

ポインタの例 (関数)

山本昌志*

2005年6月23日

1 本日の学習内容

前回の授業に引き続き、ポインタについて学習する。その前にちょっとだけ復習をしておく。前回の授業をまとめると、以下のようになる。

- ポインタとは、アドレスを格納する変数のことである。
- ポインタの宣言には、型名とアスタリスク (*) を付ける。型名を指定するのは、ポインタが示すアドレスにはどのようなデータが格納されているかを示すためである¹
- 変数のアドレスを取り出すには、変数名の前にアンパサンド (&) をつける。&はアドレス演算子である。
- ポインタが示しているデータの値を取り出すためには、ポインタ変数の前にアスタリスク (*) をつける。*は間接参照演算子である。

前回の授業ではざっとこのようなことを学習したわけであるが、今回と来週にかけて、今まで知らないうちに使ってきたポインタの例を示す。本日は、以下の例を示す。

- 関数の参照渡し (アドレス渡し)

2 関数のデータの渡し方

ここでは、a と b のデータを入れ替える関数 swap() を通して、ポインタの使い方を学習する。

2.1 関数についての復習

以前の関数の学習を思い出してほしい。C 言語の関数とは、次のようなものであった。

- 特定の機能を関数としてまとめることができる。
- プログラムは関数が集合して出来上がっている。

*独立行政法人 秋田工業高等専門学校 電気情報工学科

¹間接参照演算子 * でポインタが指し示すメモリのアドレスに格納されているデータを取り出すときに型の情報が使われる。

- 関数が仕事をするためには、データを受け渡ししなくてはならない。
 - 呼び出し側のデータの変数のことを実引数と言う。
 - 呼び出された側のデータの変数を仮引数と言う。

データの受け渡しに、以下の2つの方法を学習した。

値渡し 実引数と仮引数の記憶領域が異なる。呼び出される関数がかう仮引数の記憶領域に実引数の値がコピーされる。従って、呼び出させた側の関数は、呼び出した側の記憶領域を操作することができない。

アドレス渡し 実引数はアドレスで、仮引数はポインターである。呼び出された関数は、ポインターをとおして、呼び出し側の関数の変数の記憶領域を操作する。

次節では、実際のプログラムを通して、これらのデータの渡し方の違いを考える。

2.2 値渡し

値渡しでは、呼び出された側は実引数の値を受け取る。そして、呼び出された関数の記憶領域(変数)に値を格納する。この場合、呼び出された側は、呼び出し側の記憶領域を使うことができない。なぜならば、呼び出された側は呼び出し側の記憶領域のアドレスが分からないからである。従って、値渡しでは、呼び出し側と呼び出された関数では、変数の名前が同一でも、全く異なる記憶領域を使いデータ进行处理するのである。

リスト1に値渡しのプログラム例を示す。関数 swap で a と b のデータを入れ替えようとしているが、それは不可能である。理由は、図1に示すプログラム実行状態のメモリーの様子を見てほしい。

- 値渡しの場合、呼び出し側と呼び出された側の関数では、各々独立の記憶領域を使う。

リスト 1: 値渡しの関数の例。これでは値が交換 (swap) できない。

```

1 #include <stdio.h>
2 void swap(int a, int b);
3
4 /*=====*/
5 /*  main 関数 */
6 /*=====*/
7 int main(void){
8
9     int a,b;
10
11     a=1;
12     b=2;
13
14     swap(a, b);
15
16     printf("a=%d\tb=%d\n", a, b);
17
18     return 0;
19 }
20
```

```

21  /*=====*/
22  /*  swap 関数(値渡し) */
23  /*=====*/
24  void swap(int a, int b){
25      int temp;
26
27      temp=a;
28      a=b;
29      b=temp;
30
31  }

```

実行結果

a=1 b=2

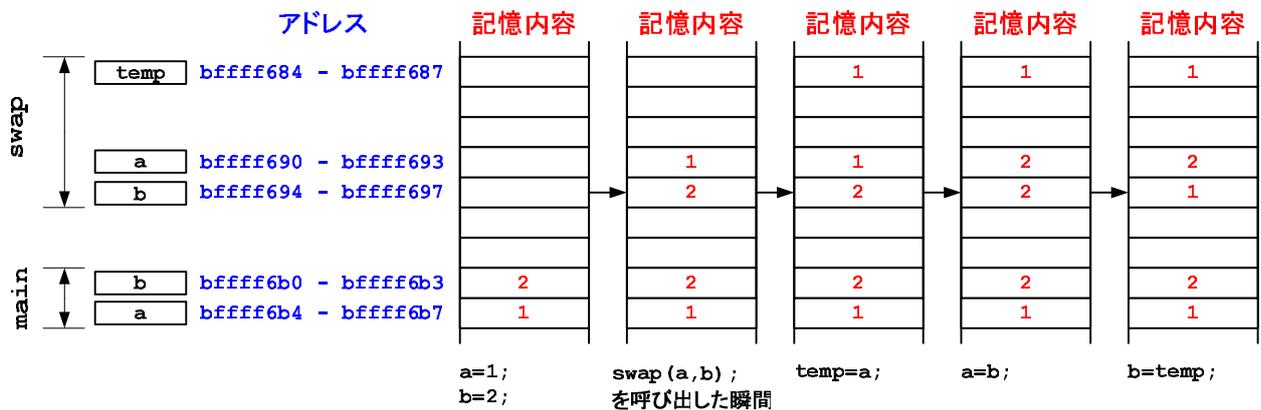


図 1: プログラムの各実行段階でのメモリの様子。メモリの内容が空白の部分は値が不明であることを示している。

2.3 アドレス渡し

アドレス渡しでは、呼び出された側は実引数のアドレスを受け取る。呼び出された関数は、その専用の記憶領域(ポインタ)にアドレスを格納する。呼び出された側は、呼び出し側の変数のアドレスが分かるので、それを操作することができる。

リスト 2 にアドレス渡しのプログラム例を示す。先ほどの値渡しと異なり、関数 swap で a と b のデータを入れ替えができています。データが入れ替えられた理由は、図 2 に示すプログラム実行状態のメモリの様子を見て考えて欲しい。

- アドレス渡しの場合、ポインタを通して、呼び出し側の記憶領域を操作できる。

リスト 2: アドレス渡し関数の例。呼び出された側はポインターを通して呼び出し側のメモリのアドレスを受け取る。そのため、呼び出し側のメモリの記憶領域を操作できる。

```
1 #include <stdio.h>
2 void swap(int *a, int *b);
3
4 /*=====*/
5 /*  main 関数 */
6 /*=====*/
7 int main(void){
8
9     int a,b;
10
11     a=1;
12     b=2;
13
14     swap(&a, &b);
15
16     printf("a=%d\tb=%d\n", a, b);
17
18     return 0;
19 }
20
21 /*=====*/
22 /*  swap 関数(アドレス渡し) */
23 /*=====*/
24 void swap(int *a, int *b){
25     int temp;
26
27     temp=*a;
28     *a=*b;
29     *b=temp;
30
31 }
```

実行結果

a=2 b=1

2.4 まとめ(ポインターとアドレス渡しの関係)

C言語では、関数によって、独立に記憶領域を持っている²。値渡しの場合、実引数の値が仮引数に渡され、それを使って処理をする。呼び出された側の関数は、呼び出し側の関数の変数の値を操作することができない。呼び出し側の関数の変数のアドレスが分からないからである。

一方、アドレス渡しの場合、呼び出し側の変数のアドレスが渡される。そのため、呼び出し側で呼び出し元の変数の値を書き換えることができる。

²関数の外で宣言されたグローバル変数は、全ての関数で共通の記憶領域となる。

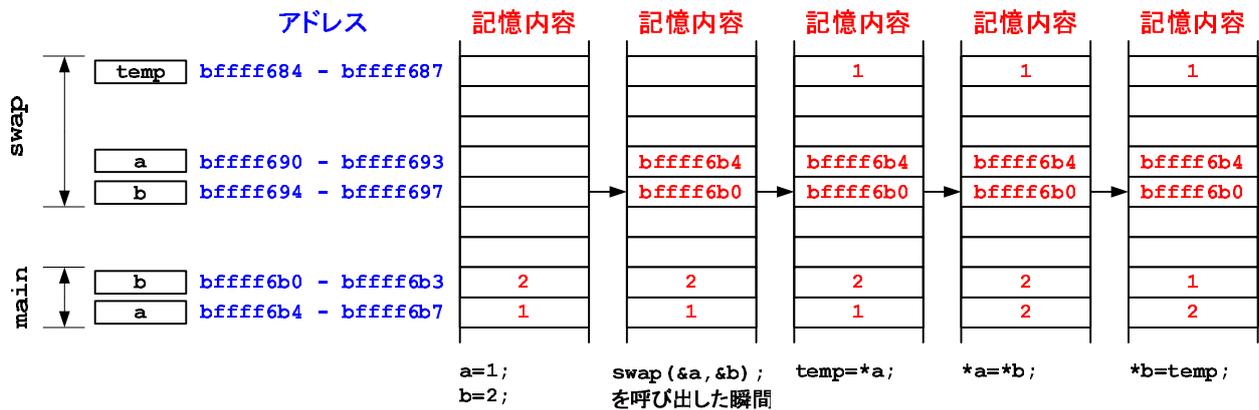


図 2: プログラムの各実行段階でのメモリの様子。メモリの内容が空白の部分は値が不明であることを示している。

3 レポート

3.1 内容

前期中間試験で、間違った部分の正解をきちんと書いて、提出すること。100 点の者は提出不要である。

- 計算過程などは省かないで、考え方が分かるように記述すること。
- 説明は論理的に分かり易く、正しい日本語の文で記述すること。

3.2 レポート 提出要領

提出方法は、次の通りとする。

- 期限 7月6日(水) PM 5:00 次回の授業の前日
- 用紙 A4
- 提出場所 山本研究室の入口のポスト
- 表紙 表紙を1枚つけて、以下の項目を分かりやすく記述すること。
 - 授業科目名「情報工学」
 - 課題名「前期中間試験の誤り訂正」
 - 2E 学籍番号 氏名
 - 提出日