Structures (1A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Young Won Lim
8/28/15

Struct Declaration

C Programming Language

```
struct aaa {
        int     i;
        short   s;
        char    c;
} ;
```

structure type

definition

```
struct aaa  var;
```

var declaration

```
&var =
&var.i
&var.s
&var.c
```

var.i
var.s
var.c

Crating a structure in Octave

```
var.i = 10;
var.s = 1;
var.c = 'c';
```

```
var =
  scalar structure containing the fields:
    i =  10
    s =  1
    c = c
```

```
x.a = 1;
x.b = [1, 2; 3, 4];
x.c = "string";
```

Dynamic Naming (var)

```
x.a = 1;
x.b = [1, 2; 3, 4];
x.c = "string";
```

```
x =
  scalar structure containing the fields:
    a =  1
    b =
       1   2
       3   4
    c = string
```

```
a = "BB";
x.a = 1;
x.(a) = 2;
```

```
x =
  scalar structure containing the fields:
    a =  1
    BB =  2
```

Using struct()

```
x = struct(“AA”, 1, “BB”, 2, “CC”, 'c');
```

```
x =
  scalar structure containing the fields:
    AA =  1
    BB =  2
    CC = c
```

Nested Structures in C

```
struct point {
        int   x ;
        int   y ;
} ;
```

```
struct rect {
    struct point   p1 ;
    struct point   p2 ;
} ;
```

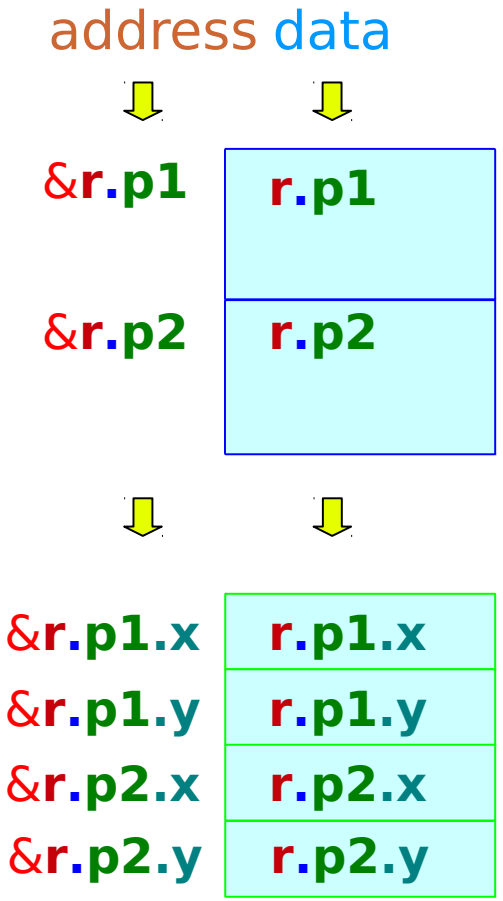
```
struct point   pa ;
```

```
struct point   pb ;
```

```
struct rect   r ;
```

```
pa.x   = 10 ;
pa.y   = 20 ;
pb.x   = 300 ;
pb.y   = 400 ;
```

```
r.p1   = pa ;
r.p2   = pb ;
```



Array Implementation

```
Var(1).i = 10;
Var(1).s = 1;
Var(1).c = 'a';

Var(2).i = 20;
Var(2).s = 2;
Var(2).c = 'b';

Var(3).i = 30;
Var(3).s = 3;
Var(3).c = 'c';
```

octave:31> **Var**
Var =
  1x3 struct array containing the fields:
    i
    s
    c

octave:31> **Var(1)**
ans =
  scalar structure containing the fields:
    i = 10
    s = 1
    c = a

octave:32> **Var(2)**
ans =
  scalar structure containing the fields:
    i = 20
    s = 2
    c = b

octave:33> **Var(3)**
ans =
  scalar structure containing the fields:
    i = 30
    s = 3
    c = c

Struct as a function argument

Struct returning function

Struct pointer as a function argument

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun