

C 標準入出力ライブラリ

[Programming C]

- ・標準入出力 library とその header file `stdio.h` は低水準入出力 system call に対する柔軟な interface を提供する。
- ・低水準 file descriptor と同じ方法で使用する。
- ・低水準 file descriptor に相当するものは、stream と呼ばれ、構造体への pointer `FILE *` として実装されている。

fopen

```
#include <stdio.h>
FILE *fopen(const char *filename, const char * mode);
```

fopen 関数は、低水準 open に似た機能を持つ。

- ・fopen は、filename に指定された名前の file を open し、stream を関連付ける。
- ・mode は、file を open する方法を指定する parameter。
- ・成功すると、null ではない、`FILE *pointer` を返す。
- ・失敗した場合、`stdio.h` で定義されている、NULL 値を返す。

mode	内容
r	読み取り専用で open
w	書き込み用に open し、もとの file の長さを 0 に切り詰め
a	書き込み用に open し、file の最後に追加
r+	更新用に open する
w+	更新用に open し、もとの file の長さを 0 に切り詰める
a+	更新用に open し、file の最後に追加する

※ b はファイルが binary であることを示す。

fread

```
#include <stdio.h>
size_t fread(void *ptr, size_t size, size_t ntimes, FILE *stream);
```

file stream から data を読み込む。

- ・data は「stream から ptr が示す data buffer に読み込まれる。
- ・data record の指定は、record size、size っと、転送する record 数 ntimes によって行う。

fwrite

```
#include <stdio.h>
size_t fwrite(void *ptr, size_t size, size_t ntimes, FILE *stream);
```

fread とよく似た interface をもち、指定された data buffer から data record を出力 stream に書き込む。

- ・書き込みに成功した record 数を返す。

fclose

```
#include <stdio.h>
int fclose(FILE *stream);
```

指定された stream を close し、まだ書き込まれていない data を書き込む。

- ・標準入出力 library では data が、buffering されるため、data の書き込みを保障する必要がある場合、fclose を使用する。
- ・program が正常に終了した場合は、まだ open されているすべての file stream に対して fclose が自動的に呼び出される。

fflush

```
#include <stdio.h>
int fflush(FILE *stream);
```

file stream にあるすべての data を直ちに書き込みます。

- ・fclose を呼び出すと、暗黙的に fflush されます。

fseek

```
#include <stdio.h>
int fseek(FILE *stream, long int offset, int whence);
```

lseek system call の file stream 版で、stream 上で次に読み書きを行う場所を指定する。

- ・成功した場合に、0 を、失敗した場合に -1 を返す。

fgetc、getc、getchar

```
#include <stdio.h>
int fgetc(FILE *stream);
int getc(FILE *stream);
int getchar();
```

fgetc 関数は、file stream から次の byte を文字として返す。最後に達するかエラーだと、EOF を返す。

getc 関数は、macro として実装できる点を除けば、fgetc と同じ。

getchar 関数は、getc(stdin) と等価。標準入力から次の文字を読み込む。

fputc、putc、putchar

```
#include <stdio.h>
int fputc(int c, FILE *stream);
int putc(int c, FILE *stream);
```

```
int putchar(int c);
```

- fputc 関数は、出力 file stream に 1 文字を書き込み、書き込んだ文字を返す。
- error が発生した場合 EOF を返す。
- macro として実装できる点を除けば、putc と fputc は同じ。
- putchar は、putc(c, stdout) と等価

fgets、gets

```
#include <stdio.h>
char *fgets(char *s, int n, FILE *stream);
char *gets(char *s);
```

- fgets 関数は、入力 file stream から文字列を読み込む。
 - 改行文字に出会うか、n-1 文字が転送されるか、file の終わりに達するまで、s で指定された buffer に書き込まれる。
 - 改行文字に出会った場合、改行文字および、'\0' が buffer に書き込まれる。
 - 1 回の呼び出しで読み取られる文字数は最大 n-1 これは、buffer の最後に 文字列の終了を示す '\0' が含まれるため。
 - fgets は、成功すると buffer への pointer s を返す。
 - stream が file の末尾に達した場合、EOF indicator を set し、null pointer を返す。
 - 読み取り error が起きた場合、null pointer を返し、errno に error の内容を示す値を設定
-
- gets 関数は、標準入力から読み取りを行う点、改行文字を破棄する点以外は、fgets と同じ
 - gets は、転送 buffer があふれる可能性があるため、通常は fgets を利用すべき。

この本からの覚書。