C 低水準ファイルアクセス

[Programming C]

■ file descriptor

file descriptor	関連付け
0	標準入力
1	標準出力
2	標準エラー

■ write

```
#include<unistd.h>
ssize_t write(int filedes, const void *buf, size_t nbytes);
```

file descriptor 'filedes' に関連付けられた File に buf から最初の 'nbytes' byte を書き込み、実際に書き込まれた byte 数を返す。

・戻り値が 0 の場合には、データが書き込まれなかったことをあらわす。
・戻り値が -1 の場合には、エラーが発生したことを表す。
・global 変数 errno にエラーの内容を示す値が設定される。

```
#include <unistd.h>

int main()
{
  if (write(1, "Here is some data\n",18) != 18) {
    write(2, "A write error has occured on file descriptor 1\n", 48);
  }
  exit(0);
}
```

■ read

```
#include <unistd.h>
ssize_t read(int filedes, void *buf, size_t nbytes);
```

file descriptor 'filedes' に関連付けられた file から、'nbytes' byte までの data を data area 'buf' に読み込み、実際に読み込んだ byte 数を返す。

・read が 0 を返す場合には、読み取るものがないことを意味する。
・error が発生した場合、-1 を返す

```
#include <unistd.h>

int main()
{
  char buf[128];
  int nread;

  nread = read(0, buf, 128);
  if (nread == -1) {
    write(2, "A read error has occured\n", 26);
  }

  if ((write(1, buf, nread)) != nread) {
    write(2, "A write error has occured\n", 27);
```

```
    }
    exit(0);
}
```

■ open

```
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
int open(const char *path, int oflags);
int open(const char *path, int oflags, mode_t mode);
```

新しい file descriptor を作成するには、open system call を使う必要がある。

・呼び出しが成功した場合には、file descriptor を返す
・失敗すると、-1 をかえし、global 変数の errno に値を設定
・返された　file descriptor は、read や write で利用できる。
・Open する file または、device は、'path' parameter に指定する。
・Parameter 'oflags' には、file を open するときの動作を指定する。

file access mode 必須

mode	説明
O_RDONLY	読み取り専用
O_WRONLY	書き込み専用
O_RDWR	読み書き用

file access mode との bit 単位の論理和によって oflags に指定できる省略可能な mode

mode	説明
O_APPEND	file の最後に追加
O_TRUNC	file の長さを 0 に切り詰め、既存の内容を破棄
O_CREAT	必要に応じて、mode で指定された permission で file を作成
O_EXCL	O_CRET とともに指定された場合、既に file が存在していれば open は失敗。

初期 permission

open に O_CREAT flag を指定して file を作成する場合には､parameter を 3 つ指令する必要がある。
3 つ目の parameter の mode には、sys/stat.h header file で定義されている flag(以下参照) の bit 単位の論理和を指定。

flag	内容
S_IRUSR	owner の読み取り permission
S_IWUSR	owner の書き込み permission
S_IXUSR	owner の実行 permission

S_IRGRP	group の読み取り permission
S_IWGRP	group の書き込み permission
S_IXGRP	group の実行 permission
S_IROTH	その他 user の読み取り permission
S_IWOTH	その他 user の書き込み permission
S_IXOTH	その他 user の実行 permission

```
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>

int main() {
  int out;
  out = open("file.out", O_CREAT, S_IRUSR|S_IWUSR|S_IROTH);
}
```

```
-rw----r-- 1 root root    0  9月  9 20:10 file.out
```

umask

file 作成時に使用される system 変数で、file permission の mask をあらわす。

・open または create を呼び出して file を作成する場合、mode parameter と umask の値が比較され、mode で指定されている bit のうち、umask でもセットされている bit が clear される。
・3 桁の 8 進数を指定する

桁	内容
1 桁目	user の permission
2 桁目	group の permission
3 桁目	他の user の permission
値	内容
0	permission を許可
4	読み取り permission を不許可
2	書き込み permission を不許可
1	実行 permission を不許可

■ close

```
#include <unistd.h>
int close(int filedes);
```

close system call は、file descriptor filedes と file との関連付けを終了する。

・成功した場合、0 を返し、エラーが発生した場合、-1 を返す。

■ ioctl

```
#include <unistd.h>
int ioctl(int filedes, int cmd, ...);
```

ioctl は何でも屋的な関数で、device と その descriptor の振る舞いを制御したり、device が提供する service を設定するための interface を提供する。

・さまざまな device でそれぞれ独自の呼び出しが定義されている。

file 操作のためのその他の system call

■ lseek

```
#include <unistd.h>
#include <sys/types.h>
out_t lseek(int filedes, off_t offset, int whence);
```

file descriptor の読み取り / 書き込み pointer を set します。

・次に file の読み取り / 書き込み位置を指定することができる。
・pointer は faile 内の絶対一か、現在の位置または file の最後からの相対位置を設定できる。
・parameter offset には位置を指定し、whence には offset の以下の意味を指定
  ・SEEK_SET offset は絶対位置
  ・SEEK_CUR offset は現在位置から相対
  ・SEEK_END offset は file の末尾から相対

```
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>

int main()
{
  char c;
  int in, out;

  in = open("file.in", O_RDONLY);
  while(read(in, &c, 1) ==1) {
    write(1, &c, 1);
    lseek(in, 1, SEEK_CUR);
  }
  exit(0);
}
```

■ fstat, stat, lstat

```
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
int fstat(int filedes, struct stat *buf);
int stat(const char *path, struct stat *buf);
int lstat(const char *path, struct stat *buf);
```

・fstat は open された file descriptor に関連付けられている file の status 情報を返す。
・stat と lstat は、指定された名前の file に関する status 情報を返す。
・file が symbolic link の場合、lstat は link それ自体、stat は link が指し示す file の情報を返す。

構造体 stat の member

member	説明
st_mod	file の permission 情報
st_ino	file に関連付けられた i-node
st_dev	file が存在する device
st_uid	file の owner の user id
st_gid	file の owner の group id
st_atime	最終 access 時刻
st_ctime	mode、owner、group、または内容の最終変更時刻
st_mtime	内容の最終修正時刻
st_nlink	file への hard link の数

■ dup, dup2

```
#include <unistd.h>
int dup(int filedes);
int dup2(int filedes, int filedis2);
```

・dup は file descriptor を複製し、同じ file に異なる複数の file descriptor で access できるようにする。
・dup2 は file descriptor を copy する。

この本からの覚書。