6/29/2012

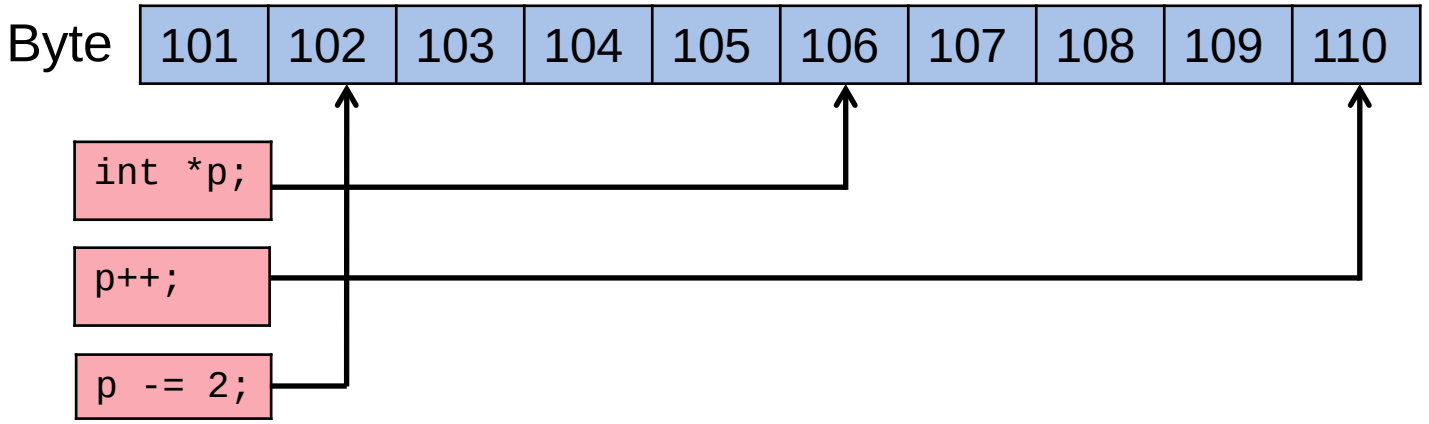
Pointer Arithmetic

Diana Wald

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Pointer Arithmetic

- Ability to modify a pointer's target address with arithmetic operation
- Adding or subtracting from a pointer moves it by a multiple of the size of the datatype it points to. For example, adding 1 to a pointer to 4-byte integer values will increment the pointer by 4 bytes.



- Pointer arithmetic cannot be performed on void pointers
- Use case in this presentation: **pointers and arrays**

Pointer Arithmetic and Array

The concept of an `array` and of a `pointer` is the same:

- Identifier of an `array` is equivalent to the address of its first element
- A `pointer` is equivalent to the address of the first element that it point to

```
int array[5];
int *pointer;
```

An array automatically allocate memory if its initialized

```
//declaration
int array[5];

//initialization
int array[] = {1,2,3,4,5};
```

Assignment operation:

```
pointer = array;
```

- `pointer` and `array` are equivalent and have the same properties
- **But:**

  value of `pointer` is changeable → **ordinary pointer**

  `array` will always point to first element of its type → **constant pointer**

```
array = pointer;
```

Pointer Arithmetic and Array

Array Indexing

- C++ doesn't really understand array indexing, except in declarations
- The compiler converts the array into a pointer

```
array[n] == *(array + n)
```

Reason why an array element count from zero

- Commutative law:

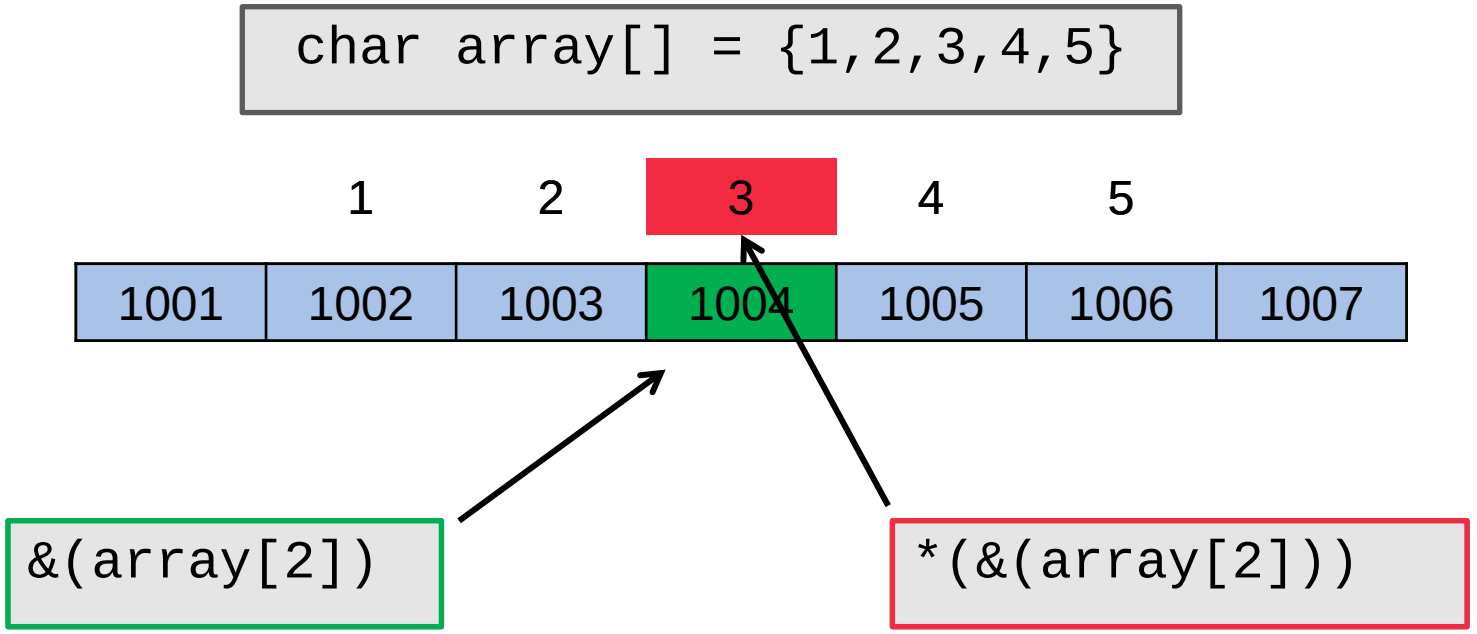
```
array[n] == *(array + n)
```

```
array + n == n + array
```

```
array[n] == n[array]
```

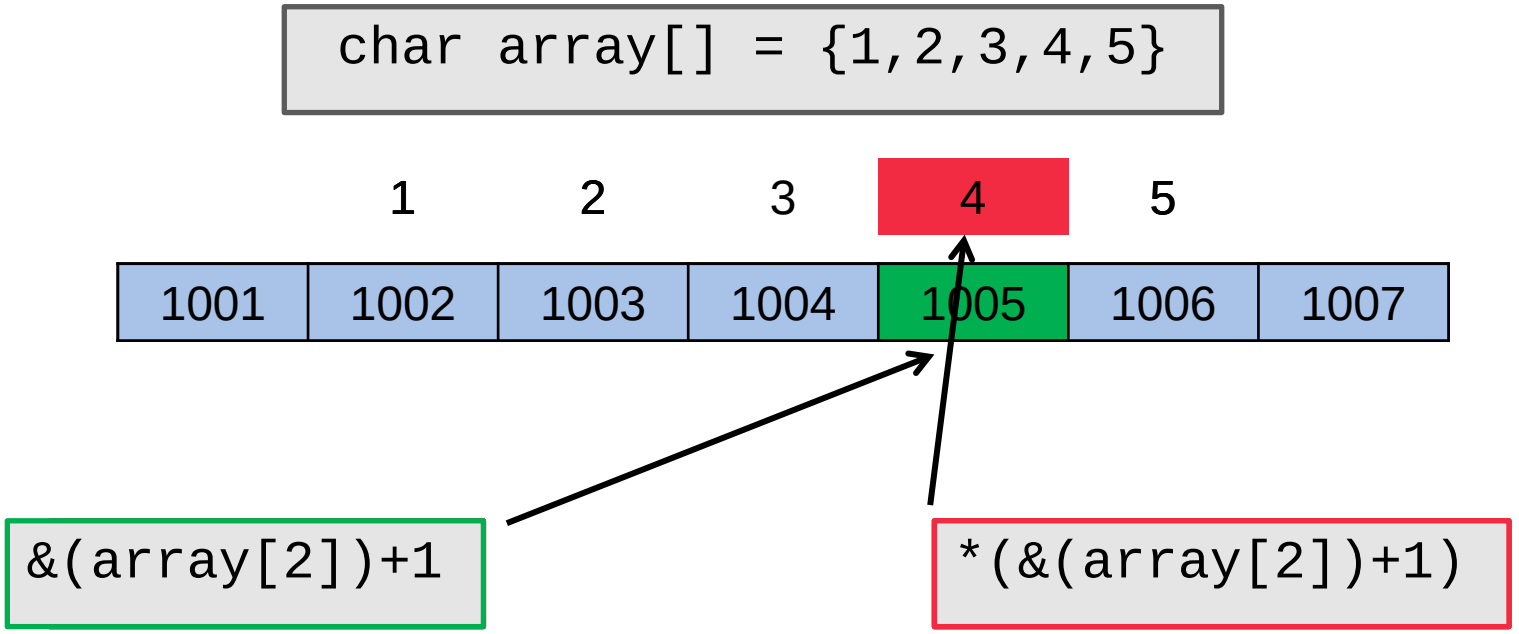
Pointer Arithmetic and Array

How Pointer Arithmetic works:



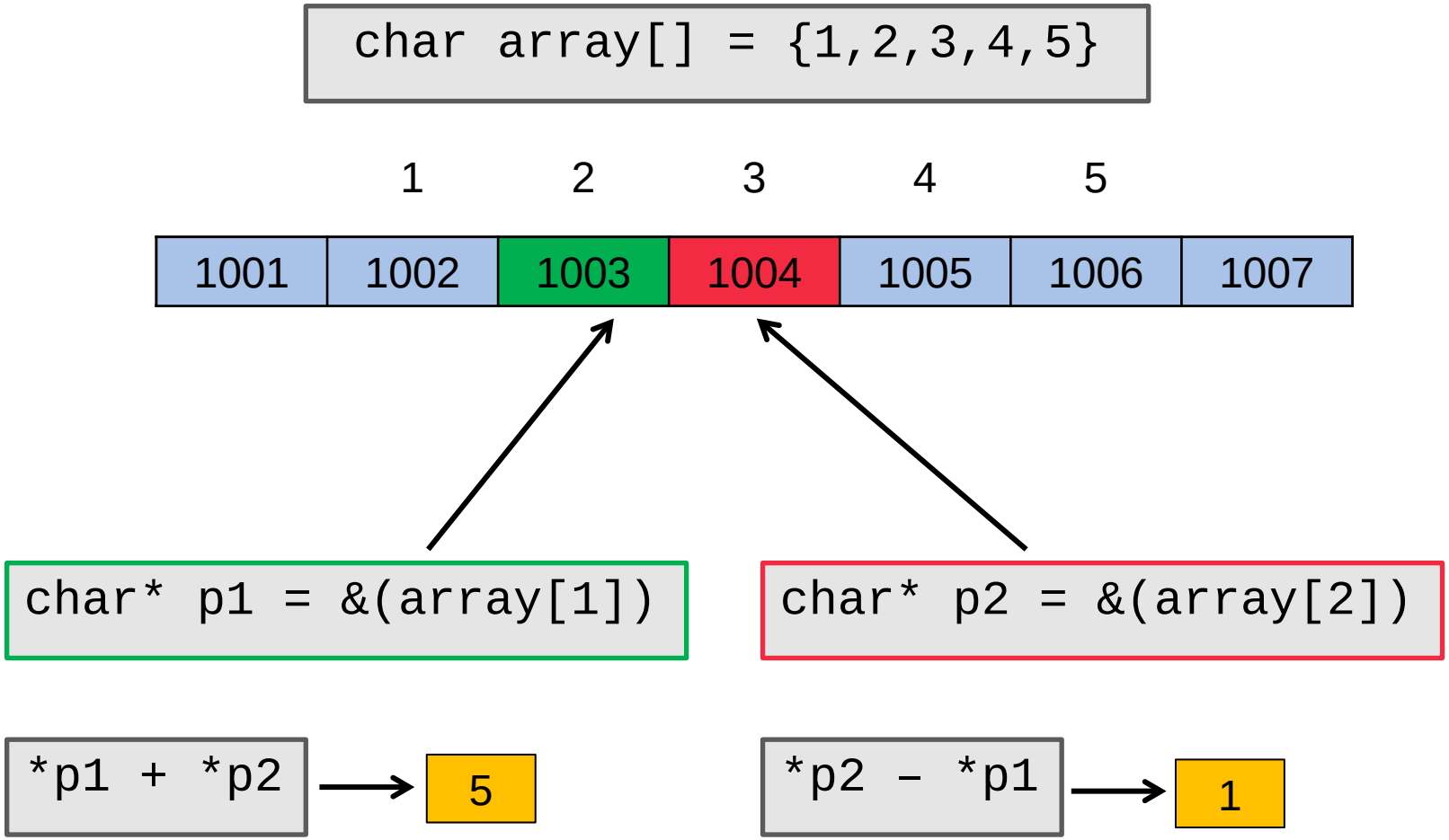
Pointer Arithmetic and Array

How Pointer Arithmetic works:



Pointer Arithmetic and Array

How Pointer Arithmetic works:



Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
*pointer = 9;
pointer = pointer - 3;
*pointer = 6;
*(pointer+2) = 7;
```

Name	Type	Value
array[0]	int	0
array[1]	int	0
array[2]	int	0
array[3]	int	0
pointer	int*	
*pointer	int	

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
```

Name	Type	Value
array[0]	int	0
array[1]	int	0
array[2]	int	0
array[3]	int	0
pointer	int*	address to array[2]
*pointer	int	

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
```

Name	Type	Value
array[0]	int	0
array[1]	int	0
array[2]	int	3
array[3]	int	0
pointer	int*	address to array[2]
*pointer	int	3

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
```

Name	Type	Value
array[0]	int	0
array[1]	int	0
array[2]	int	3
array[3]	int	0
pointer	int*	address to array[3]
*pointer	int	

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
*pointer = 9;
```

Name	Type	Value
array[0]	int	0
array[1]	int	0
array[2]	int	3
array[3]	int	9
pointer	int*	address to array[3]
*pointer	int	9

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
*pointer = 9;
pointer = pointer - 3;
```

Name	Type	Value
array[0]	int	0
array[1]	int	0
array[2]	int	3
array[3]	int	9
pointer	int*	address to array[0]
*pointer	int	

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
*pointer = 9;
pointer = pointer - 3;
*pointer = 6;
```

Name	Type	Value
array[0]	int	6
array[1]	int	0
array[2]	int	3
array[3]	int	9
pointer	int*	address to array[0]
*pointer	int	6

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
*pointer = 9;
pointer = pointer - 3;
*pointer = 6;
*(pointer+2) = 7;
```

Name	Type	Value
array[0]	int	6
array[1]	int	0
array[2]	int	7
array[3]	int	9
pointer	int*	address to array[2]
*pointer	int	7

Pointer Arithmetic and Array - Example

```
int array[]={0,0,0,0};
int *pointer;

pointer = &(array[2]);
*pointer = 3;
pointer++;
*pointer = 9;
pointer = pointer - 3;
*pointer = 6;
pointer = pointer + 2;
*pointer = 7;
```

Name	Type	Value
array[0]	int	6
array[1]	int	0
array[2]	int	7
array[3]	int	9
pointer	int*	addres[2]
*pointer	int	

array = {6, 0, 7, 9}

References

- Book: C++ Entpackt, Herbert Schild (mitp-Verlag, 2001)
- Webpages: http://www.cplusplus.com/doc/tutorial/pointers/