C under Linux

Dr. Naeem Odat

Department of Computer and Communications Engineering

C Operators

const, define and operators

Constants and define

const qualifier

- Can be applied to the declaration of any variable to specify that its value will not be altered.
- const double pi = 3.14159265358979323846264338327 95;

Intialize a variable

- The variables declared in C, initially contains a garbage value.
- To initialize a variable,

```
int flag = 10;
float amount = 1024.750;
```

Define directive (macro definition)

Define

- It is a text substitution.
- No semicolon at the end
- Syntax:

```
#define MAX 30
#define SUM(a,b) (a+b)
```

Example - output is not as expected (why?)

```
#include <stdio.h>
#define SUM(a, b) a + b
int main(){
    int nRes = 0;
    nRes = 5 * SUM(2, 3) ;
    printf("%d\n", nRes);
    return 0;
}
```

Define directive (macro definition)

Example - output is not as expected (why?)

```
#include <stdio.h>
#define  PRO (a, b)  (a * b)
int main(){
    int nRes = 0;
    nRes =PRO(3-2, 3+2);
    printf("%d", nRes);
    return 0;
}
```

Operators

Types of operators

- Arithmetic
- Relational
- Logical
- Binary
- Increment / Decrement
- Assignment

Arithmetic operators

- +, -, *, /,and %.
- Example:

```
#include<stdio.h>
int main(){
    int nRes;
    nRes = 2 + 3 * 4 - 2;
    printf("2 + 3 * 4 - 4 / 2 = %d", nRes);
    return(0);
}
```

Relational and logical operators

Relational operators

$>$, $>=$, $<$, $<=$, $==$ and $!=$.

Logical operators

&& (and), || (or) and ! (not)

Example

```
#include<stdio.h>
int main(){
    char x = 'u';
    if (('a'==x||'e'==x||'i'==x||'o'==x||'u'==x)||
    ('A'==x||'E'==x||'I'==x||'O'==x||'U'==x))
        printf("Vowel");
    else
        printf("Other Character");
    return(0);
}
```

Boolean and bitwise operators

Boolean operators

- && and.
- || or.
- ! not.

Bitwise operators

- & and.
- | or.
- ^ xor
- $\sim$ not.
- << shift left.
- >> shift right.