

前回課題の解説

(第11回: ファイル入出力)

担当: 佐藤

2018年12月10日(月)

課題11-1

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

```
int main()
```

```
{
```

```
    FILE *fp;
```

```
    char f[] = "data.txt";
```

```
    /* 書き込みモードでファイルオープン&エラー処理 */
```

```
    fp = fopen(f, "w");
```

```
    if (fp == NULL) {
```

```
        printf("%sをオープンできません\n", f);
```

```
        return 1;
```

```
    }
```

```
    /* ファイルクローズ */
```

```
    fclose(fp);
```

```
    return 0;
```

```
}
```

●ファイルオープンの手順

①ファイルポインタを宣言

②ファイルポインタをストリームにする

(fopen()の戻り値をファイルポインタへ代入)

③ エラー処理

data.txt

10 20 30

中身が消える!

◀ 明示的に構築したすべてのストリームについて行う
(stdinやstdoutは自動的にクローズされる)

課題11-2(1)

- プログラムの解説は前回のスライドで既に行っているため省略
- 補足
 - fscanf()は、書式に応じてデータの入力形式を変えることができる
 - sample11_2は「1行まとめて読み込む」形式をとっているが、書式を変更することで「数値ごとに読み込む」形式にすることも可能

1行まとめて読み込む

```
fscanf("%d %d %d", &v[0], &v[1], &v[2]);
```

data.txt

```
10 20 30
```



形式を統一!

数値ごとに読み込む

```
for (i = 0; i < 3; i++) {  
    fscanf("%d", &v[i]);  
}
```

%dは先頭の空白類文字を無視する。そのため、「_20」、「_30」は20, 30にそれぞれ変換される。よって、ループは3回行えばよい(空白文字を読み捨てるためのループは不要)

課題11-2(2)

別解(数値ごとに読み込む)

```
#include <stdio.h>

int main() {
    FILE *fp;
    int x, i;
    char f[] = "data.txt";

    /* 読み込みモードでファイルオープン&エラー処理*/
    fp = fopen(f, "r");
    if (fp == NULL) {
        printf("%sをオープンできません\n", f);
        return 1;
    }

    /* data.txtから数値の読み込みと表示を繰り返す */
    for (i = 0; i < 3; i++) {
        fscanf(fp, "%d", &x);
        printf("%d ", x);
    }
    printf("\n");

    /* ファイルクローズ*/
    fclose(fp);

    return 0;
}
```

課題11-3(1)

5行5数値のデータ入力を前提とした解答例

```
#include <stdio.h>

int main() {
    FILE *fpr, *fpw;
    int v[5], i;
    char fr[] = "multiRowData.txt", fw[] = "result.txt";
    double avg;

    fpr = fopen(fr, "r");
    if (fpr == NULL) {
        printf("%sをオープンできません\n", fr);
        return 1;
    }

    fpw = fopen(fw, "w");
    if (fpw == NULL) {
        printf("%sをオープンできません\n", fw);
        return 1;
    }

    for (i = 0; i < 5; i++) {
        fscanf(fpr, "%d %d %d %d %d", &v[0], &v[1], &v[2], &v[3], &v[4]);
        avg = (double)(v[0] + v[1] + v[2] + v[3] + v[4]) / 5;
        fprintf(fpw, "%.1f\n", avg);
    }

    fclose(fpr);
    fclose(fpw);

    return 0;
}
```

1行あたり5数値であるため、ループごとに書式を変える必要がない

「1行まとめて読み込み、その行に記載された数値(整数)を配列に格納→平均(実数)を計算→小数点第1位まで表示」を5回繰り返す

課題11-3(2)

可変行数可変個数値のデータ入力に対応する汎用性を持つ解答例

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

```
int main() {  
    FILE *fpr, *fpw;  
    int v[5], i, avg = 0;  
    char fr[] = "multiRowData.txt", fw[] = "result.txt", c;
```

vの要素数を1行あたりの
数値の最大個数以上に設定

```
    fpr = fopen(fr, "r");  
    if (fpr == NULL) {  
        printf("%sをオープンできません\n", fr);  
        return 1;  
    }
```

数値の後の'_'または
'\n'を格納する変数

```
    fpw = fopen(fw, "w");  
    if (fpw == NULL) {  
        printf("%sをオープンできません\n", fw);  
        return 1;  
    }  
    for (i = 0; fscanf(fpr, "%d%c", &v[i], &c) != EOF; i++) {  
        avg += v[i];  
        if (c == '\n') {  
            fprintf(fpw, "%.1f\n", avg / (double)(i + 1));  
            avg = 0;  
            i = -1;  
        }  
    }
```

このfor文の内容を独
力で解読してみよう!

```
    fclose(fpr);  
    fclose(fpw);  
  
    return 0;  
}
```

課題11-4

```
#include <stdio.h>

int main() {
    FILE *fpr, *fpw;
    char str[256], fr[] = "kadai11_4.c", fw[] = "copy11_4.txt";
    int i;

    if ((fpr = fopen(fr, "r")) == NULL) {
        printf("%sをオープンできません\n", fr);
        return 1;
    }
    if ((fpw = fopen(fw, "w")) == NULL) {
        printf("%sをオープンできません\n", fw);
        return 1;
    }

    for (i = 1; fgets(str, 256, fpr) != NULL; i++) {
        fprintf(stdout, "%2d: ", i); /* printf("%2d: ", i); */
        fprintf(stdout, "%s", str); /* printf(str); */
        fprintf(fpw, "%2d: ", i);
        fprintf(fpw, "%s", str);
    }

    fclose(fpr);
    fclose(fpw);

    return 0;
}
```

ループ変数としてだけでなく行数としても使う

行数と合わせるためiを0ではなく1から始める

出力先: ディスプレイ

出力先: 電子ファイル

繰り返し文の中でループ変数を使う場合、while文ではなくfor文を使うとよい