Arrays and Pointers

Arrays and Pointers Topics

- Arrays Converted to Pointers
- Array of Arrays
- Array of Pointers
- Multi-Dimensional Arrays

Arrays Converted to Pointers

- Arrays are converted to a pointer to the first element of the array when:
  - an array identifier (name) appears in an expression.
  - an array identifier (name) is passed to a function.
  - Except when the array identifier is used as an operand to the **sizeof** *operator*, in which case sizeof returns the size of the entire array, not the size of a pointer to the first array element (H&R, section 5.4.1).

Arrays Converted to Pointers

- Example

  type arr[N];

  arr[inx] is converted to *(arr+inx) by the compiler

  type * ptr = arr;

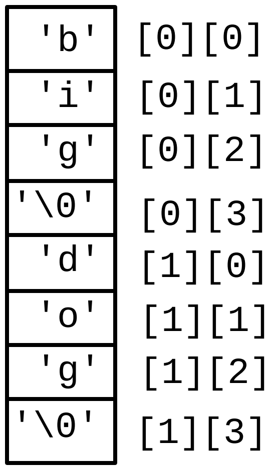
  The following are equivalent expressions:

  ptr[inx];

  *(ptr+inx)

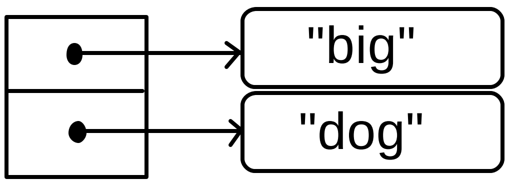
Array of Arrays

- Array of Arrays
  - char arrd[2][4]
    = { "big" , "dog" };



Array of Pointers

- Array of Pointers
  - char *arrp[2]
    = { "big" , "dog" };



Multidimensional Arrays

- Arrays in C are *row-major*, meaning that, given a multi-dimensional array
  - The outermost index varies most quickly
  - Elements associated with the outermost index are stored in memory first
- When a function parameter is declared as a multidimensional array, the extend of each dimension except the first must be declared.

```
void process_array(  int arr[][5][7] );
void func( void )
{
int test[3][5][7];
  process_array( test );
}
```

Multidimensional Arrays

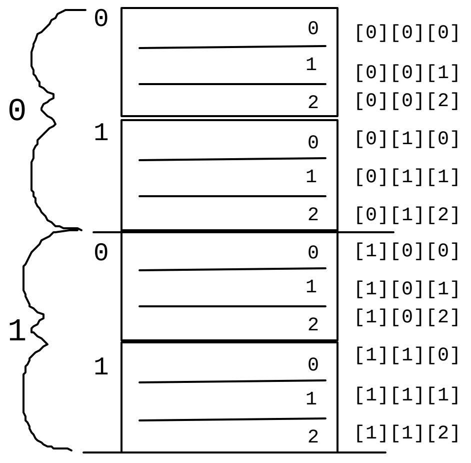
- Multidimensional Arrays
  - *type* arr[A0][A1][A2];
  - arr[n0][n1][n2];
  - *-- in pointer notation --*
  - *(*(*(arr + n0) + n1) + n2)

Multidimensional Arrays

- Example
  - *type* arr[A0][A1][A2];
  - From the ***compiler's*** perspective
    - address arr +
      (n0*A1*A2 + n1*A2 + n2)*sizeof(*type)*
    - arr[1][1][1] is address arr +
      10 * sizeof(*type*);
  - address is independent of A0!!

Multidimensional Arrays

- *type* arr[12][2][3];



0
1
0
1
0
1
0
1
2
0
1
2
0
1
2
0
1
2
[0][0][0]
[0][0][1]
[0][0][2]
[0][1][0]
[0][1][1]
[0][1][2]
[1][0][0]
[1][0][1]
[1][0][2]
[1][1][0]
[1][1][1]
[1][1][2]