Entrada e Saída

SSC0300 – Linguagem de Programação e Aplicações

Prof. Dr Jó Ueyama
Estagiário PAE: Heitor Freitas

Ementa desta aula

- Entrada e saída (teclado e monitor)
- Entrada e saída (arquivo)

stdio (standard input/output)

- **stdio.h**:
  - Possui funções e macros de entrada e saída em *buffer*;
  - As funções escrevem e leem de ***streams;***
  - ***Stream*** é uma variável associada a um arquivo (FILE*)
- Ao abrir um arquivo, três *streams* são definidos e abertos automaticamente:
  - **stdin** : *standard input* (entrada padrão, associada ao teclado);
  - **stdout**: *standard output* (saída padrão, associada ao monitor);
  - **stderr**: *standard error* (saída para erros, associada ao monitor).

Entrada *stdin* (scanf)

int scanf (const char * format, **...**);

- Lê de **stdin** e armazena, de acordo com argumento *format*, em local indicado pelo argumento adicional **...**;
- O argumento adicional **...** deve apontar para um objeto já alocado na memória do mesmo tipo indicado por *format;*
- Encerra a leitura ao ler **espaço** ou **\n**;
- Retorna a quantidade de caracteres lidos ou **EOF** se houver erro.

Entrada *stdin* (scanf)

Os tipos mais comuns aceito por *format* são:

Especificador	Descrição	Conversão
d	Inteiro decimal	Inteiro decimal com sinal
f,e	Ponto flutuante	float, double
c	Caracter	unsigned char
s	Cadeia de caracteres	string

Entrada *stdin* (scanf: exemplo)

```
1 #include <stdio.h>
2
3 int main ()
4 {
5   char str [80];
6   int i;
7
8   printf ("Enter your family name: ");
9   scanf ("%79s",str);
10   printf ("Enter your age: ");
11   scanf ("%d",&i);
12   printf ("Mr. %s , %d years old.\n",str,i);
13   printf ("Enter a hexadecimal number: ");
14   scanf ("%x",&i);
15   printf ("You have entered %#x (%d).\n",i,i);
16
17   return 0;
18 }
```

Entrada *stdin* (scanf: exemplo)

A saída gerada é:

```
Enter your family name: Soulie
Enter your age: 29
Mr. Soulie , 29 years old.
Enter a hexadecimal number: ff
You have entered 0xff (255).
```

Entrada *stdin* (scanf: exemplo)

Qual é são as diferenças entre as linhas abaixo?

```
9   scanf ("%79s",str);
11   scanf ("%d",&i);
```

Entrada *stdin* (scanf)

Para saber mais sobre scanf acesse:

http://www.cplusplus.com/reference/cstdio/scanf

Veja também:

fscanf: http://www.cplusplus.com/reference/cstdio/fscanf

fgets: http://www.cplusplus.com/reference/cstdio/fgets

sscanf: http://www.cplusplus.com/reference/cstdio/sscanf

Saída *stdout* (printf)

int printf (const char * format, ...);

- Imprime em **stdout** sua lista de parâmetros **...** Usando a cadeia de formato *format* para determinar a forma como cada um será impresso.
- Retorna o número de caracteres impressos ou um número negativo se houver erro.

Saída *stdout* (printf)

Os tipos mais comuns aceitos por *format* são:

Especificador	Descrição	Conversão
d	Inteiro decimal	Inteiro decimal com sinal
f,e	Ponto flutuante	float, double
c	Caracter	unsigned char
s	Cadeia de caracteres	string

Saída *stdout* (printf: exemplo)

```
 1 #include <stdio.h>
 2
 3 int main()
 4 {
 5    printf ("Characters: %c %c \n", 'a', 65);
 6    printf ("Decimals: %d %ld\n", 1977, 650000L);
 7    printf ("Preceding with blanks: %10d \n", 1977);
 8    printf ("Preceding with zeros: %010d \n", 1977);
 9    printf ("Some different radices: %d %x %o %#x %#o \n", 100, 100,
100, 100, 100);
10    printf ("floats: %4.2f %+.0e %E \n", 3.1416, 3.1416, 3.1416);
11    printf ("Width trick: %*d \n", 5, 10);
12    printf ("%s \n", "A string");
13    return 0;
14 }
```

Saída *stdout* (printf: exemplo)

A saída gerada é:

```
Characters: a A
Decimals: 1977 650000
Preceding with blanks:       1977
Preceding with zeros: 0000001977
Some different radices: 100 64 144 0x64 0144
floats: 3.14 +3e+000 3.141600E+000
Width trick:    10
A string
```

Saída *stdout* (printf)

Para saber mais sobre **printf** acesse:

http://www.cplusplus.com/reference/cstdio/printf

Veja também:

fprintf: http://www.cplusplus.com/reference/cstdio/fprintf

sprintf: http://www.cplusplus.com/reference/cstdio/sprintf

Arquivos

Abrir arquivo (fopen)

FILE * fopen (const char * filename, const char * mode);

- Abre o arquivo especificado por *filename* e associa a um *stream* que pode ser identificado pelo ponteiro **FILE**;
- As operações permitidas sobre esse *stream* é especificado pelo argumento mode;
- Retorna um ponteiro para **FILE**, se aberto arquivo com sucesso, ou ponteiro para **NULL** se houver erro.

Abrir arquivo (fopen)

Os tipos mais comuns aceito por *mode* são:

mode	Descrição
r	**Leitura**: abre arquivo para operações de entrada. O arquivo deve existir.
w	**Escrita**: cria arquivo para operações de saída. Apaga o conteúdo do arquivo com mesmo nome.
a	**Anexar**: abre arquivo para operações de saída. As operações são realizadas no final do arquivo.

Abrir arquivo (fopen: exemplo)

```
1 #include <stdio.h>
2 int main ()
3 {
4   FILE * pFile;
5   pFile = fopen ("myfile.txt","w");
6   if (pFile!=NULL)
7   {
8     fputs ("fopen example",pFile);
9     fclose (pFile);
10   }
11   return 0;
12 }
```

Abrir arquivo (fopen)

Para saber mais sobre **fopen** acesse:

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

Veja também:

freopen: http://www.cplusplus.com/reference/cstdio/freopen

fclose: http://www.cplusplus.com/reference/cstdio/fclose

Fechar arquivo (fclose)

int * fclose (FILE * stream);

- Fecha o arquivo especificado por *stream*;
- Retorna zero se fechado com sucesso ou **EOF** se houver erro.

Fechar arquivo (fclose: exemplo)

```
1 #include <stdio.h>
2 int main ()
3 {
4   FILE * pFile;
5   pFile = fopen ("myfile.txt","wt");
6   fprintf (pFile, "fclose example");
7   fclose (pFile);
8   return 0;
9 }
```

Fechar arquivo (fclose)

Para saber mais sobre **fclose** acesse:

http://www.cplusplus.com/reference/cstdio/fclose

Veja também:

freopen: http://www.cplusplus.com/reference/cstdio/freopen

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

Entrada arquivo (fscanf)

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

- Lê de stream e armazena, de acordo com argumento *format*, em local indicado pelo argumento adicional **...**;
- O argumento adicional **...** deve apontar para um objeto já alocado na memória do mesmo tipo indicado por *format;*
- Encerra a leitura ao ler **espaço** ou **\n**;
- Retorna a quantidade de caracteres lidos ou **EOF** se houver erro.

Entrada arquivo (fscanf)

Os tipos mais comuns aceito por *format* são:

Especificador	Descrição	Conversão
d	Inteiro decimal	Inteiro decimal com sinal
f,e	Ponto flutuante	float, double
c	Caracter	unsigned char
s	Cadeia de caracteres	string

Entrada arquivo (fscanf: exemplo)

```
1 #include <stdio.h>
2
3 int main ()
4 {
5   char str [80];
6   float f;
7   FILE * pFile;
8
9   pFile = fopen ("myfile.txt","w+");
10   fprintf (pFile, "%f %s", 3.1416, "PI");
11   rewind (pFile);
12   fscanf (pFile, "%f", &f);
13   fscanf (pFile, "%s", str);
14   fclose (pFile);
15   printf ("I have read: %f and %s \n",f,str);
16   return 0;
17 }
```

Entrada arquivo (fscanf)

Para saber mais sobre **fscanf** acesse:

http://www.cplusplus.com/reference/cstdio/fscanf

Veja também:

scanf: http://www.cplusplus.com/reference/cstdio/scanf

fgets: http://www.cplusplus.com/reference/cstdio/fgets

sscanf: http://www.cplusplus.com/reference/cstdio/sscanf

Saída - arquivo (fprintf)

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

- Imprime em stream sua lista de parâmetros ... Usando a cadeia de formato *format* para determinar a forma como cada um será impresso.
- Retorna o número de caracteres impressos ou um número negativo se houver erro.

Saída - arquivo (fprintf)

Os tipos mais comuns aceito por *format* são:

Especificador	Descrição	Conversão
d	Inteiro decimal	Inteiro decimal com sinal
f,e	Ponto flutuante	float, double
c	Caracter	unsigned char
s	Cadeia de caracteres	string

Saída - arquivo (fprintf: exemplo)

```
1 #include <stdio.h>
2
3 int main ()
4 {
5   FILE * pFile;
6   int n;
7   char name [100];
8
9   pFile = fopen ("myfile.txt","w");
10   for (n=0 ; n<3 ; n++)
11   {
12     puts ("please, enter a name: ");
13     fgets (name, 100, stdin); //Use fgets no lugar de gets
14     fprintf (pFile, "Name %d [%-10.10s]\n",n,name);
15   }
16   fclose (pFile);
17
18   return 0;
19 }
```

Saída - arquivo (fprintf)

Para saber mais sobre **fprintf** acesse:

http://www.cplusplus.com/reference/cstdio/fprintf

Veja também:

printf: http://www.cplusplus.com/reference/cstdio/printf

sprintf: http://www.cplusplus.com/reference/cstdio/sprintf