

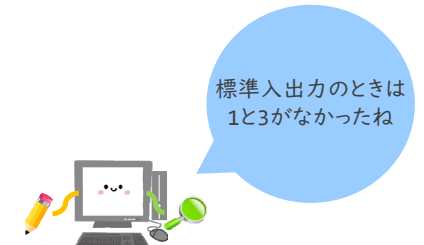
## プログラミング基礎演習A



ファイル入出力

### ファイルを使うときの手順

1. ファイルを開く  
ファイル名とモード(使い方)を指定してファイルポインタを作成
2. ファイルに書き込む / ファイルから読み込む  
ファイルポインタを使って読み書き
3. ファイルを閉じる  
ファイルポインタを削除



2

### ファイルオープン

❏ `fopen`(ファイル名, "モード")

- ◆ ファイルを開き, ファイルポインタを作成
- ◆ ファイル名: 変数名 または ""で囲んだ文字列
- ◆ 戻り値 : 成功したときは作成されたファイルポインタ, 失敗したときは0(NULL)
- ◆ ヘッダファイル`stdio.h`が必要
- ◆ よく使うモード
  - ◆ `r` : 読み込み
  - ◆ `w` : 書き込み
  - ◆ `a` : 追加書き込み

テキストを読み書きするときは`t`,  
バイナリを読み書きするときは`b`を  
モードの後に付ける

3

### ファイルクローズ

❏ `fclose`(ファイルポインタ)

- ◆ ファイルポインタを削除
- ◆ ヘッダファイル`stdio.h`が必要

```
FILE *fp;
```

```
fp = fopen("data.txt", "rt");
```

```
...
```

```
fclose(fp);
```

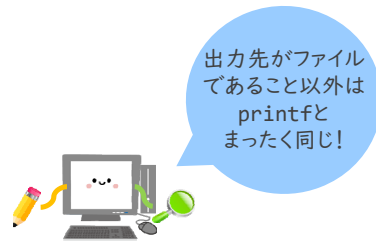


4

## fprintf関数によるファイル書込み

fprintf(ファイルポインタ, "フォーマット文字列", 変数, 変数, ...)

- ◆ フォーマット文字列中の変換指定子を後に続く引数が表す値に置き換えて文字列をファイルに書き込む関数
- ◆ ヘッダファイルstdio.hが必要



5

## ファイルオープンに失敗したときの対処

exit(EXIT\_FAILURE)

- ◆ プログラムを強制終了する関数
- ◆ 引数をEXIT\_FAILUREとすることで, 失敗したことを示す
- ◆ ヘッダファイルstdlib.hが必要

```
FILE *fp;
```

```
fp = fopen("data.txt", "rt");
if(fp == 0) {
    printf("ファイルdata.txtが開けませんでした\n");
    exit(EXIT_FAILURE);
}
```

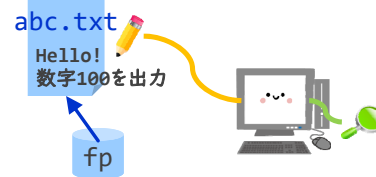
6

## プログラム例(1)

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

```
int main()
{
    FILE *fp;

    fp = fopen("abc.txt", "wt");
    if(fp == 0) {
        printf("ファイルabc.txtが開けませんでした\n");
        exit(EXIT_FAILURE);
    }
    fprintf(fp, "Hello!\n");
    fprintf(fp, "数字%dを出力", 100);
    fclose(fp);
}
```



7

## fscanf関数によるファイルからの読み込み

fscanf(ファイルポインタ, "フォーマット文字列", 変数アドレス, 変数アドレス, ...)

- ◆ ファイルから値を読み込み変数に代入する関数
- ◆ 改行, 空白文字までをひとかたまりのデータとみなす
- ◆ 戻り値 : 読み込めたときは代入した変数の数, ファイルの終端に達したときや, エラーが起きたときは-1
- ◆ ヘッダファイルstdio.hが必要
- ◆ ファイル終端 or エラーのチェックでは, 戻り値とEOF (-1を表す) を比較



↓ End Of Fileの略  
-1を表す

8

## 文字列から整数・実数への変換

### atoi(文字列)

- ◆ 文字列を整数に変換する関数
- ◆ 戻り値 : 文字列の表す整数
- ◆ ヘッダファイル`stdlib.h`が必要

### atof(文字列)

- ◆ 文字列を実数(double型)に変換する関数
- ◆ 戻り値 : 文字列の表す実数(double型)
- ◆ ヘッダファイル`stdlib.h`が必要

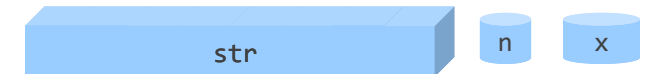
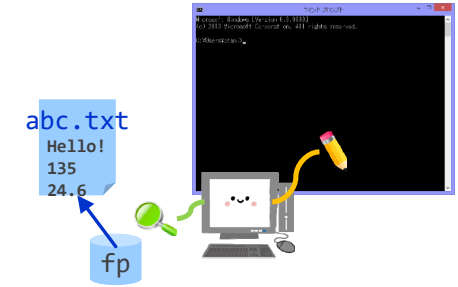
9

## プログラム例(2)

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

```
int main()
{
    FILE *fp;
    char str[8];
    int n;
    double x;

    fp = fopen("abc.txt", "rt");
    if(fp == 0) {
        printf("ファイルabc.txtが開けませんでした\n");
        exit(EXIT_FAILURE);
    }
```



10

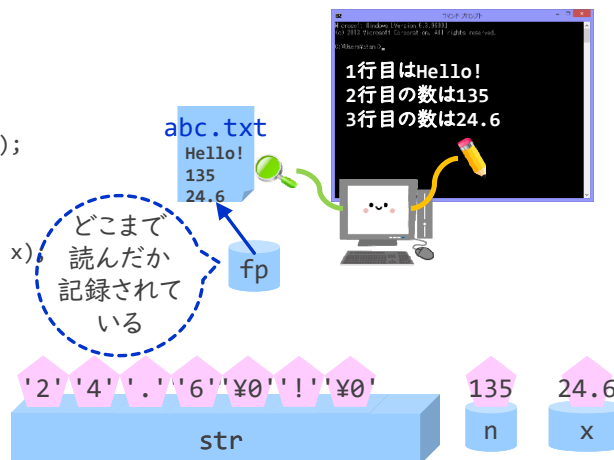
## プログラム例(2)

```
fscanf(fp, "%s", str);
printf("1行目は%s\n", str);

fscanf(fp, "%s", str);
n = atoi(str);
printf("2行目の数は%d\n", n);

fscanf(fp, "%s", str);
x = atof(str);
printf("3行目の数は%.1f\n", x);

fclose(fp);
}
```



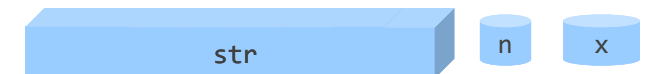
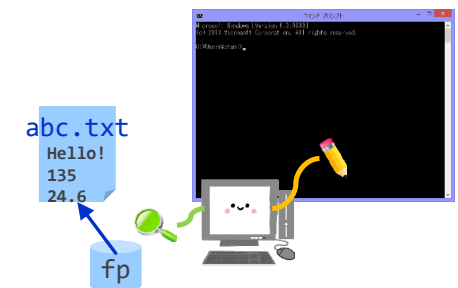
11

## プログラム例(3)

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

```
int main()
{
    FILE *fp;
    char str[8];
    int n;
    double x;

    fp = fopen("abc.txt", "rt");
    if(fp == 0) {
        printf("ファイルabc.txtが開けませんでした\n");
        exit(EXIT_FAILURE);
    }
```



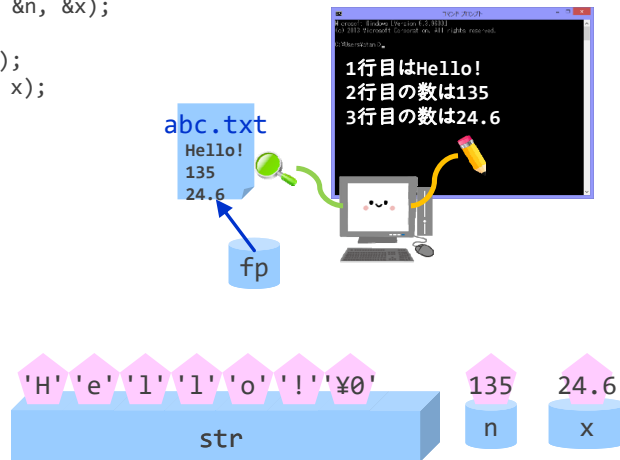
12

### プログラム例(3)

```
fscanf(fp, "%s%d%lf", str, &n, &x);
printf("1行目は%s\n", str);
printf("2行目の数は%d\n", n);
printf("3行目の数は%.1f\n", x);

fclose(fp);
}
```

改行までを  
1区切りとして  
一遍に  
読み込める



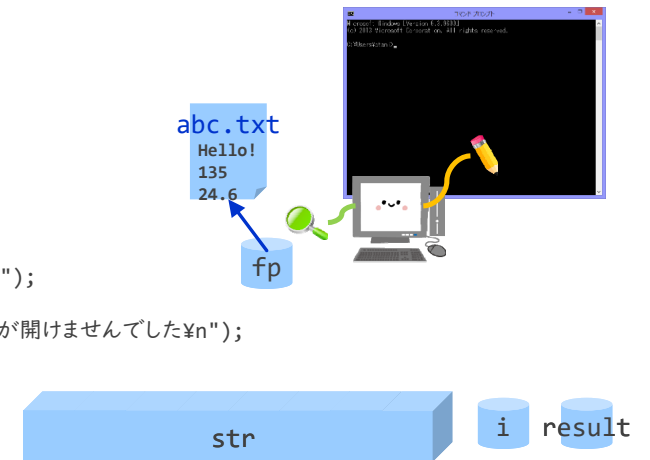
13

### プログラム例(4)

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

int main()
{
    FILE *fp;
    char str[8];
    int i;
    int result;

    fp = fopen("abc.txt", "rt");
    if(fp == 0) {
        printf("ファイルabc.txtが開けませんでした\n");
        exit(EXIT_FAILURE);
    }
}
```

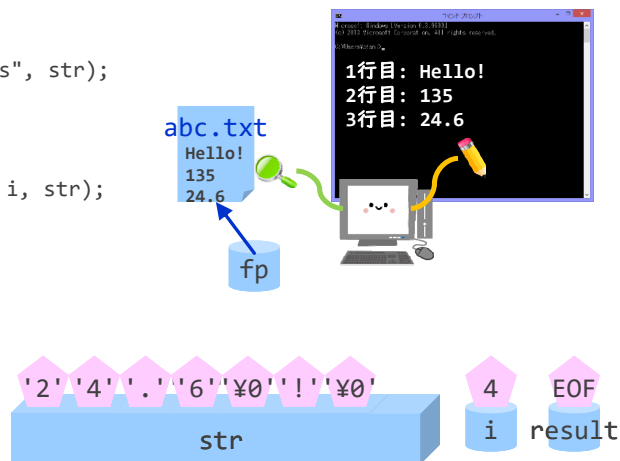


14

### プログラム例(4)

```
i = 1;
while(1) {
    result = fscanf(fp, "%s", str);
    if(result == EOF) {
        break;
    }
    printf("%d行目: %s\n", i, str);
    i++;
}
fclose(fp);
}
```

最後まで  
読んだら  
ループを  
抜ける



15

### fputs関数による文字列のファイル書込み

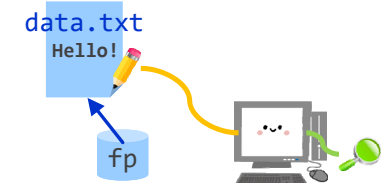
発展

- ◆ **fputs**(文字列, ファイルポインタ)
- ◆ 文字列をファイルに書き込む関数
- ◆ ヘッダファイル**stdio.h**が必要

```
FILE *fp;

fp = fopen("data.txt", "wt");
fputs("Hello!\n", fp);
fclose(fp);
```

自動的に改行されないの  
自分で書く



16

## fgets関数によるファイルからの文字列の読み込み

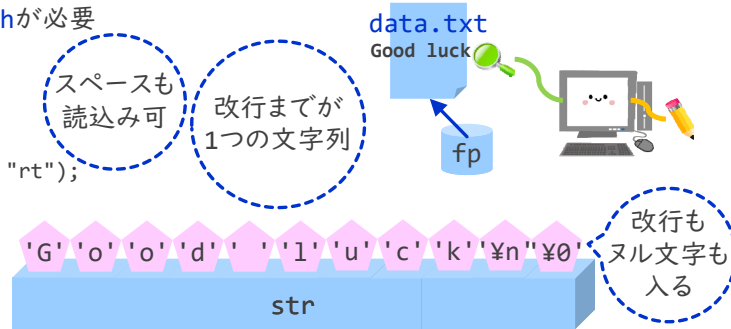
発展

🔧 fgets(文字列変数, 文字列変数の大きさ, ファイルポインタ)

- ◆ ファイルから文字列を1行読み込み変数に代入する関数
- ◆ 戻り値: 読み込めたときは文字列の先頭アドレス, ファイルの終端に達したときや, エラーが起きたときNULL (0)
- ◆ ヘッダファイルstdio.hが必要

```
char str[11];  
FILE *fp;
```

```
fp = fopen("data.txt", "rt");  
fgets(str, 11, fp);  
fclose(fp);
```



17

## 演習

九九の計算をすべて行い, 以下のような内容をファイルresult99.txtに書き出すプログラムを作成せよ。

プログラム名はe13とすること。

<ファイルresult99.txtの内容>

```
1 * 1 = 1  
1 * 2 = 2  
1 * 3 = 3  
(途中省略)  
1 * 7 = 7  
1 * 8 = 8  
1 * 9 = 9
```

```
2 * 1 = 2  
2 * 2 = 4  
(途中省略)  
9 * 8 = 72  
9 * 9 = 81
```

18