

第7回 プログラミング応用レポート

15302114 番 山下尚人

提出日：2017 年 12 月 22 日

1 ex103.c

● ソースコード

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void){
5      int val;
6      int sum = 0;
7      int cnt = 0;
8      FILE *fpin, *fpout;
9
10     if ( (fpin=fopen("exdata1.dat","r")) == NULL){
11         printf("File not found. --- exdata1.dat\n");
12         exit(EXIT_FAILURE);
13     }
14
15     if ( (fpout=fopen("out.dat","w")) == NULL ){
16         fclose(fpin);
17         printf("File not made. --- out.dat\n");
18         exit(EXIT_FAILURE);
19     }
20
21     while( fscanf(fpin,"%d",&val) != EOF){
22         fprintf(fpout,"Input data>>>%d\n", val);
23         sum += val;
24         ++cnt;
25     }
26
27     if(cnt>0){
28         printf("SUM=%d, AVE=%g\n", sum, (double)sum/cnt);
29     }
30
31     fclose(fpin);
32     fclose(fpout);
33
34     return EXIT_SUCCESS;
35 }
```

- 入力データファイル (exdata1.dat)

```
1 150
2 164
3 153
4 174
5 189
6 185
7 168
8 156
9 152
10 174
11 173
12 169
```

- 実行結果

```
1
2 [myuser@linux01 8th_lecture]$ ls
3 a.out ex103.c exdata1.dat
4 [myuser@linux01 8th_lecture]$ ./a.out
5 SUM=2007, AVE=167.25
```

- 出力データファイル (out.dat)

```
1 Input data >>> 150
2 Input data >>> 164
3 Input data >>> 153
4 Input data >>> 174
5 Input data >>> 189
6 Input data >>> 185
7 Input data >>> 168
8 Input data >>> 156
9 Input data >>> 152
10 Input data >>> 174
11 Input data >>> 173
12 Input data >>> 169
```

- 考察

10～13 行目で exdata1.dat を”r”モードで fpin に展開している。

15～19 行目で out.dat を”w”モードで fpout に展開している。

21～25 行目で fpin から 1 行ずつデータを val に読み込み、val を fpout に書き込んだ後、sum に val を加算している。

27～29 行目で `fpin` のデータの合計と平均を標準出力に出力している。

2 ex104 : append(追記) の場合

- ソースコード

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main (void){
5      FILE *fp;
6
7      if ( (fp=fopen("sample.dat","a")) == NULL){
8          printf("File not found. ---sample.dat\n");
9          exit(EXIT_FAILURE);
10     }
11
12     printf("debug\n");
13
14     fprintf(fp,"But ,if you do your best ,\n");
15     fprintf(fp,"you will manage.\n");
16
17     fclose(fp);
18
19     return EXIT_SUCCESS;
20 }
```

- 実行結果

```
1  [myuser@linux01 8th_lecture]$ ls
2  a.out  ex103.c  ex104.c  exdata1.dat  logdata1.txt  logdata2
   .txt  out.dat
3  [myuser@linux01 8th_lecture]$ ./a.out
4  debug
5  [myuser@linux01 8th_lecture]$ ls
6  a.out      ex104.c      logdata1.txt  out.dat
7  ex103.c    exdata1.dat  logdata2.txt  sample.dat
8  [myuser@linux01 8th_lecture]$ cat sample.dat
9  But, if you do your best,
10 you will manage.
11 [myuser@linux01 8th_lecture]$ ./a.out
12 debug
13 [myuser@linux01 8th_lecture]$ ls
14 a.out      ex104.c      logdata1.txt  out.dat
15 ex103.c    exdata1.dat  logdata2.txt  sample.dat
16 [myuser@linux01 8th_lecture]$ cat sample.dat
17 But, if you do your best,
18 you will manage.
19 But, if you do your best,
20 you will manage.
```

- コードの考察

7~10 行目で sample.dat を”a”モードで fp に展開している。

13,14 行目で fp に文字列を書き込んでいる。

- 実行結果の考察

1~10 行目で sample.dat がない状態でコードを実行すると、文字列が 1 回書き込まれた sample.dat が作られる。

11~20 行目で sample.dat がある状態でコードを実行すると、文字列が末尾に新たに書き込まれる。

3 ex104 : write(上書き) の場合

- ソースコード

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main (void){
5      FILE *fp;
6
7      if ( (fp=fopen("sample.dat","w")) == NULL){
8          printf("File not found. ---sample.dat\n");
9          exit(EXIT_FAILURE);
10     }
11
12     printf("debug\n");
13
14     fprintf(fp,"But , if you do your best ,\n");
15     fprintf(fp,"you will manage.\n");
16
17     fclose(fp);
18
19     return EXIT_SUCCESS;
20 }
```

- 実行結果

```
1  [myuser@linux01 lecture08]$ ls
2  a.out      ex104_1.c  exdata1.dat  logdata2.log  sample.dat
3  ex103.c    ex104_2.c  logdata1.log  out.dat
4  [myuser@linux01 lecture08]$ cat sample.dat
5  But, if you do your best,
6  you will manage.
7  But, if you do your best,
8  you will manage.
9  [myuser@linux01 lecture08]$ ./a.out
10 debug
11 [myuser@linux01 lecture08]$ ls
```

```
12 a.out      ex104_1.c  exdata1.dat  logdata2.log  sample.dat
13 ex103.c    ex104_2.c  logdata1.log  out.dat
14 [myuser@linux01 lecture08]$ cat sample.dat
15 But, if you do your best,
16 you will manage.
17 [myuser@linux01 lecture08]$ rm sample.dat
18 [myuser@linux01 lecture08]$ ls
19 a.out      ex104_1.c  exdata1.dat  logdata2.log
20 ex103.c    ex104_2.c  logdata1.log  out.dat
21 [myuser@linux01 lecture08]$ ./a.out
22 debug
23 [myuser@linux01 lecture08]$ ls
24 a.out      ex104_1.c  exdata1.dat  logdata2.log  sample.dat
25 ex103.c    ex104_2.c  logdata1.log  out.dat
26 [myuser@linux01 lecture08]$ cat sample.dat
27 But, if you do your best,
28 you will manage.
```

- コードの考察

2 の追記の場合のコードから、7 行目の”a”を”r”に変更した。

- 実行結果の考察

1～16 行目で sample.dat がある状態でコードを実行すると、元のデータが消え、新たに文字列が 1 回書き込まれる。

17～20 行目で sample.dat を消している。

21～28 行目で sample.dat がない状態でコードを実行すると、sample.dat が新たに作成され、文字列 1 回が書き込まれる。