

MC-102 — Aula 23

Arquivos Binários

Instituto de Computação – Unicamp

Segundo Semestre de 2011

Roteiro

1 Arquivos binários

2 Aplicações

Motivação

- Variáveis `int` ou `float` têm tamanho fixo na memória. Por exemplo, um `int` ocupa 4 bytes.
- Representação em texto precisa de um número variável de dígitos (10, 5.673, 100.340), logo de um tamanho variável.
- Armazenar dados em arquivos de forma análoga a utilizada em memória permite:
 - Reduzir o tamanho do arquivo.
 - Realizar busca não seqüencial.

fread e fwrite

- As funções `fread` e `fwrite` permitem a leitura e escrita de blocos de dados.
- Devemos determinar o número de elementos a serem lidos ou gravados e o tamanho de cada um.

```
unsigned fread(void *buffer, int Num_bytes,  
              int cont, FILE *stream);
```

```
unsigned fwrite(void *buffer, int Num_bytes,  
              int cont, FILE *stream);
```

fread e fwrite

```
unsigned fread(void *buffer, int Num_bytes,  
              int cont, FILE *stream);
```

- buffer: região de memória que armazena dados lidos de fp
- num bytes: tamanho da unidade lida
- cont: quantas unidades lidas
- fread retorna número de unidades lidas (pode ser `j` count se final de arquivo)

```
unsigned fwrite(void *buffer, int Num_bytes,  
              int cont, FILE *stream);
```

- fwrite: "companheira" de fread, porém escreve no arquivo
- fwrite retorna número de itens escritos

Veja o exemplo em `fwrite-fread.c`

Acesso não seqüencial

- Fazemos o acesso não seqüencial usando a função `fseek`.
- Esta função altera a posição de leitura/escrita no arquivo.
- O deslocamento pode ser relativo ao:
 - início do arquivo (`SEEK_SET`)
 - ponto atual (`SEEK_CUR`)
 - final do arquivo (`SEEK_END`)

```
int fseek(FILE *stream, long offset, int whence);
```

Veja o exemplo em `fseek.c`.

Registros

- Um arquivo pode armazenar registros (como um banco de dados).
- Isso pode ser feito de forma bem fácil se lembrarmos que um registro, como qualquer variável em C, tem um tamanho fixo.
- O acesso a cada registro pode ser direto, usando a função `fseek`.
- A leitura ou escrita do registro pode ser feita usando as funções `fread` e `fwrite`.

Veja o exemplo em `registro.c`.

Exercício

Merge

- Escreva um programa que leia dois arquivos de inteiros ordenados e escreva um arquivo cuja saída é um único arquivo ordenado.
 - Vale a pena colocar o conteúdo dos arquivos de entrada em dois vetores?
 - Escreva duas versões deste programa, uma para arquivos texto e outra para arquivos binários.

Exercício

Ordenação

- Escreva um programa que lê uma série de linhas de um arquivo texto e escreve um arquivo contendo estas linhas ordenadas.
 - Como você faria para armazenar estas linhas em memória?
 - Qual algoritmo de ordenação você acha mais adequado?