

1 scanf 利用上の注意点

scanf は癖のある関数です。不用意に使うと奇妙な (と思える) 動作をします。ここではいくつか特徴的なことを説明します。

1.1 データを入れる引数はポインタである

scanf でデータを格納するための引数はポインタです。しかし、もしポインタを指定しなかったとき、たとえば、

```
int a;  
scanf("%d", a);
```

としても、C コンパイラはこれを正常なものとして処理します。厳密に言えば、これでも C の規則に合致しているからです。しかし、プログラムの意図とは異なります。このような誤りをしないためには、ユーザーが注意するしかありません。

1.2 空白を読み込まない

scanf の "%s" 変換では空白がくると読み込みを終わります。ですから、

```
this is string
```

といった文字列は読み込むことができません。この入力では「this」だけが読み込まれます。空白を含む文字列を読み込むときは、gets を使います。gets は「エンターキー入力」までを読み込みます。

1.3 %c なら空白も読む

scanf の変換文字の中で %c だけは例外です。%c は、空白も改行文字のすべて 1 文字読み込みします。

1.4 復改文字が残る

scanf で正しく入力しても最後の復改文字はバッファの中に残ります。そのため、おかしな動作になることがあります。たとえば次のプログラムは意図した通りに動きません。

1.5 復改文字が残る例

```
/*
   ex16.c
*/

#include <stdio.h>

int main()
{
    int a, c1, c2;

    printf("数値 = ");
    scanf("%d", &a);
    printf("a = %d\n", a);

    printf("文字 1 = ");
    c1 = getchar();
    putchar(c1);

    printf("文字 2 = ");
    c2 = getchar();
    putchar(c2);

    return 0;
}
```

1.6 実行結果

```
数値 = 123   . . . 数値入力

a = 123     . . . それを出力

文字 1 =    . . . 1 文字目を入力しようとしたら勝手に復改された

文字 2 = A  . . . 2 文字目は入力できる

A           . . . その文字を出力
```

見かけ上、最初の `getchar` が実行されず、おかしい出力が行なわれています。これは、最初の数値入力の際に入力された復改文字がバッファの中に残っていて、次に実行された `getchar` がその復改文字をバッファから読み込んだからです。scanf や `getchar` は、入力データをバッファリングしていますから、そのバッファ内のデータが空にならないとキーボードからの新たな入力動作にならないのです。

復改文字を読み込んでバッファが空になったために、2 回目の `getchar` ではキーボードからの入力動作が可能になっています。

改定新 C 言語入門シニア編 林晴比古著 ソフトバンクパブリッシング株式会社
324 頁