**Introduction To Pointers**

**Hour 11**

**Objectives**
- Pointer variables: address and content
- Pointer and address operators: * and &
- Defining pointer variables
- Dereference and indirection
- Address of operator

---

**Variables: Address vs Content**

Address is intrinsic, content may change

- **A left value (aka l-value)**
  - is a value that can be on the left hand side of =
  - is the address of a variable

- **A right value (aka r-value)**
  - is a value that can be on the right hand side of =
  - can be any valid expression (i.e., constant, variable, function call, or any of these joined with operators and grouped with parentheses)

```c
i = 10; /* i is an l-value, 10 is an r-value */
j = i;    /* j is an l-value, i is an r-value */
```

```plaintext
i  10 0xffffffff00
r-value of i is 10 (i.e., the content of i)
l-value of i is 0xffffffff00 (i.e., the address of i)
```

---

**Pointer and Address Operators**

Pointers are variables that contain memory addresses

- `*` pointer-type specifier (variable definitions & declarations)
  - `type t;`
  - `type* p;` /* same as below */
  - `type p;` /* same as above */

- `&` address of operator returns the address of variables

- `*` dereference or indirection operator (expressions)

- **Operator relationships**
  - `p` is assignment compatible with `t` (i.e., `t = *p`)
  - `&t` is assignment compatible with `p` (i.e., `p = &t`)
  - `*(&variable)` / variable

- **Special pointers (that point to nothing)**
  - `NULL` (defined in stdio.h, stdlib.h and others)
  - `0` (zero; back-ported from C++)

---

**Address Operators**

Memory content vs memory address

```c
int i;
int* p = &i;
```

```plaintext
i  0xffffffff00010 i is undefined
p  0xffffffff00010 p is 0xffffffff00010
```

```c
i = 10;
```

```plaintext
i  10 0xffffffff00010 i is 10
p  0xffffffff00010 p is 0xffffffff00010
```

```c
*p is now 10
```

```plaintext
*p is now 10
```

---

**Pointer Syntax**

Simple examples

```c
int i;
int* ip1;
int* ip2;
i = 10;
ip1 = &i; /* *ip1 is 10 */
ip2 = ip1; /* *ip2 is 10 */
```

```c
printf("%p   %p   %x\n", &i, ip1, ip2);
```

```c
printf("%d   %d   %d\n", i, *ip1, *ip2);
```

```c
ip1 = NULL; /* points at nothing */
ip2 = NULL; /* ditto */
```

```c
while (*ip1 == NULL) ....
while (ip2 != 0) ....
if (ip1 == ip2) ....
if (ip1 != ip2) ....
```

---

Copyright © 1998-2002 Delroy A. Brinkerhoff. All Rights Reserved.