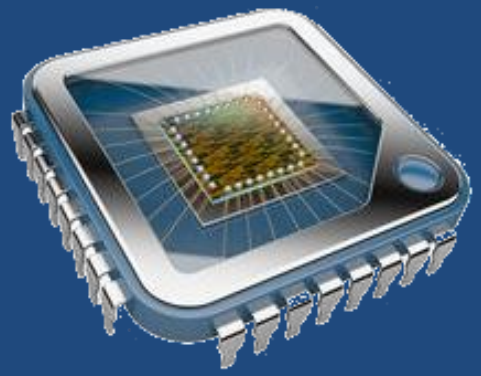
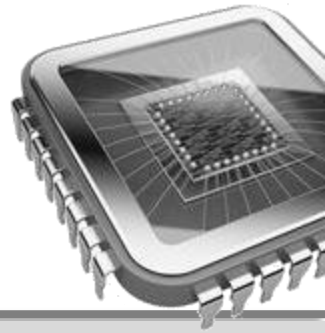


Introdução à Programação
Aula 06

Prof. Max Santana Rolemberg Farias
max.santana@univasf.edu.br
Colegiado de Engenharia de Computação

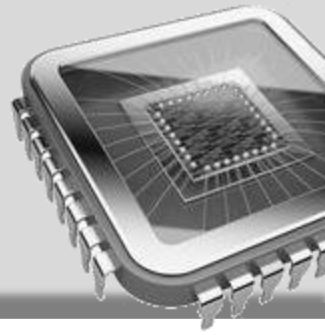
GESE
Grupo de Engenharia de Sistemas Embarcados

CEComp
Engenharia da Computação
UNIVASF



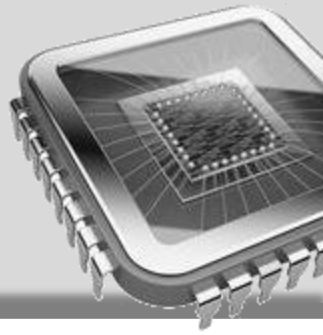
ESCOPO DE VARIÁVEIS

Escopo de variáveis



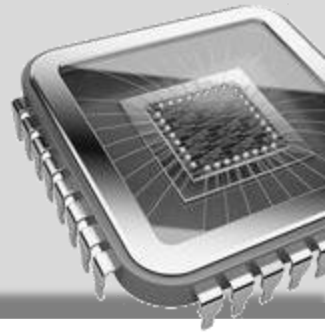
- O escopo de uma variável é o trecho do programa que ela pode ser acessada.
  - Na linguagem C, uma variável pode ser declarada no interior de um bloco, na lista de parâmetros ou fora de um bloco;
- Toda variável declarada no interior de um bloco ou na lista de parâmetros de uma função, possui escopo de bloco
- Toda variável declarada fora de um bloco ou lista de parâmetros de uma função, possui escopo de arquivo.
- **As variáveis de escopo diferentes podem ter o mesmo nome.**

Escopo de variáveis (continuação)



- As variáveis podem ser classificadas, de acordo com o seu escopo
  - Globais
  - Locais
- As variáveis com escopo de arquivo são classificadas como globais
- As variáveis com escopo de bloco são classificadas como locais

Escopo de variáveis (continuação)



arquivo.c

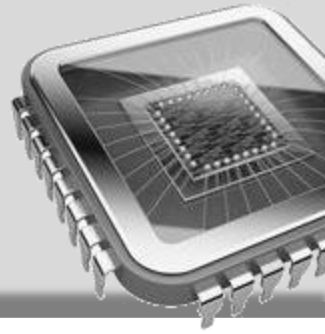
```
#include <stdio.h>
int valor;
int main(void){
  int taxa = 20;
  valor = 2000;
  printf("Valor total: %d", valor*taxa);
  return 0;
}

void imprimir(double salario, int idade){
  printf("%f %d \n", salario, idade);
}
```

A variável valor possui escopo de arquivo, pois foi declarada fora de qualquer bloco ou lista de parâmetro.

A variável taxa possui escopo de bloco, pois foi declarada dentro do bloco que delimita a função `main`.

As variáveis salario e idade possuem escopo de bloco, pois foram declaradas na lista de parâmetros da função imprimir.

Escopo de variáveis (continuação)



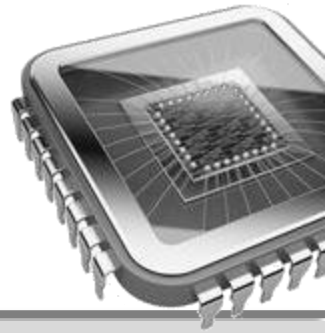
arquivo.c

```
#include <stdio.h>
int valor;
int main(void){
   int taxa = 20;
   valor = 2000;
   printf("Valor total: %d", valor*taxa);
   return 0;
}

void imprimir(double salario, int idade){
   printf("%f %d \n", salario, idade);
}
```

A variável valor declarada no início do programa (arquivo.c) têm escopo de arquivo e é uma variável global.

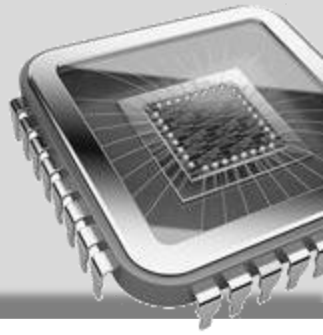
A variável taxa declarada no corpo da função `main` é uma variável local do bloco.

As variáveis salario e idade também são variáveis locais da função imprimir.



COMANDO DE COMPILAÇÃO

COMPILAÇÃO

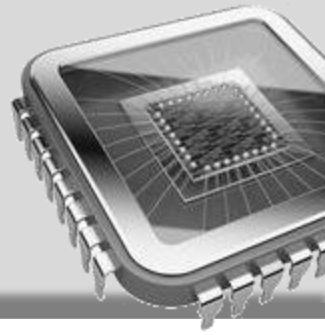


- O comando mais simples do compilador **gcc**, realiza todas as etapas da compilação bastando somente relacionar o código-fonte.
  - Esse comando gera um código executável como o nome **a.out**.

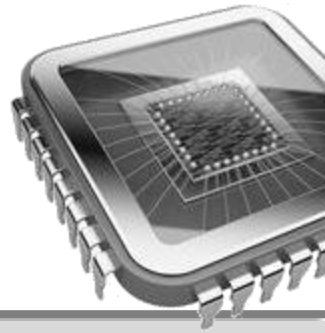
gcc <nome_programa>.c

Compilando o programa com a opção de compilação – o (**gcc <nome_programa>.c – o <nome_executável>**) faz o código executável ser armazenado em um arquivo executável especifico.

Compilação (continuação)

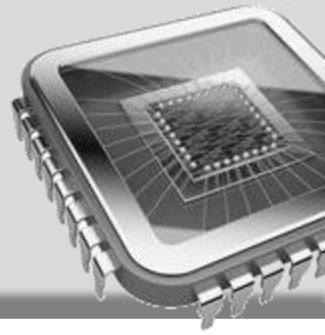


Comandos	Descrição
gcc prog.c	Compila o programa **prog.c** e gera um executável **a.out**
gcc prog.c aux.c io.c	Compila o programa **prog.c** e os códigos distribuídos **aux.c** e **io.c** e gera um executável **a.out**
gcc prog.c –o exe	Compila o programa **prog.c** e gera um executável chamado **exe**
gcc prog.c aux.c io.co –o exe	Compila o programa **prog.c** e os códigos distribuídos **aux.c** e **io.c** e gera um executável chamado **exe**



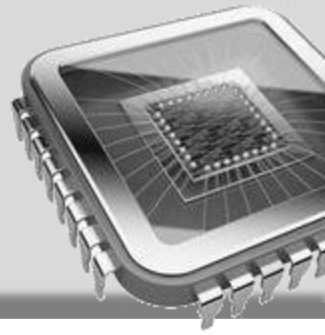
OPERADORES E EXPRESSÕES

Operadores



- Na linguagem C os operadores podem ser classificados quanto ao número de operandos:
  - Unário: Requerem um operando
  - Binários: Requerem dois operandos
  - Ternários: Requerem três operandos
- A notação para representar uma operação pode ser:
  - Prefixada: Se o operador ocorre antes dos operandos.
  - Pós-fixada: Se o operando ocorre depois dos operandos.
  - Infixada: Se o operador ocorre entre os operandos.

Operadores Aritméticos

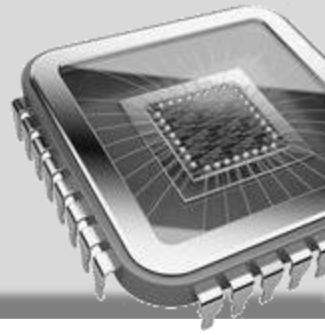


- Os operadores aritméticos, podem ser aplicados a quase todos os operandos do tipo aritmético.

Operação Binária	Operador	Exemplo
Adição	+	A+B
Subtração	-	A–B
Multiplicação	*	A*B
Divisão	/	A/B
Resto (mod)	%	A%B
Incremento	++	
Decremento	--	

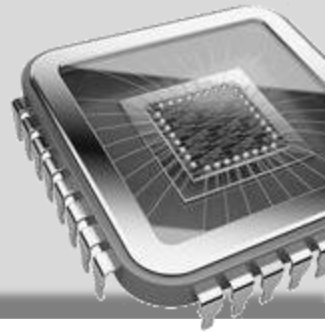
Operação Unárias	Operador	Exemplo
Mais	+	A+B
Menos	-	A–B
Incremento prefixada	++	++A
Incremento pós-fixada	++	A++
decremento prefixada	--	--B
decremento pós-fixada	--	B--

Operadores Atribuições



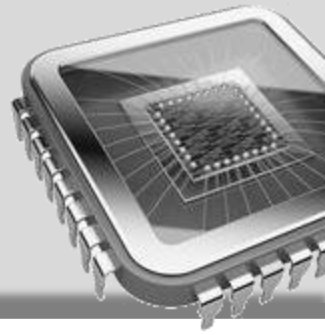
Operador	Exemplo	Equivalente
=	A = B	-
+=	A += B	A = A+B
-=	A -= B	A = A–B
*=	A *= B	A = A*B
/=	A /= B	A = A/B
%=	A %= B	A = A%B
<<=	A <<= B	A = A<<B
>>=	A >>= B	A = A>>B
&=	A &= B	A = A&B
^=	A ^= B	A = A^B
|=	A |= B	A = A | B

Operadores Relacionais



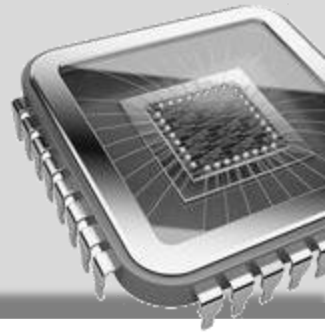
Operação Relacional	Operador	Exemplo	Resultado
Menor	<	A < B	1, se o valor de A for menor que o valor de B; 0 , em caso contrário
Menor ou igual	<=	A <= B	1, se o valor de A for menor ou igual ao valor de B; 0 , em caso contrário
Maior	>	A > B	1, se o valor de A for maior que o valor de B; 0 , em caso contrário
Maior ou igual	>=	A >= B	1, se o valor de A for maior ou igual ao valor de B; 0 , em caso contrário
Igual	==	A == B	1, se o valor de A for igual ao valor de B; 0 , em caso contrário
Diferente	!=	A != B	1, se o valor de A for igual ao valor de B; 0 , em caso contrário

Operadores Lógicos



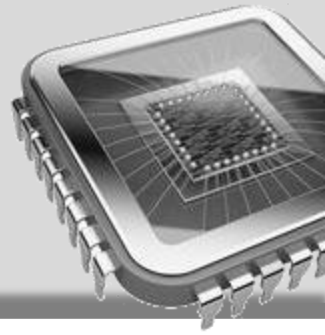
Operação Lógica	Operador	Exemplo	Resultado
Conjunção (E)	&&	A && B	1, se o valor de A e B forem ambos verdadeiros; 0 , em caso contrários.
Disjunção (OU)	||	A || B	1, se o valor de A ou o valor de B for verdadeiro; 0 , em caso contrário
Complemento lógico (negação)	!	!A	1, se o valor de A for falso; 0, se for verdadeiro

Operadores
Hierarquia



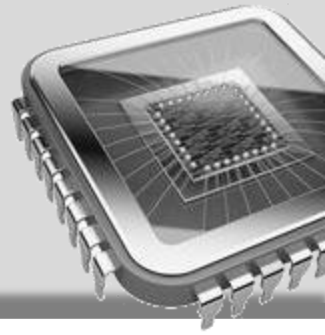
Hierarquia	Operadores relacionais e lógicos
1ª	!
2º	>, >=, <, <=
3º	==, !=
4º	&&
5ª	||

Operadores Lógicos Binário



Operação Lógica	Operador	Exemplo	Resultado
Conjunção (E)	&	A & B	Cada bit do resultado é igual a 1, se os bits correspondentes dos operadores forem iguais a 1; ou é igual a 0, em caso contrário
Disjunção (OU)	|	A | B	Cada bit do resultado é igual a 0, se os bits correspondentes dos operandos forem iguais a 0; ou é igual a 1, em caso contrário
Disjunção exclusiva (XOR)	^	A ^ B	Cada bit do resultado é igual a 1, se os bits correspondentes dos operandos forem diferentes entre si; ou é igual a 0, em caso contrário
Negação	~	~A	Cada bit do resultado é igual a 1, se o bit correspondente do operando for igual a 0; ou é igual a 1, em caso contrário

Operadores
Deslocamento Binário



Operação Lógica	Operador	Exemplo	Resultado
Deslocamento à esquerda	<<	A << B	Desloca de B unidades para a esquerda os bits que compõem a representação de A
Deslocamento à direita	>>	A >> B	Desloca de B unidades para a direita os bits que compõem a representação de A