Laboratório de Introdução à Ciência da Computação I

Aula 13 – Arquivos

Professores:

Jó Ueyama

Arquivo: tipo texto

- *Stream* de texto
- Um arquivo texto é uma sequência de linhas, onde cada linha contém zero ou mais caracteres e termina com um ou mais caracteres que assinalam o fim da linha
- O comprimento máximo de uma linha é 255 caracteres
- Uma linha não é uma *string*, pois seu término não é indicado por ‘\0’
- Na linguagem C o final de linha é indicado pelo caractere ‘\n’ e ao salvar num arquivo, o ‘\n’ é convertido para CR-LF (Carrieage-return/linefeed). Na leitura ocorre o inverso.

Arquivo: tipo binário

- *Stream* binário
- Todos os dados são gravados e lidos exatamente como estão na memória

Nome de arquivos

- Os nomes são armazenados em *strings*
- Deve-se seguir as regras de nomenclatura do sistema operacional
  - Note que no Windows o caminho é indicado pela barra invertida
- O nome do arquivo pode ser informado junto com sua localização no disco
  - Ex: c:\data\conta.txt
- Como em C a barra invertida também tem um significado especial, para representar o caminho numa string é necessário preceder cada barra com outra barra invertida
  - char *filename = “c:\\data\\conta.txt”;

Arquivo: abrir

- Modos de abrir um arquivo:
  - r
  - w
  - a
  - r+
  - w+
  - a+
- Veja o significado de cada parâmetro em:

http://www.cplusplus.com/reference/clibrary/cstdio/fopen/

Exemplo

```
FILE *fp;
char *filename = "conta.txt";
char *mode = "w";
if ((fp = fopen(filename,mode)) != NULL)
   printf("Arquivo aberto com sucesso \n");
else
   printf("Erro na abertura do arquivo \n");
fclose(fp);
```

fopen retorna um ponteiro para o tipo FILE, que é uma estrutura declarada em STDIO.H

Arquivo: abrir do tipo binário

- Por padrão, o arquivo manipulado é do tipo texto

- Para trabalhar com arquivo binário, deve-se acrescentar o caractere ‘b’ no final do modo de abertura
  - Ex: char *mode = "wb";

Entrada e saída de dados formatados

- Arquivos do tipo texto

- Função de saída:

int fprintf (FILE * stream, const char * format, ...);

- Função de entrada:

int fscanf (FILE * stream, const char * format, ...);

- O *fprintf* e o *fscanf* funcionam de modo semelhante ao *printf* e o *scanf*, porém a partir de um *stream* especificado, e não do *stdin (padrão)*

Exemplo com fprintf

```
FILE *fp;
char *filename = "conta";
char *mode = "w";
float salario = 4000.25;
char *nome = "Joao";
float aux_nome[5], aux_sal;

if ((fp = fopen(filename,mode)) != NULL){
   printf("Arquivo texto aberto com sucesso \n");
   fprintf(fp,"%s %f\n", nome, salario);
}
else{
   printf("Erro na abertura do arquivo texto \n");
}
fclose(fp);
```

Exemplo com fscanf

```
FILE *fp;
char *filename = "conta";
char *mode = “r";
float salario = 4000.25;
char *nome = "Joao";
float aux_nome[5], aux_sal;

if ((fp = fopen(filename,mode)) != NULL){
   fscanf(fp,"%s %f", aux_nome, &aux_sal);
   printf("Salario = %s %f\n", aux_nome, aux_sal);
}
else{
   printf("Erro na abertura do arquivo texto \n");
}
fclose(fp);
```

Entrada e saída de caracteres

- Arquivos do tipo texto
- Entrada
  - int getc (FILE * stream);
  - int fgetc (FILE * stream);
  - char *fgets (char* str, int num, FILE *stream);
    - http://www.cplusplus.com/reference/clibrary/cstdio/fgets/
  - getc pode ser implementada como macro, enquanto que fgetc não

Exemplo para inserir no código visto anteriormente

```
int c;
do{
    c = getc (fp);
    printf ("%c\n",c);
    } while (c != EOF);
```

Entrada e saída de caracteres

- Saída
  - int putc (int character, FILE * stream);

http://www.cplusplus.com/reference/clibrary/cstdio/putc/

  - int fputs (const char * str, FILE * stream);

http://www.cplusplus.com/reference/clibrary/cstdio/fputs/

Entrada e saída direta de arquivos

- Arquivos do tipo binário
- Normalmente utilizado para salvar dados que serão lidos pelo mesmo programa
- Blocos de dados da memória podem ser lidos/gravados diretamente de/para arquivo
- Os dados são armazenados de modo compacto

Saída direta

int fwrite(void *buf, int size, int count, FILE *fp);

- O argumento *buf* é um ponteiro para a região da memória que deseja salvar (por ser void, pode apontar para qualquer tipo)
- *Size* especifica o tamanho em bytes dos itens individuais
- *Count* indica a quantidade de itens
- **fp* indica o arquivo
- A função retorna a qtde de itens salvos com sucesso

Implemente e compare o tamanho dos arquivos do tipo texto e binário

```
FILE *fp, *fpb;
char *filename = "conta_tex", *filenameb = "conta_bin";
char *mode = "w", *modeb = "wb";
float salariob = 4000.123;
float salario = 4000.123;
if ((fp = fopen(filename,mode)) != NULL){
        fprintf(fp,"%f",salario);

}
else{
   printf("Erro na abertura do arquivo texto \n");
}
if ((fpb = fopen(filenameb,modeb)) != NULL){
        fwrite(&salariob, sizeof(float),1,fpb);

}
else{
   printf("Erro na abertura do arquivo texto \n");
}
fclose(fp);
fclose(fpb);
```

Leitura direta

int fread (void *buf, int size, int count, FILE *fp);

- De modo oposto ao fwrite, o fread lê um bloco de dados de um arquivo em modo binário para a memória
- Nesse caso o dado é lido do arquivo apontado pelo *fp para a variável *buf

Posição de acesso do arquivo

- Especificada em termos de bytes a partir do início do arquivo
- Ao abrir um arquivo o indicador de posição aponta para o início do arquivo (posição 0), a não ser que o arquivo foi aberto no modo *append* ou e o seu tamanho for > 0;
- Por padrão, a posição muda sequencialmente de acordo com a quantidade de bytes lidos/escritos do arquivo
- Há funções que permitem realizar o reposicionamento

Funções de reposicionamento

- Para obter a posição atual de leitura/escrita
  - long int ftell (FILE * stream);
- Para retornar à posição zero
  - void rewind (FILE * stream);
- Mover para uma posição específica
  - int fseek (FILE * stream, long int offset, int origin);

http://www.cplusplus.com/reference/clibrary/cstdio/fseek/

fseek()

```
FILE * pFile;
pFile = fopen ( "example.txt" , "w" );
fputs ( "This is an apple." , pFile );
fseek ( pFile , 9 , SEEK_SET );
fputs ( " sam" , pFile );
fclose ( pFile );
```

Exercício

- Acrescente as seguintes funcionalidade no primeiro exercício da aula anterior:
  - Permita ao usuário salvar o texto digitado em arquivo no momento que desejar, adicionando ao conteúdo existente no arquivo - o arquivo deverá ser criado automaticamente pelo programa, caso o mesmo ainda não exista;
  - Dar opção ao usuário de visualização do conteúdo do arquivo na tela

Referência

- http://www.cplusplus.com/reference/clibrary/