

第11章 入出力関数とライブラリ関数

CPro:11-01

概要

•getchar	putchar
•gets	puts
•scanf	printf
•strcat	strcmp
•strcpy	strlen
•atoi	atof
•sprintf	sscanf

11.1 コンソール入出力関数

11-02

入力	出力	
getchar	putchar	... 一文字
gets	puts	... 文字列(一行)
scanf	printf	... 書式付き

```
“test11_2.c” 11-03
1 main()          /* "test11_2.c" (p. 266) */
2 {
3     while(1) {
4         putchar(getchar());
5     }
6 }

% a.exe
abcde,<-
abcde,
12def345<-
12def345
^C
%
```

```
“test11_4.c” 11-04
1 main()          /* "test11_4.c" (p. 268) */
2 {
3     static char string1[] = "abc";
4     static char string2[] = "defYngghiYn";
5     static char string3[] = "jkl";
6
7     puts(string1);
8     puts(string2);
9     puts(string3);
10 }

% a.exe
abc
def
ghi

jkl
%
```

• printf, scanf (書式付きコンソール入出力)

```
printf(" %書式指定 その他文字列 ...", 変数または定数, ...);
scanf(" %書式指定 ...", 変数のアドレス, ...);
```

書式指定

(整数)

- %d 10進整数(符号つき)
- %u 10進整数(符号なし)
- %o 8進整数
- %X 16進数で出力

(実数)

- %f 実数
- %e %E 0.000 e ±00 0.000 E ±00
- %g %G fまたはeで桁数の短い方

(文字)

- %c 文字(一文字)

(文字列)

- %s 文字列(ASCIZで先頭アドレスを与える)

11.2 文字列操作関数

strcat(str1, str2)	文字列str1にstr2を連結
strcmp(str1, str2)	文字列str1とstr2を比較 結果は文字列の差(等しい場合は0)
strncmp(str1, str2, n)	文字列str1とstr2を先頭からn文字比較
strcpy(str1, str2)	文字列str1にstr2をコピー
strncpy(str1, str2, n)	文字列str1にstr2をn文字だけコピー
strlen(str)	文字列strの長さ(文字数)を返す (最後の'¥0'はカウントしない)

※いずれも文字列はASCIZ文字列('¥0'終端)で、
先頭アドレスで指定する

※連結やコピーする場合、連結/コピー先が結果の文字列を格納
するのに十分な領域を確保しておく必要がある

11.3 文字から整数/実数への変換

CPro:11-07

atoi(文字列) 文字列が意味する整数を返す
整数を意味していなければ(数字以外が入っている等)エラー

atof(文字列) 文字列が意味する実数を返す
数を意味していなければ(数字と小数点以外が入っている等)エラー

[例]

```
n = atoi("123");  
f = atof("12.3");  
  
strcpy(str, "4567");  
m = atoi(str);  
a = atof(str);
```

11.4 sprintf() と sscanf()

11-08

sprintf(str, "%書式指定等...", 変数または定数, ...);

printfと同様の結果をコンソールに出力するのではなく
strに文字列として格納する

sscanf(str, "%書式指定等...", 変数のアドレス, ...);

scanfと同様の入力をコンソールからではなく文字列strから
行う

第17章 ファイル入出力

11-09

概要

•fopen	fclose	...	ファイルストリームのopen/close
•getc	putc	...	ファイルから/への1文字入出力
•fgets	fputs	...	ファイルから/への文字列入出力
•fscanf	fprintf	...	ファイルから/への書式指定入出力
•feof	ferror	...	ファイルストリームの終了/入出力エラー判定
•fseek		...	ファイルポジションの指定
•fread	fwrite	...	ファイルから/への バイト単位のバイナリの入出力

17.1 ファイルストリームのオープン/クローズ (fopen / fclose) 11-10

```
FILE *fp;
```

```
fp = fopen(ファイル名文字列, モード文字列);
```

モード文字列

"r"	読み込み
"w"	書き出し
"a"	追加書き出し
"r+"	読み書き(ファイルがなければエラー)
"w+"	読み書き(ファイルがなければ新規作成, あれば上書き)
"a+"	追加読み書き

クローズ

```
fclose(fp);
```

17. 2 ファイルの1文字入出力 (getc / putc)

11-11

getc(fp)

fpから一文字読み込んだ値を返す

putc(c, fp)

fpにc(char)一文字を書き出す

17. 3 ファイルの文字列入出力 (fgets / fputs)

11-12

fgets(str, n, fp)

str 読み込み先(先頭アドレス)

n 最大文字数

fp ファイルストリーム

fputs(str, n, fp)

str 書き出す文字列

n 最大文字数

fp ファイルストリーム

17.4 ファイルの書式指定入出力 (fscanf / fprintf)

CPro:11-13

```
fscanf(fp, "書式指定...", 変数のアドレス, ...);
```

```
fprintf(fp, "書式指定等...", 変数または定数, ...);
```

※書式指定は, scanf / printfと同じ

17.5 ファイルの終了判定 (feof), 入出力エラー検知 (ferror) 11-14

feof(fp)

ファイルストリームが終了ならEOF(-1),
終了でなければ偽(0)を返す

ferror(fp)

ファイルストリームで入出力でエラーが起きた場合真(0以外),
起きていなければ偽(0)を返す

17.6 ファイルポジションの指定 (fseek)

11-15

fseek(fp, offset, mode)

offset	何バイト目か (long整数)
mode	0 ... ファイル先頭から
	1 ... 現在位置から
	2 ... ファイル終端から

17.7 バイナリ入出力 (fread / fwrite)

11-16

fread(buffer, size, count, fp)

buffer	読み込むメモリ位置 (先頭アドレス)
size	読み込むブロックの大きさ (int整数)
count	読み込むブロック数 (int整数)
fp	ファイルストリーム

fwrite(buffer, size, count, fp)

buffer	書き出すメモリ (先頭アドレス)
size	書き出すブロックの大きさ (int整数)
count	書き出すブロック数 (int整数)
fp	ファイルストリーム

○ ... 基本かつ重要問題, 無印 ... ふつうの問題, ☆ ... 難易度高の問題

- 11-1 char 型の変数に整数 106 および 234 を代入し, printf関数を用いて, それぞれ符号付き10進数, 符号なし10進数, 8進数, 16進数および文字として出力せよ. 結果を考察せよ.
- ☆11-2 char 型の変数に数値を4進数として読み込み, その変数の値を10進数, 符号なし10進数, 8進数, 16進数および文字として printf 関数で出力するプログラムを作成せよ. なお, 数値を読み込むには getchar 関数を使って4進数を入力するための関数を作成し, それを用いること. 試す値として, 1300 (4) や211 (4) 等を用いること.
- 11-3 double 型の変数に, scanf 関数を用いて値を読み込み, その変数の値を10倍したもの, 100倍したもの, 1000倍したもの. . . と, 1/10倍したもの, 1/100倍したもの, 1/1000倍したもの. . . を, 有効桁20桁まで "0.??...??E±??" の形式で出力するプログラムを作成せよ. ただし, 値の出力には printf 関数 (の%編集) 等を用いず, putchar 関数を使って自分で出力関数を作成し, これを用いること. 読み込む値として, 0.1, その他を用いること.

- 11-4 文字列 "cdefg" と つぎの各文字列を strcmp 関数で比較した結果の値を確認せよ.

```
a
abcde
c
cdefg
cedfg
def
zzzzz
01234
```

- 11-5 gets 関数により標準入力から1行を文字列として読み込み, 最初の数文字が"INT" なら整数, "FLOAT" なら浮動小数, "HEX" なら16進整数として, つづく2つの数を sscanf 関数で文字列から読み込み, その和, 差, 積, 商を計算するプログラムを作成せよ. EOF が入力されるまで何行でも処理を繰り返すこと.

(ヒント)

最初の数文字の文字列比較には, strncmp 関数を用いるとよい. 引数の意味等の詳細はオンライン・マニュアルで調べること.

- 11-6 putchar(), puts(), fputs(), および, getc(), getchar(), fgets()のそれぞれの関数の違いについてオンライン・マニュアルその他で調べよ.

- ☆11-7 printf 関数は引数としてアドレスではなく変数をそのまま与えるのに対して、scanf 関数を用いて変数に数値を読み込むときには、scanf 関数の引数は変数のアドレスを与えるようになっている。その理由を考えよ。
- 11-8 strcmp()関数と同じ動作をする関数 mystrcmp を作成せよ。
- 11-9 strcpy()関数と同じ動作をする関数を作成せよ。ただし、文字列の最初から順にコピーする関数を mystrcpy1、文字列の後ろから1文字ずつ逆のぼって順にコピーする関数を mystrcpy2 という関数名とすること。
- ☆11-10 ある文字（何でもよいが例えば '+'）を、単に 100,000 個（くらい。使うコンピュータの速度にもよるが、10,000 ~ 1,000,000 位で差が出るような個数を選ぶこと）を、標準出力に出力するプログラムを作れ。
ただし、文字の出力に、①関数 (putchar) と②システムコール (write) を使うプログラムをそれぞれ作成し、双方の実行時間を time コマンド等で計測し比較してみること。その結果を考察せよ。
(ヒント)
オンラインマニュアル2章の write を調べること。この問題では、write の引数 fd に標準出力を意味する 1 を指定すればよい。
- ☆11-11 次のプログラムを作成し、
①実行結果を予測せよ。（得られる結果として何を意図しているか）
②gccでは、コンパイルはできるが実行時エラーとなる（古い cc では実行できるものもあった）。その理由を考えよ。

```

1      /* prog11-11a.c */
2      #include <stdio.h>
3
4      main()
5      {
6          char    *p, *q;
7
8          p = "abc";
9          q = "xy";
10         *p = *q;
11         printf("%s %s\n", p, q);
12     }

```

(ヒント) 次のように修正すれば、gccでも実行できる。

```

1      /* prog11-11b.c */
2      #include <stdio.h>
3      #include <string.h>
4      main()
5      {
6          char    p[4], q[4];
7
8          strcpy(p, "abc");
9          strcpy(q, "xy");
10         *p = *q;
11         printf("%s %s\n", p, q);
12     }

```

○17-1 コマンド行の引数として与えられた文字列や、実行時の環境変数を確認するプログラムを作成せよ。

(ヒント)

コマンド行の引数等は、Cプログラムの main 関数の引数として得ることができる。

```
main(argc, argv, envp)
int   argc; /* 引数の個数(コマンドを含む) */
char **argv; /* 引数の内容 (文字列) */
char **envp; /* 環境変数の内容 (文字列) */
{
```

○17-2 コマンド行の引数として与えられた文字列をファイル名とし、そのファイルの内容を、最終行から行ごとに逆順で出力するプログラムを作成せよ。