

Pointers

Héctor Menéndez¹

AIDA Research Group
Computer Science Department
Universidad Autónoma de Madrid

November 21, 2013

¹based on the original slides of the subject

Index

1 Introduction

Index

1 Introduction

Introduction

- A pointer is a data type whose value refers directly to (or "points to") another value stored elsewhere in the computer memory using its address.
- There is a value reserved for indicating that the pointer does not refer to a valid object, it is called NULL pointer.

Introduction: declaration syntax

- Its declaration syntax is:

```
data_type * pointer_name;
```

- A pointer which does not have any address assigned to it is called a **wild pointer**.
- Any attempt to use an uninitialized pointer can cause unexpected behaviour, either because the initial value is not a valid address, or because using it may damage other parts of the program.
- The result is often a segmentation fault or storage violation

Introduction: example

```
char c = 'a';  
char *ptrc=NULL;  
int *ptri;
```

1011	...	
1007	NULL	ptrc
1003	?	ptri
1002	a	c
1001	...	

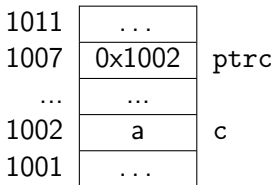
& Operator

- & Operator provides a variable starting memory address.
- It is used as follows:

```
char c = 'a';  
char *ptrc=NULL;
```

```
ptrc = &c;
```

- Now, ptrc “points to” c.



* Operator

- The dereferencing operator * refers to the value of the variable which the pointer is pointing to.
- It is used as follows:

```
char c = 'a';  
char *ptrc=NULL;
```

```
ptrc = &c;  
*ptrc = 'b';
```

- Now, ptrc “points to” c.

1011	...	
1007	0x1002	ptrc
...	...	
1002	b	c
1001	...	

= Operator

- The assignment operator = is used to assign values to pointers.
- Pointers must be initialized to NULL.

```
char *ptrc=NULL;
```

= Operator

- Directly assignation:

```
int *pointer = 0x1F3CE00A;
```

- Relative assignation:

```
int c = 4;  
int *pointer = &c;
```

- Pointer assignation:

```
int c;  
int *pointer = &c;  
int *pointer2 = pointer;
```

= Operator

- The assignment operator = is used to assign values to pointers.
- Pointers must be initialized to NULL.

```
char *ptrc=NULL;
```

Example

```
#include<stdio.h>

int main (int argc, char *argv[])
{
    int variable = 1;
    int *pointer = NULL;

    pointer = &variable;

    printf("Variable: %d", variable);
    printf("Variable: %d", *pointer);
    printf("Variable address: %p", &variable);
    printf("Variable address: %p", pointer);
}
```

Example: Common Errors

```
#include<stdio.h>

int main (int argc, char *argv[])
{
    int x = 10;
    int * ptr1 = NULL;
    double y = 0.5;
    double * ptr2 = NULL;

    ptr1 = &x;
    ptr2 = &y;
    ptr1 = ptr2; // ERROR

}
```

Example: Common Errors

```
#include<stdio.h>

int main (int argc, char *argv[])
{
    int * ptr1;
    *ptr1 = 3; // ERROR
}

```

Example: Common Errors

```
#include<stdio.h>

int main (int argc, char *argv[])
{
    char c;
    int * ptr = &c;
    ptr = 'a'; // ERROR
}
```

Example: Common Errors

```
#include<stdio.h>

int main (int argc, char *argv[])
{
    int * ptr1 = NULL;
    *ptr1 = 3; // ERROR
}
```