



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Charqueadas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Operadores Aritméticos e Lógicos em C

Programação Estruturada

Prof. André del Mestre

Operadores Aritméticos

Operadores Aritméticos

Operadores para acumulação

- Destaque para os novos operadores!
 - Exclusivos para ACUMULACAO!

Operação	Tradicionais	Exemplo	Novos	Exemplo
Soma	+	<code>a=a+X ;</code>	<code>+=</code>	<code>a+=X ;</code>
Subtração	-	<code>b=b-Y ;</code>	<code>-=</code>	<code>b-=Y ;</code>
Multiplicação	*	<code>c=c*4 ;</code>	<code>*=</code>	<code>c*=4 ;</code>
Divisão	/	<code>d=d/3 ;</code>	<code>/=</code>	<code>d/=3 ;</code>
Resto de divisão	%	<code>e=e%2 ;</code>	<code>%=</code>	<code>e%=2 ;</code>

Operadores Aritméticos

Operadores para acumulação

- Exemplo 1

```
ac = 7  
ac = 3  
ac = 1
```

```
#include <stdio.h>  
int main () {  
  
    int ac=1;  
  
    ac*=7;  
    printf("ac = %i\n", ac);  
    ac/=2;  
    printf("ac = %i\n", ac);  
    ac%=2;  
    printf("ac = %i\n", ac);  
  
    return 0;  
}
```

Operadores Aritméticos

Operadores para acumulação

- Destaque para os novos operadores!
 - Exclusivos para ACUMULACAO!

Operação	Pós-fixado	Pré-fixado	Equivalências	
Incrementa	<code>i++;</code>	<code>++i;</code>	<code>i=i+1;</code>	<code>i+=1;</code>
Decrementa	<code>i--;</code>	<code>--i;</code>	<code>i=i-1;</code>	<code>i-=1;</code>

Operadores Aritméticos

Operadores para acumulação

- Exemplo 2

```
cont=1  
cont=2  
cont=3
```

```
#include <stdio.h>
```

```
int main () {
```

```
    int cont=0;
```

```
    cont=cont+1;
```

```
    printf("cont= %i\n", cont);
```

```
    cont+=1;
```

```
    printf("cont= %i\n", cont);
```

```
    cont++;
```

```
    printf("cont= %i\n", cont);
```

```
    return 0;
```

```
}
```

Operadores Aritméticos

Operadores para acumulação

- Exemplo 3

```
i = 9  
i = 9  
i = 8  
i = 7
```

```
#include <stdio.h>  
int main () {  
  
    int i=10;  
  
    i--;  
    printf("i= %i\n", i);  
    printf("i= %i\n", i--);  
    printf("i= %i\n", i);  
    printf("i= %i\n", --i);  
  
    return 0;  
}
```

Operadores Aritméticos

Precedência de operadores

- Quanto eh dois mais dois dividido por dois?

alta
↓
baixa

Prioridades de operadores		
()		
++		--
*	/	%
+		-

```
2 + 2 / 2 = 3
(2 + 2) / 2 = 2
```


Operadores Lógicos

Operadores Lógicos

Entendendo V ou F

- Linguagem C não tem o tipo booleano!
 - FALSO -> 0
 - VERDADEIRO -> qualquer coisa diferente de 0
 - Operadores booleanos retornam 1 para VERDADEIRO

Operação	Operador
Eh igual?	==
Eh diferente?	!=
Eh maior/menor que?	> <
Eh maior/menor OU igual que?	>= <=

Operadores Lógicos

Entendendo V ou F

- Exemplo

```
V ou F? 1
V ou F? 0
V ou F? 0
V ou F? 0
```

```
#include <stdio.h>
int main () {
    int a=2, b=3, c;

    c=0!=1;
    printf("V ou F? %i\n", c);
    c=a>2;
    printf("V ou F? %i\n", c);

    printf("V ou F? %i\n", 1>=2);
    printf("V ou F? %i\n", a==b);

    return 0;
}
```

Operadores Lógicos

Entendendo V ou F

- Linguagem C não tem o tipo booleano!
 - FALSO -> 0
 - VERDADEIRO -> qualquer coisa diferente de 0
 - Operadores booleanos retornam 1 para VERDADEIRO

Operação	Operador
Negação	!
E lógico	&&
OU lógico	

```
#include <stdio.h>
```

```
int main () {
```

```
    int c=5;
```

```
    c = !c;
```

```
    printf("V ou F? %i\n", c);
```

```
    printf("V ou F? %i\n", -1 && c);
```

```
    printf("V ou F? %i\n", c || -4);
```

```
    return 0;
```

```
}
```

Operadores Lógicos

Entendendo V ou F

- NEGACAO - tabela verdade

Entrada	Saida
V	F
F	V

```
#include <stdio.h>
```

```
int main () {
```

```
    int c=5;
```

```
    c = !c;
```

```
    printf("V ou F? %i\n", c);
```

```
    printf("V ou F? %i\n", -1 && c);
```

```
    printf("V ou F? %i\n", c || -4);
```

```
    return 0;
```

```
}
```

Operadores Lógicos

Entendendo V ou F

- E - tabela verdade

Entradas		Saida
V	V	V
V	F	F
F	V	F
F	F	F

```
#include <stdio.h>
```

```
int main () {
```

```
    int c=5;
```

```
    c = !c;
```

```
    printf("V ou F? %i\n", c);
```

```
    printf("V ou F? %i\n", -1 && c);
```

```
    printf("V ou F? %i\n", c || -4);
```

```
    return 0;
```

```
}
```

Operadores Lógicos

Entendendo V ou F

- OU - tabela verdade

Entradas		Saida
V	V	V
V	F	V
F	V	V
F	F	F

```
#include <stdio.h>
```

```
int main () {
```

```
    int c=5;
```

```
    c = !c;
```

```
    printf("V ou F? %i\n", c);
```

```
    printf("V ou F? %i\n", -1 && c);
```

```
    printf("V ou F? %i\n", c || -4);
```

```
    return 0;
```

```
}
```

Operadores Lógicos

Entendendo V ou F

```
V ou F? 0
V ou F? 0
V ou F? 1
```

```
#include <stdio.h>
```

```
int main () {
```

```
    int c=5;
```

```
    c = !c;
```

```
    printf("V ou F? %i\n", c);
```

```
    printf("V ou F? %i\n", -1 && c);
```

```
    printf("V ou F? %i\n", c || -4);
```


```
    return 0;
```

```
}
```


Operadores Lógicos

Precedência de Operadores

alta



baixa

Prioridades de operadores			
()			
!	++	--	
*	/	%	
+		-	
<	<=	>=	>
!=		==	
&&			

MUITO
OBRIGADO

Prof. André del Mestre

www.ifsul.edu.br
almmartins@charqueadas.ifsul.edu.br