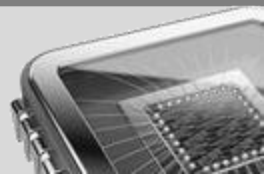
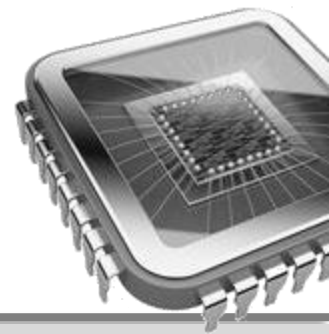


# Introdução à Programação

## Aula 07

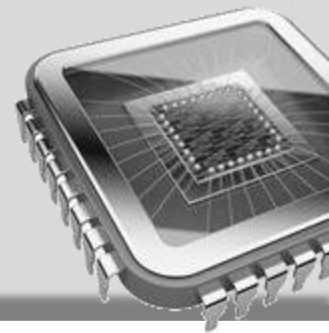
Prof. Max Santana Rolemberg Farias  
max.santana@univasf.edu.br  
Colegiado de Engenharia de Computação





# ARGUMENTOS POR LINHA DE COMANDO

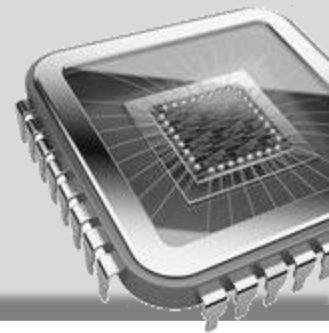
# Argumento por linha de comando



- Na linguagem C podemos passar argumentos através da linha de comando para um programa quando ele inicia.
  - Para isso a função *main* deve receber parâmetros específicos

```
int main (int argc, char *argv[])
```

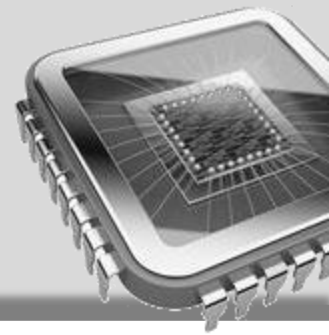
# Argumento por linha de comando



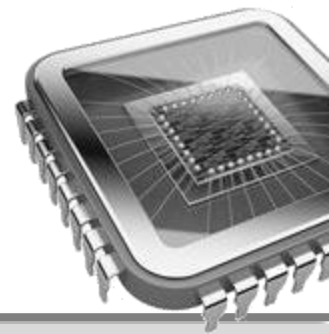
```
int main (int argc, char *argv[])
```

- **argc**
  - É um valor inteiro que indica a quantidade de argumentos que foram passados ao chamar o programa.
- **argv**
  - É um vetor de char que contém os argumentos, um para cada *string* passada na linha de comando.
  - **argv[0]** armazena o nome do programa que foi chamado no prompt, sendo assim, argc é pelo menos igual a 1, pois no mínimo existirá um argumento.
- Os argumentos passados por linha de comando devem ser separados por um espaço ou tabulação.

# Argumento por linha de comando: Exemplos



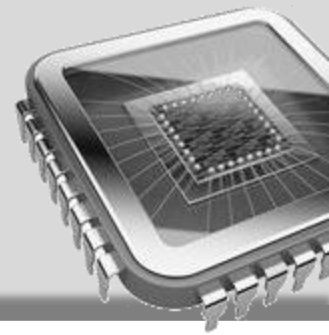
```
# include < stdio.h>
int main (int argc, char *argv[]) {
    int resultado, valorA = 0, valorB = 0;
    print (“\n Multiplicando valores passados na linha de comando \n”);
    //atoi converte de alfanumérico para inteiro
    valorA = atoi(argv[1]);
    valorB = atoi(argv[2]);
    resultado = valorA * valorB;
    printf (“\n %d x %d = %d \n”, valorA, ValorB, resultado) ;
    return 0;
}
```



# ESTRUTURA DE CONTROLE DE FLUXO

# Estrutura de Controle de Fluxo

## Instrução Condicional *if*

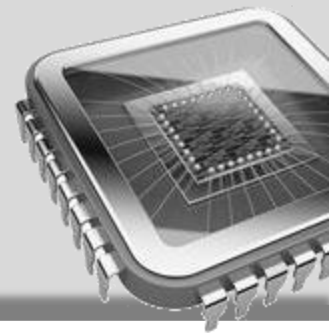


```
if (<expressão_lógica>) {  
    <instrução>;  
}
```

```
if (<expressão_lógica>)  
    <instrução>;
```

# Estrutura de Controle de Fluxo

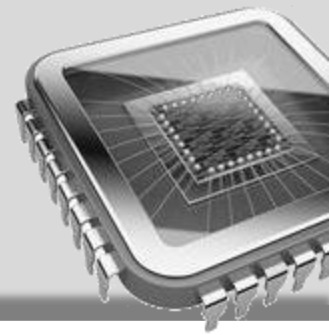
## Instrução Condicional *if*



```
if (<expressão_lógica>) {  
    <instrução_1>;  
    <instrução_2>;  
    ...  
    <instrução_n>;  
}
```

# Estrutura de Controle de Fluxo

## Instrução Condicional *if/else*

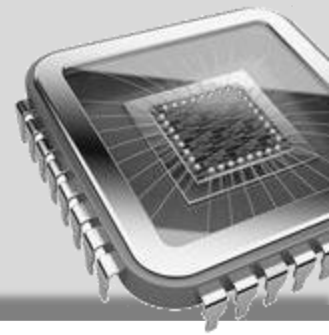


```
if (<expressão_lógica>) {  
    <instrução_if>;  
} else {  
    <instrução_else>;  
}
```

```
if (<expressão_lógica>)  
    <instrução_if>;  
else  
    <instrução_else>;
```

# Estrutura de Controle de Fluxo

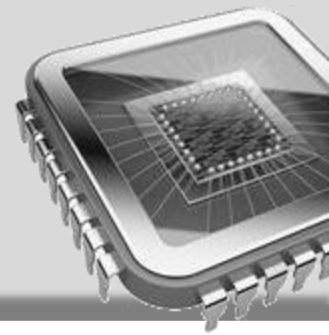
## Instrução Condicional *if/else*



```
if (<expressão_lógica>) {  
    <instrução_if_1>;  
    ...  
    <instrução_if_n>;  
} else {  
    <instrução_else_1>;  
    ...  
    <instrução_else_n>;  
}
```

# Estrutura de Controle de Fluxo

## Seleção Múltipla *switch/case*



```
switch (<variável>
  case <constante1>:
    <instrução_1>;
    ...
    <instrução_n>;
    break;
  ...
  case <constante_n>:
    <instrução_1>;
    break;
  default:
    <instrução_1>;
    ...
    <instrução_n>;
}
```



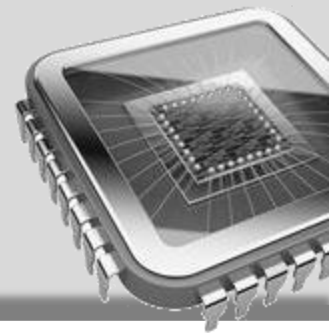
A **variável** do comando `switch/case` deve ser do tipo inteiro (`int` ou `char`)



O comando **break** causa a interrupção do `switch/case`.

# Estrutura de Controle de Fluxo

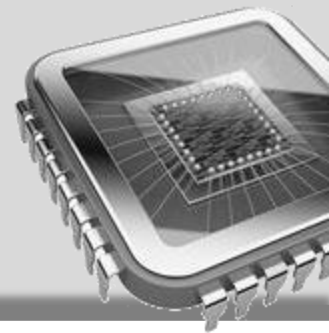
## Instrução de Repetição *while*



```
while (<expressão_lógica>) {  
    <instrução_1>;  
    <instrução_2>;  
    ...  
    <instrução_n>;  
}
```

# Estrutura de Controle de Fluxo

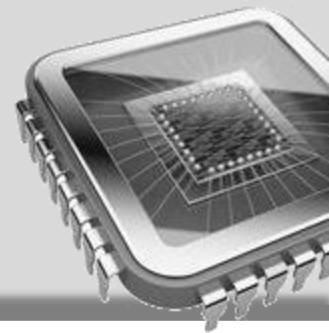
## Instrução de Repetição *do/while*



```
do {  
    <instrução_1>;  
    <instrução_2>;  
    ...  
    <instrução_n>;  
} while (<expressão_lógica>;
```

# Estrutura de Controle de Fluxo

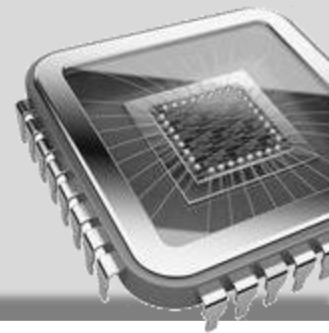
## Instrução de Repetição *for*



```
for (<ini_variável>; <exp_lógica>; <inc_variável>) {  
    <instrução_1>;  
    ...  
    <instrução_n>;  
}  
for (<instrução>; <exp_lógica>; <instrução>) {  
    <instrução_1>;  
    ...  
    <instrução_n>;  
}
```

# Estrutura de Controle de Fluxo

## Comando *continue*



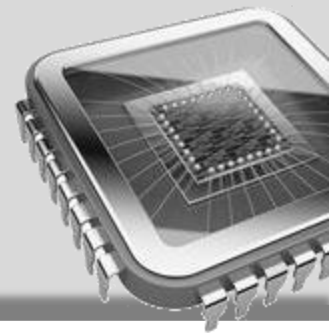
- O comando `continue` pode ser usado apenas no corpo dos comandos:
  - `while`
  - `do/while`
  - `for`
- Ele interrompe a repetição corrente sem, entretanto, interromper a execução do comando.



O comando `break` também pode ser usado no interior de comando `while`, `do/while` e `for`. E quando executado interrompe a execução do comando.

# Estrutura de Controle de Fluxo

## Comando *break* e *continue*



### Break

```
/* A maior soma dos n primeiros
números menor ou igual ao limite */
soma = 0;
p = 0;
n = 5;
lim = 60
while (p <= n) {
    if (soma + p > lim) {
        break;
    }
    soma = soma + p++;
}
```

### Continue

```
/* lê 3 números pares e maiores do
que 0 e somar todo os números */
qtd = 0;
while (qtd < 3) {
    scanf("%d", &num);
    if ((num % 2) != 0 || (num <= 0))
        continue;
    qtd++; soma = 0;
    for (i = 1; i <= num; i++)
        soma = soma + i;
}
```