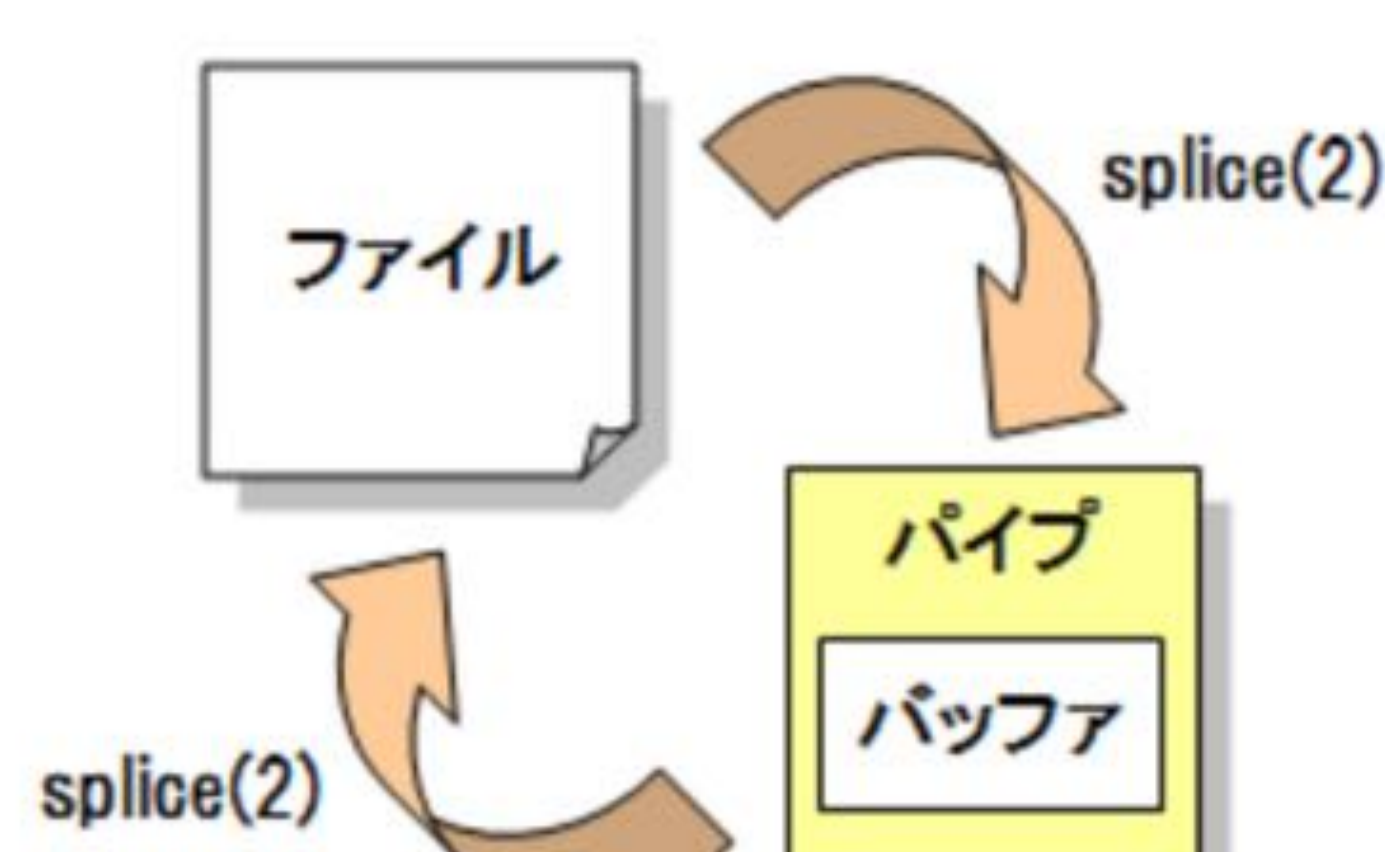


## 7.2. splice

spliceは、バッファを介さず直接ファイル/パイプに書き込むことができます。  
splice(2)は、linux-2.6.17 から追加された機能です。

### splice(2)

splice(2) は、ユーザ空間でバッファを用意する必要がありません。

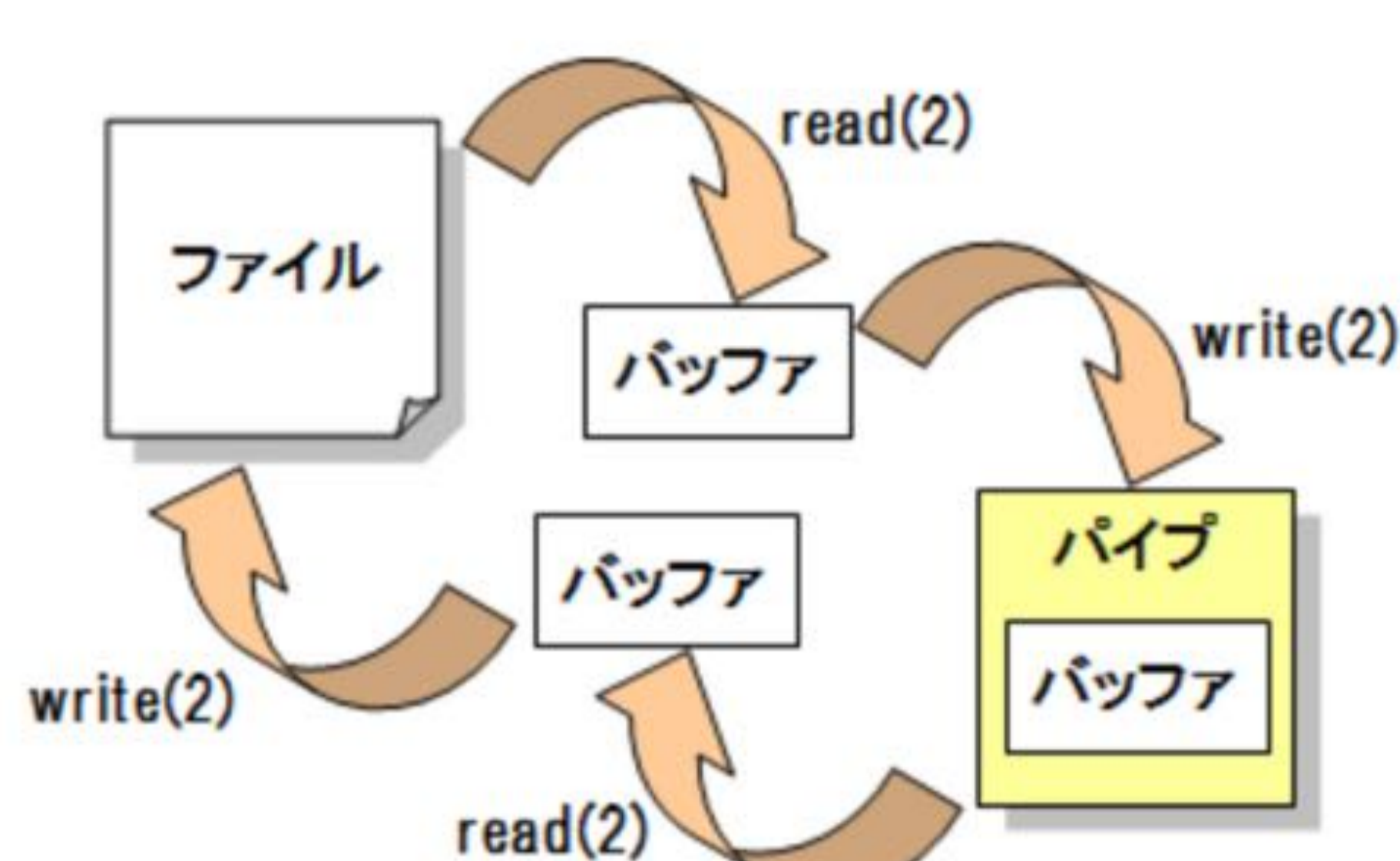


1. splice(2) を使用し、ファイルの内容をそのままパイプへ書き込む
2. splice(2) を使用し、パイプの内容をそのままファイルへ書き込む

splice(2) ではファイルとパイプの間で直接データの読み書きを行うことができます。  
そのため、read(2)/write(2) で必要だったユーザ空間上のバッファは、splice(2) では用意する必要がありません。

### read(2)/write(2) との違い

read(2)/write(2) を使用する場合、ユーザ空間でバッファを用意する必要があります。



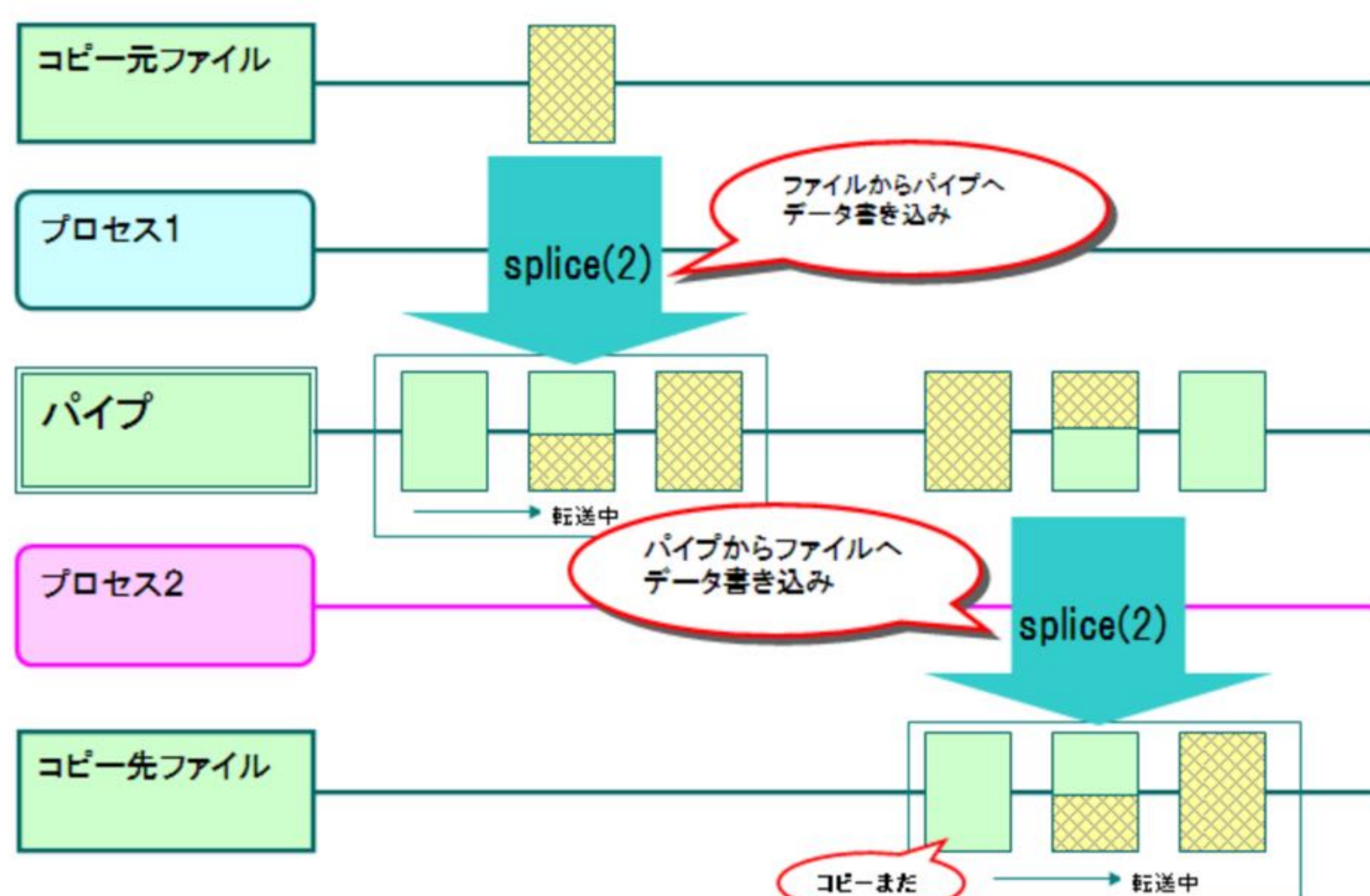
1. ファイルの内容を read(2) し、ユーザ空間のバッファに格納
2. バッファの内容をパイプへ write(2) する
3. パイプの内容を read(2) し、ユーザ空間のバッファに格納
4. バッファの内容をファイルへ write(2) する

read(2)/write(2) ではファイルとパイプの間で直接データの読み書きを行うことはできません。  
そのため、ユーザ空間でバッファを用意し、読み込んだデータを一時的にバッファに格納し、バッファの内容をパイプやファイルに書き込む必要があります。

### splice(2) の利用

splice(2) の利用方法について説明します。

以下の図は、プロセス 1 が splice(2) を用いてコピー元ファイルの内容を直接パイプへ書き込み、プロセス 2 が splice(2) を用いてパイプの内容を直接コピー先ファイルへ書き込んでいることを表しています。



- ファイルからパイプへデータ書き込み  
プロセス1がコピー元ファイルの内容をパイプへsplice(2)します。
- パイプからファイルへデータ書き込み  
プロセス2がパイプの内容をコピー先ファイルへsplice(2)します。

### ソースコード

```

#define _GNU_SOURCE
#include <stdio.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>

int main()
{
    int srcfd, destfd;
    int pipefd[2];
    size_t size, written = 0;

    pipe(pipefd);
    srcfd = open("./aaaa.txt", O_RDONLY);
    destfd = open("./bbbb.txt", O_WRONLY);

    size = 5;
    if (fork() != 0) {
        /* プロセス1 */
        splice(srcfd, NULL, pipefd[1], NULL, size, SPLICE_F_NONBLOCK);
    }
    else {
        sleep(1);
        /* プロセス2 */
        splice(pipefd[0], NULL, destfd, NULL, size, SPLICE_F_NONBLOCK);
    }
    return 0;
}

```

### splice(2) の引数指定

```

#define _GNU_SOURCE
#include <stdio.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>

int main()
{
    int srcfd, destfd;
    int pipefd[2];
    size_t size, written = 0;

    pipe(pipefd);

    if (fork() != 0) {
        /* プロセス1 */
        splice(srcfd, NULL, pipefd[1], NULL, size, SPLICE_F_NONBLOCK);
    }
    else {
        sleep(1);
        /* プロセス2 */
        splice(pipefd[0], NULL, destfd, NULL, size, SPLICE_F_NONBLOCK);
    }
}

```

flags

- SPLICE\_F\_MOVE: データのコピーではなく、移動を行う
- SPLICE\_F\_NONBLOCK: ノンブロックモードで転送する
- SPLICE\_F\_MORE: データのコピーの続きがあることを示す

※ splice(2) を使用する場合は、入出力のファイルディスクリプタのどちらかがパイプである必要があります。

このプログラムを実際に動かしてみましょう。

```

$ /bin/touch aaaa.txt ←ファイルを作成
$ /bin/touch bbbb.txt
$ echo "abcd" >& aaaa.txt ←aaaa.txtに5バイトのデータを書き込み
$ od -x aaaa.txt ←16進表記でデータを表示
$ od -x bbbb.txt
$ gcc splice.c ←コンパイル
$ ./a.out ←実行
$ cmp -l aaaa.txt bbbb.txt ←ファイルの比較

```

作成したプログラムによってファイルデータのコピーが行えたことがわかります。  
od(1)コマンドで実際のデータを見比べてもsplice(2)によってコピーできたことがわかります。