

C言語講座 第4回

ファイルの入出力



システム標準関数

- 第1回に習った”printf”や”scanf”は、もともと Microsoft Visual C++ 2010 Express によって備え付けられた関数である。これを「システム標準関数」という。
- このシステム標準関数を利用することによって、キーボードからの入力や画面への出力に限らず、ファイルの入出力も可能になる。

ファイル入出力のシステム関数

- fopen : ファイルを開く。
- fclose : ファイルを閉じる。
- fprintf : ファイルに書式付の文字列を書き込む。
(書き込みモードに使用される)
- fscanf : ファイルから文字列を読み込み、指定された形式に変換する。(読み込みモードに使用される)
- fgets : ファイルから1行読みだす。
- fputs : ファイルに1行書き込む。

他にもいくつか存在する。モードの説明は後程。

ファイルオープン・クローズの書き方

- ファイルオープン

```
FILE *変数名;
```

```
変数名 = fopen(“ファイル名” ,”モード” );
```

- ファイルクローズ

```
fclose(変数名);
```

前スライドの説明

- FILE ***変数名**;
FILE*型ポインタ変数の宣言。
- **変数名** = fopen(“**ファイル名**” , “**モード**”);
ファイルを開く操作の記述。
- fclose(**変数名**);
開いたファイルを閉じる操作の記述。

変数名は自分で自由に決めて良い。

ファイルとモード

- ファイルには「テキストファイル」と「バイナリファイル」の2種類があり、どちらのファイルをどのような形で開くのかを指定するのが、モードの役割である。
- テキストファイルはごく一般的なファイルであるが、バイナリファイルでは音声や画像を扱うことが可能である。

モードの種類

- r : テキストファイルの読み込みモード。
- w : テキストファイルの書き込みモード。
- a : テキストファイルの追加モード。
- rb : バイナリファイルの読み込みモード。
- wb : バイナリファイルの書き込みモード。
(ファイルが既に存在する場合は上書き)
- r+ : テキストファイルの更新モード。
- w+b(wb+) : バイナリファイルの更新モード。
(ファイルが存在する場合は上書き)

例題1 : 4-1.c

- ```
#include <stdio.h>
int main(){
FILE *fp;
char str[256];
fp = fopen("open.txt","w");
```

```
if(fp==NULL){
printf("ファイルオープン失敗。¥n");
}
```

```
fclose(fp);
return 0;
}
```

- ファイル「open.txt」はフォルダの中に作られているはず。
- 5行目のかっこ内を「"close.txt","r"」、「"close.txt","a"」にすると？



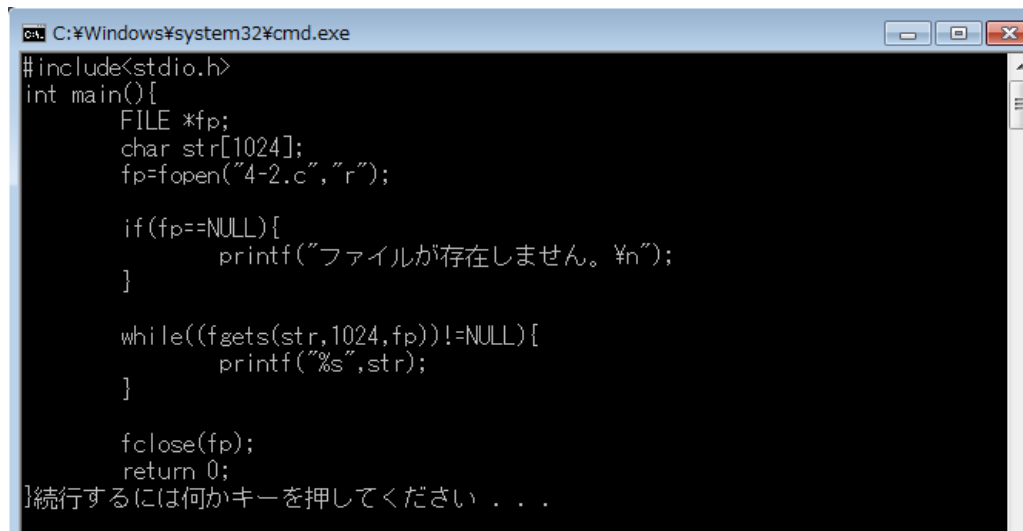
## 例題2 : 4-2.c

```
#include <stdio.h>
int main(){
FILE *fp;
char str[1024];
fp = fopen("4-2.c","r");

if(fp==NULL){
printf("ファイルが存在しません。＼n");
}

while((fgets(str,1024,fp))!=NULL){
printf("%s",str);
}

fclose(fp);
return 0;
}
```



```
cmd: C:\Windows\system32\cmd.exe
#include<stdio.h>
int main(){
FILE *fp;
char str[1024];
fp=fopen("4-2.c","r");

if(fp==NULL){
printf("ファイルが存在しません。＼n");
}

while((fgets(str,1024,fp))!=NULL){
printf("%s",str);
}

fclose(fp);
return 0;
}続行するには何かキーを押してください . . .
```

- 5行目の「4-2.c」を「4-1.c」や前回までの講座の例題のファイル名にすると？

# 例題1,2の補足

- 例題1

5行目のかっこ内を「" close.txt","r"」にしてデバッグすると、エラーが出力される。

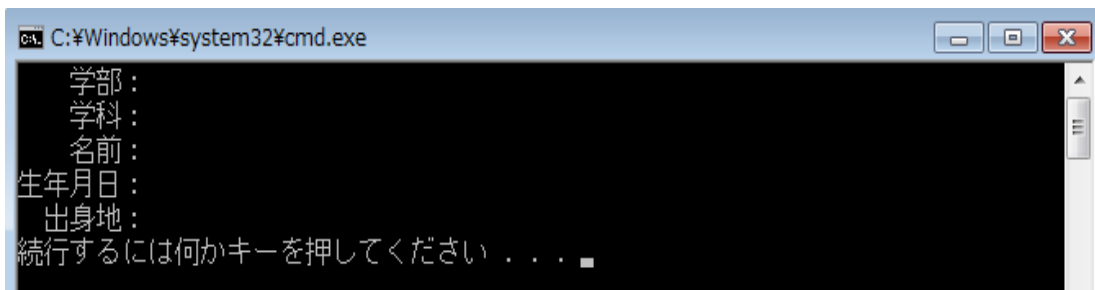
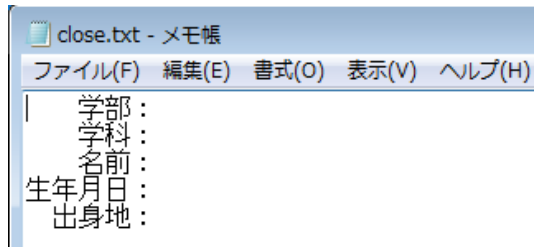
「" close.txt","a"」だと、フォルダに新しく close.txtが作られる。先に「" close.txt","a"」をデバッグし、「" close.txt","r"」でデバッグすると、何も起こらず終了する。

- 例題2

存在するファイル名であれば、そのファイルが読み込まれる。存在しないファイル名ならば、エラーで返される。

# 演習1：4-3.c

- 例題1で作った「open.txt」に、次のスライドのテンプレを使って自己紹介を書いてください。  
※書いたら保存させてください。
- その自己紹介をデバッグ画面及び「close.txt」に同時に表示させるプログラムを作成してください。



# 自己紹介のテンプレ

学部：

学科：

名前：

生年月日：

出身地：

## バイナリファイルのファイル入出力

- 基本的には今までやってきたテキストファイルのオープンクローズと変わらない。モードの部分に”b”を書き足すことで、バイナリファイルに関するものとなる。(スライド7枚目参照)
- システム関数として、書き込みにはfwrite関数、読み込みにはfread関数を用いる。

## fwrite, freadの書き方。

- fwrite(配列名, データの大きさ, データの数, 変数名);  
※freadも同じ。
- 基本的に、これらはfclose関数の直前に書く。

## 例題3 : 4-4.c

```
#include <stdio.h>
int main()
{
char fi[50]="red.bmp";
int i, j;
FILE *fp;
unsigned short header[13] = {0x4d42, 54490, 1, 0, 0, 26, 0, 12, 0, 200, 200, 1, 24};
unsigned char bmp[200][200][3];
for(i=0;i<200;i++)
{
 for(j=0;j<200;j++)
 {
 bmp[i][j][0]=0;
 bmp[i][j][1]=0;
 bmp[i][j][2]=255;
 }
}
```

次のスライドに続く。

## 例題3：続き

```
fp = fopen(fi,"wb");
if((fp=fopen(fi,"wb"))==NULL)
{
fprintf(stderr, "Error: file open [%s].\n", fi);
exit(1);
}
fwrite(header, 2, 13, fp);
fwrite(bmp, 1, 200*200*3, fp);
fclose(fp);
return 0;
}
```



フォルダに「red.bmp」で保存されているはずである。  
前スライドのfor文の数字を0～200の間で変えるとどうなるか。  
また、配列の数字を0～255の間で変えるとどうなるか。



## 例題3の補足

- 「`unsigned short header[13] = {0x4d42, 54490, 1, 0, 0, 26, 0, 12, 0, 200, 200, 1, 24};`」によって、 $200 \times 200$ ピクセル、24ビットの画像を表す。
- 「`unsigned char bmp[200][200][3];`」は画像の大きさ(ピクセル)と三原色を表す。
- 三原色は配列の上から順に青、緑、赤であり、数値が大きいほどその色は濃くなる。もちろん、2色、3色と混ぜることも可能。
- 「`fwrite(header, 2, 13, fp);`」はヘッダーの書き込み、  
「`fwrite(bmp, 1, 200*200*3, fp);`」はbmp画像の書き込みを行う。

## 最終問題：4-5.c

- 例題3を参考に、下のような画像を生成してください。画像の大きさは200×300ピクセルとする。出力ファイル名は自分で決めてください。



# ヒント

- 「unsigned short header[13] = {0x4d42, 54490, 1, 0, 0, 26, 0, 12, 0, 300, 200, 1, 24};」 、  
「unsigned char bmp[200][300][3];」はこのまま写して使ってください。
- 黄色は2色を混ぜると出来ます。黒は・・・？

※完成した人は周りの人の手助けや、ファイル名を変えるなどしていろいろな色で試してみよう！

# 演習1解答例(1/2)

```
#include <stdio.h>
int main(){
FILE *fp,*gp;
char str[256];

fp = fopen("open.txt","r");
if(fp==NULL){
printf("ファイルオープンエラー");
}

gp = fopen("close.txt","w");
if(gp==NULL){
 printf("ファイルオープンエラー");
}
```

# 演習1解答例(2/2)

```
while(fgets(str,255,fp)){
 printf("%s", str);
 fputs(str,gp);
}
```

```
printf("¥n");
fclose(fp);
fclose(gp);
return 0;
}
```

# 最終問題解答例(1/3)

```
#include <stdio.h>
main()
{
char fi[50]="bin.bmp";
int i, j;
FILE *fp;
unsigned short header[13] = {0x4d42, 54490, 1, 0, 0, 26, 0, 12, 0, 300, 200, 1, 24};
unsigned char bmp[200][300][3];
for(i=0;i<200;i++)
{
 for(j=0;j<100;j++)
 {
 bmp[i][j][0]=0;
 bmp[i][j][1]=0;
 bmp[i][j][2]=0;
 }
}
```

# 最終問題解答例(2/3)

```
for(i=0;i<200;i++)
{
 for(j=100;j<200;j++)
 {
 bmp[i][j][0]=0;
 bmp[i][j][1]=255;
 bmp[i][j][2]=255;
 }
}
for(i=0;i<200;i++)
{
 for(j=200;j<300;j++)
 {
 bmp[i][j][0]=0;
 bmp[i][j][1]=0;
 bmp[i][j][2]=255;
 }
}
```

# 最終問題解答例(3/3)

```
fp=fopen(fi,"wb");
if((fp=fopen(fi,"wb"))==NULL)
{
fprintf(stderr, "Error: file open [%s].\n", fi);
exit(1);
}
fwrite(header, 2, 13, fp);
fwrite(bmp, 1, 200*300*3, fp);
fclose(fp);
}
```