

1 ポインタと配列

データは単純変数として使うことの他に、配列として使うこともできます。これに対応するように、ポインタも単純データを表現するとともに、配列データを表現することができます。配列とポインタは密接な関係があります。配列が添え字を使うことによってできる操作は、ポインタですべてできます。

1.1 配列名はポインタである

ポインタと配列の関係を述べる前に、まず配列そのものを理解することにしましょう。今、

```
char ss[5];
```

という配列が使われているとします。このとき各要素を参照するためには、

```
a      = ss[3];  
ss[4] = 0;
```

のように操作します。ss[3] は配列の要素です。では、ここで、

```
ss
```

とは何でしょうか。ss に値を代入することはできません。printf で ss の値を出力することもまずありません。しかし、ここで ss はちゃんと意味をもっています。それは、

-ss は配列 ss[0] から ss[4] の先頭アドレスを示すアドレスを示す定数である。

ということです。いま配列 ss が 1000 番地からはじまっているとすると、その関係は次のようになります。

ss	1000	->ss[0]	1000 番地
		ss[1]	1001 番地
		ss[2]	1002 番地
		ss[3]	1003 番地
		ss[4]	1004 番地

この関係図を見てわかるように、配列名はポインタの一種です。ただし、それは配列の先

頭アドレスを固定的に指している定数としてのポインタです。したがって、配列名 `ss` を使って配列の先頭アドレスを別のポインタに代入することができます。しかし、逆に配列例 `ss` に何かを代入するなどの変更行為はできません。たとえば、

```
char ss[3];
char *p;
p = ss;      //正しい。p は ss[0] のアドレスを指す。
ss = p;      //誤り
p = ss + 2; //正しい。p は ss[2] のアドレスを指す。
```

となります。なお、配列の先頭ではなく途中の要素のアドレス、たとえば `ss[3]` のアドレスをポインタに設定したい場合、配列名を使うなら、

```
p = ss + 3;
```

とします。また、ポインタ演算子`&`を使って

```
p = &ss[3];
```

とすることもできます。通常、配列名は先頭アドレス参照だけに用いられ、途中要素のアドレスを取り出すときには後者のほうがよく用いられます。

新 C 言語入門 ソフトバンクパブリッシング 林晴比古 175 頁