

プログラミング2

第十三回 標準ライブラリ関数の応用

機械システム系 劉 子昂(LIU Ziang)

第6回レポート課題について

- 課題はMoodleを見て下さい. 課題1～課題2まであります.
- Moodleのページからソースファイル(.c)を提出(アップロード)してください.

レポート再提出について

未提出のレポートの提出は
期末試験開始時刻まで受け付ける。

exitで強制終了

- 通常のプログラムは、main()の終端の'}'に出会うか、main()のreturn文に出会うと終了します
- exit()関数を使って任意の場所で処理終了

```
#include <stdlib.h>
exit(0); //正常終了
exit(1); //エラー時の終了
```

- returnとexit
 - main関数の中では同じ機能
 - 一般関数の中では異なる機能

exitで強制終了

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

int foo(void)
{
    return 1; //この関数を終了(mainに戻る)
    exit(1);  //プログラム終了
}

int main(void)
{
    foo();
    return 1; //この関数を終了(プログラム終了)
    exit(1);  //プログラム終了
}
```

ファイルの入出力

- 84 ファイルのオープンとクローズ
- 85 ファイル入出力関数
- 86 ファイル1行入出力
- 87 ファイル1文字入出力

標準入出力

- これまでのデータ入出力
 - 標準入力: キーボード
 - 標準出力: ディスプレイ
- ファイルの入出力

ファイル処理の手順

1. ファイルのオープン(開く)
2. ファイルの読み書き
3. ファイルのクローズ(閉じる)

ファイル処理の手順

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int ch;

    fp = fopen("myfile1.txt", "r");
    if (fp == NULL)
        return 1;

    ch = fgetc(fp);
    printf("ch=%c\n", ch);

    fclose(fp);

    return 0;
}
```

- `FILE`型構造体
- `fp`ストリームへのポインタ
- `fp`オープンされたファイルを操作するための識別子
- ストリーム（実体は`FILE`構造体）
入出力データを効率的に処理する
- `fopen()` ファイルオープン
- `fgetc()` ファイルから入力
- `fclose()` ファイルクローズ

ファイルのオープン

- 書式: `FILE *fopen(char *filename, char *openmode);`
 - `filename` で示すファイルを `openmode` で指定するモードで開く
 - 戻り値
 - 正常: ストリームへのポインタ `FILE *`
 - エラー: `NULL`

- `filename` ファイル名 (文字列)
 - 例: `"myfile1.txt"`

- `Openmode` モード

モード	処理	ファイルがない時	ファイルがある
"r"	読み込み (read)	<code>NULL</code> を返す	正常処理
"w"	書き込み (write)	新規作成	前の内容は捨てる
"a"	追加書き込み (append)	新規作成	前の内容の後に追加

ファイルのオープン

- `FILE *fopen(char *filename, char *openmode);`

ストリームへのポインタを入れる変数を宣言
(次の行で行っているfopen()の戻り値用)

```
FILE *fp;

fp = fopen("myfile1.txt", "r");
if (fp == NULL)
    return 1;
```

myfile1.txtを読み込み用にオープン.
操作するための情報をfpに入れる.
(fpは, myfile1.txtを扱うための識別子として考えればOK)

正常にオープンできない場合は, fopen()はNULLを返すので, エラー処理を記述.
(return 1;はプログラムをそこで強制終了し, そのときプログラム呼び出し側に1という値を返す, という意味です.)

ファイルのクローズ

- `int fclose(FILE *fp);`
 - 戻り値
 - 正常: `0`
 - エラー: `EOF`
- ストリームへのポインタ`fp`で示されるファイルをクローズする.

使用例

```
FILE *fp;  
  
fp = fopen("myfile1.txt", "r");  
if (fp == NULL)  
    return 1;  
  
fclose(fp);
```

作ってみよう！

1. ファイル“`text.txt`”を読み込み用として開き，もしそのファイルが開けたら「OK」，開けなかったら「NG」を画面に出力するプログラムを作れ.
ヒント: `fopen()`, `fclose()`を使う. p.219

※ファイル“`text.txt`”が存在する場合と，存在しない場合の両方の条件で動作を確認すること.
また，`text.txt`はNoEditor等を使って各自作ること.

1. 解答例

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    fp = fopen("text.txt", "r");
    if(fp==NULL){
        printf("NG\n");
        return 1;
    }
    printf("OK\n");
    fclose(fp);
    return 0;
}
```

実行結果 (text. txtがないとき)

NG

実行結果 (text. txtがあるとき)

OK

ファイル入出力関数

関数	説明
<code>fprintf(fout, format, 引数並び)</code>	ファイルfoutに書式formatで出力 例： <code>fprintf(fp, "%d", idt);</code>
<code>fscanf(fin, format, 引数並び)</code>	ファイルfinから書式formatで入力 例： <code>fscanf(fp, "%d", &idt);</code> ファイル終了でEOFを返す

- `fp` オープンされたファイルを操作するための識別子
- `"%d"` 書式
- `idt` 変数
- 標準入出力関数
 - `printf()` 標準出力(通常ディスプレイ)に書式formatで出力
 - `scanf()` 標準入力(通常ははキーボード)から書式formatで入力
- `printf(...)`は`fprintf(stdout,...)`と同じ.
- `scanf(...)`は`fscanf(stdin,...)`と同じ

ファイル入出力関数

関数	説明
fputc(ch, fp)	fpに文字chを書き込む
fgetc(fp)	fpから1文字入力する 最後はEOFを返す
fputs(s, fp)	fpに文字列sを書き込む
fgets(s, n, fp)	fpから最大n文字を入力する 最後はNULLを返す

宣言

```
FILE *fout, *fin;  
int c, n;  
char s[256];
```

文字

```
c = fgetc(fin);  
fputc(c, fout);
```

文字列

```
fgets(s, 256, fin);  
fputs(s, fout);
```


作ってみよう！

2. ファイル“`text.txt`”を読み込み用として開き、すべての行を読み込み、画面に表示せよ.

ヒント:

先ほどのプログラムに、`fgets()`を使った読み込み処理を追加する.

p.222-223が参考になる.

※ファイル`text.txt`はNoEditor等を使って事前に作っておく.

2. のプログラム例

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    char ss[256];
    fp = fopen("text.txt", "r");
    if (fp == NULL)
    {
        printf("NG\n");
        return 1;
    }
    printf("OK\n");
    while (fgets(ss, 256, fp) != NULL)
    {
        printf("%s", ss);
    }
    fclose(fp);
    return 0;
}
```



ファイル終了まで1行入力する.
fgets()は入力終了でNULLを返す.

作ってみよう！

3. ファイル“text.txt”を書き込み用として開き,
「Hello!」をファイル出力した後, 閉じるプログラムを作れ.

ヒント:

fopen(), fputs(), fclose()を使う.
先ほどのプログラムを修正する.
p.224,225が参考になる.

3. のプログラム例

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    fp = fopen("text.txt", "w");
    if(fp==NULL){
        printf("NG\n"); return 1;
    }
    printf("OK\n");
    fputs("Hello!\n", fp);
    fclose(fp);
    return 0;
}
```

実行後は, text.txtをエディタで開いて内容を確認してみよう！

その他

- 開くときのモードは読み込み用“r”または書き込み用“w”のどちらかを選択する.
- ファイルを開いた後で、モードを変えたければ、いったん閉じる必要がある.
例: “w”で開く→閉じる→“r”で開く→閉じる…
- 複数のファイルを同時に開くことは可能. ただし、各ファイル毎に異なるストリームポインタを使う.
- fclose()を実行しなくても、プログラムが終了する際、開かれているすべてのファイルは強制的に閉じられる.