

リダイレクトとは

- Windows や Unix, Linux の端末において提供されている機能
 - 標準入力や標準出力などのストリームをユーザが指定した場所に結び付ける
 - 主な使い道
 - ファイルを指定することで、出力に関する処理を明記しなくても、簡単に入出力処理を実現できる
 - プログラムAの出力をプログラムBの入力にするなどの処理を容易にする(参考:パイプ処理)

復習: 標準入出力

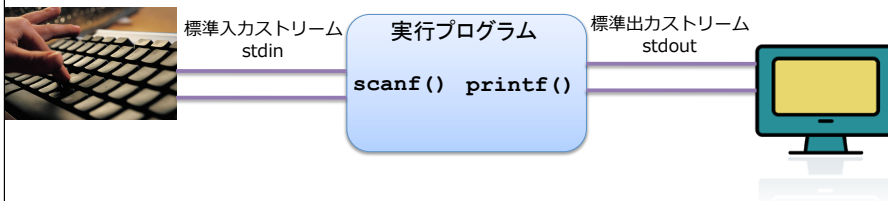
- 標準入力
 - 通常はキー入力に結びつけられている
 - C言語では `stdin` というファイルポインタで扱える
 - `scanf("%d", &n)` は `fscanf(stdin, "%d", &n)`と同じ
- 標準出力
 - 通常は画面出力に結びつけられている
 - C言語では `stdout` というファイルポインタで扱える
 - `printf("...")` は `fprintf(stdout, "...")`と同じ

`stdin, stdout`は`stdio.h`で定義された変数

復習: 標準入出力

キーボードから入力したデータは
標準入力ストリームを流れて
プログラムに読み込まれる
(`scanf()`, `getchar()`)

プログラムから出力したデータは
標準出力ストリームを流れて
画面上に表示される
(`printf()`, `putchar()`)



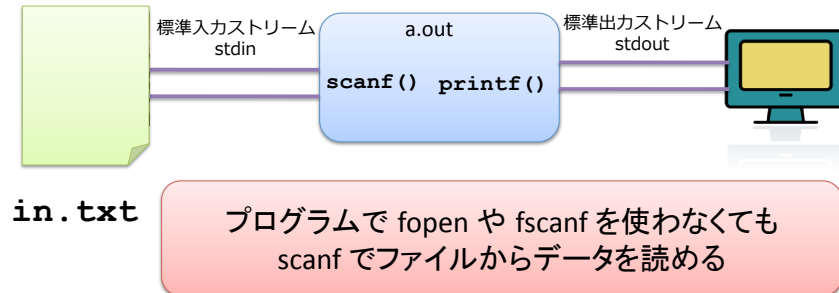
リダイレクト

- 標準入出力をファイルにする
 - \$ 実行ファイル < infile
 - キーボードからデータを読み込む代わりに、infileからデータを読み込む
 - \$ 実行ファイル > outfile
 - 画面にデータを表示する代わりに、outfileにデータを書き込む
 - \$ 実行ファイル < infile > outfile
 - infileからデータを読み込み、outfileにデータを書き込む

リダイレクト <

キーボードの代わりに
ファイル in.txt からデータを読む

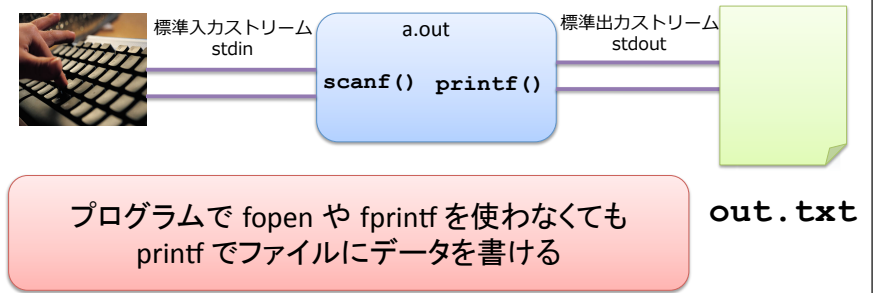
```
./a.out < in.txt
```



リダイレクト >

画面の代わりに
ファイル out.txt にデータを書く

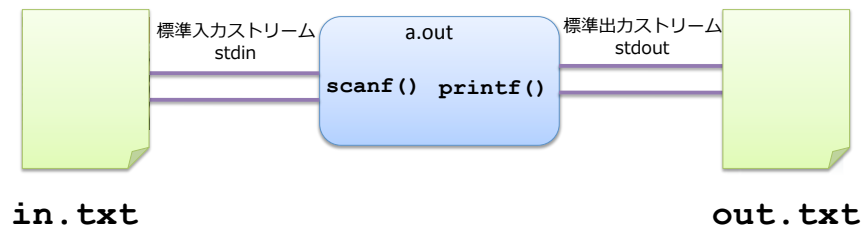
```
./a.out > out.txt
```



リダイレクト <>

ファイル in.txt からデータを読み
ファイル out.txt にデータを書く

```
./a.out < in.txt > out.txt
```



参考: 標準エラー出力

- 標準エラー出力
 - 標準出力と似ているが、エラーメッセージ用の出力
 - 通常は画面出力に結びつけられている
 - C言語では stderr というファイルポインタで扱える
 - エラーメッセージは次のように標準エラー出力に出力した方がよい

stderrはstdio.hで
定義された変数

```
fprintf(stderr, "ファイル %s が開けません\n",  
        filename);
```