

バイナリファイルの読みかた

千代浩司

高エネルギー加速器研究機構

素粒子原子核研究所

バイナリファイルの特徴

- 行というものが存在しない
- 何バイト読めばよいか自分で指定する
- キー関数: `fopen()`, `fread()`, `fclose()`

fread() - マニュアルを読む

NAME

fread, fwrite - binary stream input/output

SYNOPSIS

```
#include <stdio.h>
```

```
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
```

```
size_t fwrite(const void *ptr, size_t size, size_t nmemb,  
              FILE *stream);
```

DESCRIPTION

The function `fread()` reads `nmemb` elements of data, each `size` bytes long, from the stream pointed to by `stream`, storing them at the location given by `ptr`.

The function `fwrite()` writes `nmemb` elements of data, each `size` bytes long, to the stream pointed to by `stream`, obtaining them from the location given by `ptr`.

For non-locking counterparts, see `unlocked_stdio(3)`.

fread() - マニュアルを読む

RETURN VALUE

fread() and fwrite() return the number of items successfully read or written (i.e., not the number of characters). If an error occurs, or the end-of-file is reached, the return value is a short item count (or zero).

fread() does not distinguish between end-of-file and error, and callers must use feof(3) and ferror(3) to determine which occurred.

CONFORMING TO

C89, POSIX.1-2001.

SEE ALSO

read(2), write(2), feof(3), ferror(3), unlocked_stdio(3)

fread()

- FILE *fp = fopen("filename", "r");
- int n = fread(buf, size, nmemb, fp);
- sizeバイトをnmemb個読む。
- 16バイト読むとして組み合わせは
1 x 16, 2 x 8, 4 x 4, 8 x 2, 16 x 1
- 正常に読めたときの戻り値が違う
- ファイル終端付近で指定したバイト数だけ読めないことがある。これを検知するには
fread(buf, 1 /*byte*/, nmemb, fp)するのが楽

組み合わせで性能が変わるか？

glibc-2.xx/libio/iofread.c

```
_IO_size_t
_IO_fread (buf, size, count, fp)
    void *buf;
    _IO_size_t size;
    _IO_size_t count;
    _IO_FILE *fp;
{
    _IO_size_t bytes_requested = size * count;
    _IO_size_t bytes_read;
    CHECK_FILE (fp, 0);
    if (bytes_requested == 0)
        return 0;
    _IO_acquire_lock (fp);
    bytes_read = _IO_sgetn (fp, (char *) buf, bytes_requested);
    _IO_release_lock (fp);
    return bytes_requested == bytes_read ? count : bytes_read / size;
}
```

size * count バイト分読んでそうなので組み合わせは性能に関係なさそう

fread()使い方 (1/2)

```
int main(int argc, char *argv[])
{
    size_t n;
    char buf[1024];

    if (argc != 2) {
        usage();
        exit(EXIT_FAILURE);
    }

    FILE *fp = fopen(argv[1], "r");
    if (fp == NULL) {
        err(EXIT_FAILURE, "fopen error");
    }
}
```

fread()使い方 (2/2)

```
for ( ; ; ) {
    n = fread(buf, 1 /*byte*/, sizeof(buf), fp);
    if (n == 0) {
        if (feof(fp)) {
            break;
        }
        else if (ferror(fp)) {
            fprintf(stderr, "fread error¥n");
            exit(EXIT_FAILURE);
        }
        else {
            fprintf(stderr, "fread unkonwn error¥n");
            exit(EXIT_FAILURE);
        }
    }
    /* fread returns successfully */
    if (n != sizeof(buf)) {
        fprintf(stderr, "paritial read: %ld bytes (should be %ld bytes)¥n",
            n, sizeof(buf));
    }
    /* do something */
}
return 0;
}
```


- 実習 ex04 でやってみる。