
                fscanf(pFile, "%[^\n]", linha);
                fscanf(pFile, "%*c");
                fprintf(pFileC, "Copia: %s\n", linha);
                printf("(%2d) %s\n", ++n, linha);
        }
        fclose (pFile);
        fclose (pFileC);
}
```

Arquivos Texto

UFES
Informática

Exemplo 2

```
char nome[1000];
FILE * pFile;
int n = 0, n1, n2;
int rtn;
pFile = fopen ("Padrao.txt","r");
if ( !pFile )
        printf("Error opening file\n");
else {
        if ( fscanf(pFile, "%d%*[^\n]\n ", &n) < 1 )
                printf(" Erro lendo numero de linhas\n");

        while ( !feof(pFile) && n > 0 ) {
                rtn = fscanf(pFile, "%[^;];%d;%d%*[^\n]\n", nome, &n1, &n2);
                if ( rtn < 3 ) {
                        printf(" Erro lendo dados (lidos %d)\n", rtn);
                        break;
                }
                printf("-> %s %d %d\n", nome, n1, n2 );
                n--;
        }
        if (n != 0) printf("Arquivo terminou prematuramente\n");
        fclose (pFile);
}
```

Padrao.txt

```
3;
Ola;2;5;
Thiago;10;20;
Joao;100;50;
```

A Função main()

UFES
Informática

Características

- Função como outra qualquer
- Permite passagem de parâmetros
- Permite retorno de um valor inteiro

A Função main()

UFES
Informática

Passagem de Parâmetros

- Faz a comunicação com o mundo externo ao programa
- Feito na linha de comando

```
int main(int argc, char * argv[])
{
   int x;
   for (x =0; x < argc; x++){
      printf("%s \n", argv[x]);
   }

   return 0;
}
```

```
int main(int argc, char * argv[])
{
   if (argc <= 1){
      printf("Favor informar o diretorio!\n");
      return 1;
   }

   printf("O diretorio informado foi: %s", argv[1] );

   return 0;
}
```

A Função main()

UFES
Informática

Retorno

- Faz a comunicação com o mundo externo ao programa
- Permite a verificação de erro
- Formas possíveis
  - Comando return
  - Função exit()
    - Termina em qualquer lugar do programa (incluindo outra função)

```
int main()
{
   ...
   return 0;
}
```

```
int main()
{
   ...
   exit(0);
}
```

Arquivos Texto – (Parâmetro pelo Terminal)

UFES
Informática

Exemplo 3

```
 int main(int argc, char * argv[])
{
   char nome[1000], diretorioNome[2000];
   FILE * pFile;
   int n = 0, n1, n2;
   int rtn;

   if (argc <= 1){
     printf("Favor informar o diretorio!\n");
     return 1;
   }

   printf("O diretorio informado foi: %s\n", argv[1] );
   sprintf(diretorioNome, "%s/Padrao.txt",argv[1] );
   pFile = fopen (diretorioNome,"r");
   .
   .
   .

   return 0;
}
```

Padrao.txt

```
3;
Ola;2;5;
Thiago;10;20;
Joao;100;50;
```

Arquivos Texto

UFES
Informática

Outras Funções

- fseek
  - Permite reposicionar a posição de leitura e escrita
- remove
  - Remove o arquivo
- Etc.

Visão Geral da Aula

UFES
Informática

- Introdução
- Stream
- Arquivos
- Arquivos texto
- **Arquivos binários**

Arquivos Binários

UFES
Informática

Manipulação de Arquivos Binários

- É feita por funções específicas
- Bytes são escritos como vistos em memória
- Gera um arquivo não legível
- Dados numéricos
  - São escritos diretamente de uma variável
  - São lidos diretamente para uma variável
- Requer abertura do arquivo em modo binário
  - Acrescentar “b” no parâmetro *mode* de *fopen*
  - Exemplo
    - fopen("arquivo.bin", "rb")

Arquivos Binário

UFES
Informática

Funções para Manipulação de Arquivos Binários

- size_t fread (void*ptr, size_t size, size_t count, FILE*stream)
  - Lê uma seqüência de bytes para uma variável apontada por *ptr*
  - *size* representa o tamanho do elemento a ser lido
  - *count* representa o número de elementos a ser lido
  - Retorna o número o número de elementos lidos
- size_t fwrite (const void*ptr, size_t size, size_t count, FILE*stream)
  - Escreve uma seqüência de bytes de uma variável apontado por *ptr*
  - *size* representa o tamanho do elemento a ser escrito
  - *count* representa o número de elementos a ser escrito
  - Retorna o número o número de elementos escritos

Arquivos Binário

UFES
Informática

Exemplo

```
FILE *pf;
float testVect[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
int i;

pf = fopen("arquivo.bin", "wb");
if( fwrite(testVect, sizeof(float), 5, pf) != 5 )
        printf("Erro na escrita do arquivo");
fclose(pf);

pf = fopen("arquivo.bin", "rb");
if(fread(testVect, sizeof(float), 5, pf) != 5)
        printf("Erro na leitura do arquivo");
fclose(pf);

for (i = 0; i < 5; i++){
   printf("%.1f ", testVect[i]);
}
```

Perguntas???

UFES
Informática