オブジェクト指向とゲームプログラミング

● 基礎編 ― 第14回 ファイルの操作 ●

ファイルのオープン

C言語でファイルを読み書きするには、まず最初にファイルをオープンしなければなりません。ファイルをオープンするには、fopen関数を使用します。

fopen関数
―説明―
fopen関数は、ファイルを開き、ファイルへアクセスするためのファイルポインタを返します。

―書式―
```
FILE* fopen(const char* filename, const char* mode);
```

―パラメータ―
1つ目の引数(filename)は、オープンするファイル名です。

2つ目の引数(mode)は、オープンモード文字列です。以下の文字列を指定します。
・テキストモード

"r"	読み取りモード。ファイルがなければ失敗します。
"w"	書き込みモード。ファイルがなければ作成され、あれば上書きされます。
"a"	追加書き込みモード。ファイルがなければ作成され、あれば追加書き込みされます。
"r+"	読み書きモード。ファイルがなければ失敗します。
"w+"	読み書きモード。ファイルがなければ作成され、あれば上書きされます。
"a+"	読み込みと追加書き込みモード。ファイルがなければ作成され、あれば追加書き込みされます。

・バイナリモード

"rb"	読み取りモード。ファイルがなければ失敗します。
"wb"	書き込みモード。ファイルがなければ作成され、あれば上書きされます。
"ab"	追加書き込みモード。ファイルがなければ作成され、あれば追加書き込みされます。
"rb+"	読み書きモード。ファイルがなければ失敗します。
"wb+"	読み書きモード。ファイルがなければ作成され、あれば上書きされます。
"ab+"	読み込みと追加書き込みモード。ファイルがなければ作成され、あれば追加書き込みされます。

―戻り値―
成功した場合は指定したファイルに対するファイルポインタが返ります。失敗した場合はNULLが返ります。

```
// ファイルオープン
FILE*   fp = fopen("Text\\Text00.txt", "r");
```

fopen関数が返すファイルポインタを使用してファイルの読み書きやクローズといった処理を行います。fopen関数はオープンに失敗するとNULLを返し、この場合はファイルの操作はできません。
fopen関数の2つ目の引数はオープンモードです。1～3文字の文字列で指定します。1文字目は読み書きの指定で、rは読み込み、wは書き込み、aは追加書き込みです。wとaの違いは、wでは既存のデータが存在する場合はそれを破棄して作り直しますが、aでは既存のデータが存在する場合にはデータの最後から追加書き込みを行います。
2、3文字目にはb(バイナリモード)もしくは+(更新モード)を指定することができます。バイナリモードを指定した場合は、ファイルをバイナリモードで処理します。bを指定しなかった場合は、テキストモードとして処理されます。また、+を指定した場合、1文字目がどんな指定でも更新モードになり、読み書きが可能になります。ただしa+およびab+は「読み込みと追加書き込み」を行うためのモードとなります。以前のデータはそのままで、データの最後から追加書き込みを行うことができます。

ファイルポインタ

ファイルの操作を行うために必要なFILE構造体がヘッダファイル<stdio.h>で定義されています。このFILE型で宣言されるポインタ型を、とくにファイルポインタと呼びます。ファイルポインタは、ファイルをオープンしたときに取得することができます。以後、ファイルに対する操作(読み込み、書き込み、クローズなど)は、すべてこのファイルポインタを各関数の引数に指定することになります。

ファイルのクローズ

fopen関数でオープンしたファイルはfclose関数でクローズします。その際、ファイルをオープンしたときに取得したファイルポインタを引数として使用します。
C言語では、fopen関数でオープンしたファイルをクローズし忘れても、プログラム終了時に自動的にクローズしてくれます。しかし、書き込み処理を行っている場合、プログラムでは書き込みが終わっていても、実際にはまだファイルに書き込まれていないデータやEOFコードの書き込みはクローズ時に行われます。ですから、オープンしたファイルは必ずクローズするようにしましょう。

```
// ファイルオープン
FILE*   fp = fopen("Text\\Text00.txt", "r");
if(fp == NULL) {
    // ファイルが読めない場合の処理(エラー処理)
    (省略)
}

// ファイル処理(読み込みなど)
    ⋮
    ⋮

// ファイルクローズ
fclose(fp);
```

ファイルの形式

ファイルの扱い方にはテキストモードとバイナリモードの２種類あります。テキストモードでファイルをオープンすると、Windowsの改行コード(C言語上では'\n')である「0x0D」「0x0A」の２バイトが、「0x0A」の１バイトに変換されます。これは、UNIXやLinuxなどのOSと互換をとるためです。さらに、ファイル中の「0x1A」という値は、ファイルの終端(EOF:End Of File)として認識され、以降のデータは読み込むことができません。これに対してバイナリモードでファイルをオープンすると、いっさいの変換は行われません。ファイルのデータがそのまま読み込まれ、書き込んだデータはそのままファイルに記録されます。
テキストファイルなど文字列をそのまま記録したファイルはテキストモード、ビットマップファイルなどはバイナリモードでオープンします。

ファイルから読み込む

ファイルの読み書きを行うために専用の関数が標準ライブラリに用意されています。一般的なファイル入力関数を以下に示します。

fgetc関数

fgetc関数は、ファイルから１文字読み込みます。戻り値は読み込んだ文字ですが、エラーの場合やファイルの終端になった場合にEOF(-1)を返すため、int型になっています。

```
// １文字(１バイト)読み込み
int   c = fgetc(fp);
```

fgets関数

fgets関数は、ファイルから文字列を読み込みます。指定した数から１引いた文字数か、文字列終端'\0'または改行コード'\n'が現れるまで文字列を読み込みます。読み込んだ文字列の最後に、終端文字'\0'を自動的に付加します。なお、読み込んだ改行コード'\n'は削除されません。

引数は、１つ目がデータを読み込む領域のアドレス、２つ目が最大読み込み文字数、３つ目がファイルポインタです。戻り値は、成功した場合にはデータを読み込んだ領域のアドレス、エラーやファイルの終端になった場合にはNULLを返します。

```
// １行読み込み(最大２５５文字)
char    Buffer[256];
fgets(Buffer, 256, fp);
```

fscanf関数

fscanf関数は、scanf関数をファイルから読み込むようにしたものです。１つ目の引数にファイルポインタを指定する必要がありますが、それ以外の書式はscanf関数と同じです。戻り値は、正しく読み込むことができた数で、ファイルの終端になった場合はEOFが返ります。fscanf関数は、変数や構造体のメンバを１度にまとめて読み込むときに使用します。

```
// ファイル読み込み
char    Buffer[256];
int     i, j;
fscanf(fp, "%d %s %d", &i, Buffer, &j);
```

これらの関数でファイルから読み込みを行うと、読み込んだ分だけファイルポインタが進められます。ファイルポインタがファイルの終端に達した場合は、それ以上読み込むことはできないので、関数はEOFまたはNULLを返します。

練習問題

１　以下のようなテキストファイルを作成し、"Text.txt"というファイル名で保存しましょう。

－Text.txt－

```
あんどうたただし 10 10
汚迦茂斗夜死鬼 20 5
娑琶愚蜘 1 1
ビル・ゲイII 100 100
```

２　Text.txtをテキスト読み込みモードでオープンするプログラムを作成しましょう。

３　２でオープンしたファイルをクローズする処理を追加しましょう。

４　２と３の間に、fgets関数を用いてText.txtから先頭の１行を読み込み、画面に表示する処理を追加しましょう。

５　４をファイルの終わりまで繰り返すように変更しましょう。

６　４を変更し、以下の変数に先頭のデータが読み込まれるようにしましょう。

```
char    name[32];  // 名前
int     HP;        // ヒットポイント
int     MP;        // マジックポイント
```

７　６で読み込んだデータを整形して出力する処理を追加しましょう。

８　６と７をファイルの終わりまで繰り返すようにしましょう。