ファイル処理

０．目次

１．はじめに

ファイル処理プログラムの形は次のようになる。

```
#include <stdio.h>
main() {
  FILE *fp1,*fp2;  ファイルポインタの宣言。
                   ファイルはすべて、ファイルポインタを
                   用いて指定する。

  fp1 = fopen("infile","r");  ファイル（infile）を読み取り
                              専用としてオープン（ファイル
                              アクセスの準備）する。
  fp2 = fopen("outfile","w");  ファイル（outfile）を書き込み
                               専用としてオープンする。

   ファイル処理

  fclose(fp1);  ファイルをクローズ（ファイルアクセスの
                後始末）する。
  fclose(fp2);
}
```

Cでは、ファイルへの入出力はバイトの列として行われる。

関数	fopen
機能	ファイルをオープンする。
引数	第１引数はファイル名。 第２引数はアクセスのモード。 "r"は読み込むためにオープンすることを意味する。 "w"は書き出すためにオープンすることを意味する。 "a"はファイルの最後に追加するためにオープンする "r+"はファイルを更新するためにオープンすることを意味する。
結果	オープンできたときは、ファイルポインタを返す。 できなかったときは、NULLを返す。

関数	fclose
機能	ファイルをクローズする。
引数	第１引数はファイルを指すファイルポインタ。
結果	正常にクローズしたときは0を返す。 クローズできなかったときはEOFを返す。

２．ファイル内容の表示

ファイルから１バイト単位（１文字）で入力し、画面に表示する。

関数	getc
機能	ファイルから１バイト読み込む。
引数	第１引数は入力元ファイルを指すファイルポインタ。
結果	読み込んだデータ（整数型）を返す。 ファイルの終わりに達したときEOFを返す。

```
 1  /* << f211.c >> */
 2  #include <stdio.h>
 3  main() {
 4    char ch;
 5    char infile[81]; /* 入力データ用配列。*/
 6    FILE *fp1;       /* ファイルポインタ。*/
 7    /* 入力ファイルを指定する。*/
 8    printf("input file: "); scanf("%s",infile);
 9    fp1 = fopen(infile,"r");
10    /* 入力ファイルから1バイトずつ読み込み、画面に表示する。
11       各行の最後は改行コード（10進で10）がある。*/
12    while( (ch = getc(fp1)) != EOF ) {
13      printf("[%c] [%x]¥n",ch,ch);
14    }
15    printf("[%d]¥n",ch);
16    fclose(fp1);
17  }
```

実行結果

```
% cat f211.dat
12
ab
% cc f211.c
% a.out
input file: f211.dat
[a] [97]
[b] [98]
[c] [99]
[   <-- 改行が実行されている。
] [10]
[-1]
```

３．ファイル内容の複写

３．１　文字単位

ファイルから１バイト単位（１文字）で入力し、別のファイルに
１バイト単位で出力する。

関数	putc
機能	ファイルに１バイトを書き込む。
引数	第１引数は文字。 第２引数は出力先ファイルを指すファイルポインタ。
結果	エラーを検出したときEOFを返す。

```
 1  /* << f311.c >> */
 2  #include <stdio.h>
 3  main() {
 4    char ch;
 5    char infile[81],outfile[81];
 6    FILE *fp1,*fp2;
 7    /* 入力ファイルを指定する。*/
 8    printf("input  file: "); scanf("%s",infile);
 9    fp1 = fopen(infile,"r");
10    /* 出力ファイルを指定する。*/
11    printf("output file: "); scanf("%s",outfile);
12    fp2 = fopen(outfile,"w");
13    /* 入力ﾌｧｲﾙから1ﾊﾞｲﾄずつ読み込み、出力ﾌｧｲﾙに書き込む。*/
14    while( (ch = getc(fp1)) != EOF ) {
15      putc(ch,fp2);
16    }
17    fclose(fp1);
18    fclose(fp2);
19  }
```

実行結果

```
% cat f311.dat
0123456789
abcdefg
ABCDEFG
茨城大学
% cc f311.c
% a.out
input  file: f311.dat
output file: f311.rst
% cat f311.rst
0123456789
abcdefg
ABCDEFG
茨城大学
```

３．２　行単位

ファイルから１行単位で入力し、別のファイルに１行単位で出力する。

関数	fgets
機能	ファイルから1行のデータを読み込む。 読み込んだ1行の文字列の末尾にナル文字を付ける。
引数	第１引数はデータを格納する場所を指定するcharポインタ。 第２引数は文字列のサイズの上限。 第３引数は入力元ファイルを指定するファイルポインタ。
結果	ファイルの終わりに達したときNULLを返し、その他の場合は 第１引数を返す。

関数	fputs
機能	ファイルに1行のデータを書き込む。
引数	第１引数は出力する文字列を指すcharポインタ。 第２引数は出力先ファイルを指定するファイルポインタ。
結果	エラーを検出したときEOFを返す。

```
1  /* << f321.c >> */
2  #include <stdio.h>
3  main() {
4    FILE *fp1,*fp2;
5    char infile[81],outfile[81],string[256];
6    /* 入力ファイルの指定。*/
7    printf("input  file: "); scanf("%s",infile);
8    fp1 = fopen(infile,"r");
9    /* 出力ファイルの指定。*/
10   printf("output file: "); scanf("%s",outfile);
11   fp2 = fopen(outfile,"w");
12   while( fgets(string,256,fp1) != NULL ) {
13     fputs(string,fp2);
14   }
15   fclose(fp1);
16   fclose(fp2);
17 }
```

実行結果

```
% cat f321.dat
0123456789
abcdefg
茨城大学
% cc f321.c
% a.out
input  file: f321.dat
output file: f321.rst
% cat f321.rst
0123456789
abcdefg
茨城大学
```

４．書式付き入出力

ファイルから書式にしたがってデータを入力し、別のファイルに書式にしたがってデータを出力する。

関数	fscanf
機能	ファイルから書式にしたがってデータを読み込む。 第１引数としてファイルポインタをとる以外はscanfと同じ。

関数	fprintf
機能	ファイルに書式にしたがってデータを書き込む。 第１引数としてファイルポインタをとる以外はprintfと同じ。

```
 1  /* << f411.c >> */
 2  #include <stdio.h>
 3  main() {
 4    FILE *fp1,*fp2;
 5    int x;
 6    char infile[81],outfile[81];
 7    /* 入力ファイルの指定。*/
 8    printf("input  file: "); scanf("%s",infile);
 9    fp1 = fopen(infile,"r");
10    /* 出力ファイルの指定。*/
11    printf("output file: "); scanf("%s",outfile);
12    fp2 = fopen(outfile,"w");
13    while( fscanf(fp1,"%d",&x) == 1 ) {
14      x = -x;
15      fprintf(fp2,"%8d ¥n",x);
16    }
17    fclose(fp1);
18    fclose(fp2);
19  }
```

実行結果

```
% cat f411.dat
12
345
6789
% cc f411.c
% a.out
input  file: f411.dat
output file: f411.rst
% cat f411.rst
     -12
    -345
   -6789
```

５．文字配列への入出力

文字配列から書式にしたがってデータを入力したり、変数の値を書式にしたがって文字配列に出力することができる。

関数	sscanf
機能	文字配列から書式にしたがってデータを読み込む。
引数	第１引数は文字配列。 第２引数は書式（scanfと同じ）。 第３引数は変数アドレスの並び。
結果	読み込んだ要素の数。

関数	sprintf
機能	変数の値を書式にしたがって文字配列に書き込む。
引数	第１引数は文字配列。 第２引数は書式（printfと同じ）。 第３引数は変数の並び。
結果	書き込んだ文字数。

```
 1  /* << f511.c >> */
 2  #include <stdio.h>
 3  main() {
 4    int  a1,a2,m,n;
 5    char b1[20],b2[20],c[20];
 6    /* サンプルデータ。*/
 7    a1 = 12345;
 8    b1[0] = 'a'; b1[1] = 'b'; b1[2] = 'c'; b1[3] = '¥0';
 9    printf("a1 = %d¥n",a1);
10    printf("b1 = %s¥n",b1);
11    n = sprintf(c,"%d %s",a1,b1);
12    printf("c = %s¥n",c);
13    printf("n = %d¥n",n);
14    m = sscanf(c,"%d%s",&a2,b2);
15    printf("a2 = %d¥n",a2);
16    printf("b2 = %s¥n",b2);
17    printf("m = %d¥n",m);
18  }
```

実行結果

```
% cc f511.c
% a.out
a1 = 12345
b1 = abc
c = 12345 abc
n = 9
a2 = 12345
b2 = abc
m = 2
```

６．課題

６．１　課題１（ファイル圧縮・復元）

簡単な圧縮ルール(連続するk(1<=k<=100)個の空白を、空白と2桁の数字k-1に変換)に基づき、ファイルを圧縮する。
［例］ abc123　　def --> abc123 04def

```
1  /* << f611.c >> */
2  /* ファイルの圧縮 */
3  #include <stdio.h>
4  main(int argc, char *argv[]) {
5    int ch,count,i;
6    FILE *fp1,*fp2;
7    printf("圧縮開始 ¥n");
8    fp1 = fopen(argv[1],"r"); fp2 = fopen(argv[2],"w");
9    count = 0; /* 連続する空白の個数。*/
10   while( (ch = getc(fp1)) != EOF ) {
11     if( count == 0 ) {
12       if(                ) { count++; } else { putc(ch,fp2); }
13     } else {
14       if(              ) {
15         count++;
16       } else {
17         if(                ) { fprintf(fp2," 0%d",count-1); }
18         if(                ) { fprintf(fp2," %2d",count-1); }
19         putc(ch,fp2); count = 0;
20       }
21     }
22   }
23   fclose(fp1); fclose(fp2); printf("圧縮終了 ¥n");
24 }
```

実行結果

```
% cat f611.dat
a b
a  b
a   b
a    b
a     b
% cc f611.c
% a.out f611.dat f611.rst
圧縮開始
圧縮終了
% cat f611.rst
a 00b
a 01b
a 02b
a 03b
a 04b
```

圧縮規則（連続するk(1<=k<=100)個の空白を、空白と2桁の数字k-1に変換）にしたがって圧縮されたファイルを元に復元する。
［例］ abc123 04def --> abc123 def

```
1  /* << f621.c >> */
2  /* ファイルの復元 */
3  #include <stdio.h>
4  main(int argc, char *argv[]) {
5    int ch,
6        count, /* 連続する空白の個数。*/
7        d1,d2,i;
8    FILE *fp1,*fp2;
9    printf("復元開始 ¥n");
10   fp1 = fopen(argv[1],"r"); fp2 = fopen(argv[2],"w");
11   while( (ch = getc(fp1)) != EOF ) {
12     if( ch != ' ' ) {
13       putc(ch,fp2);
14     } else {
15       d2 = getc(fp1); d2 = d2 - '0';
16       d1 = getc(fp1); d1 = d1 - '0';
17       count =
18       for( i=1; i<=count; i++ ) { putc(' ',fp2); }
19     }
20   }
21   fclose(fp1);
22   fclose(fp2);
23   printf("復元終了 ¥n");
24 }
```

実行結果

```
% cat f621.dat
a 00b
a 01b
a 02b
a 03b
a 04b
% cc f621.c
% a.out f621.dat f621.rst
復元開始
復元終了
% cat f621.rst
a b
a  b
a   b
a    b
a     b
```