

Universidade de Brasília

Departamento de Ciência da Computação

Curso C: Controle de Fluxo

Prof. Ricardo Pezzuol Jacobi
rjacobi@cie.unb.br

Linguagem C

Ricardo Jacobi

O comando *if*

```
if ( expressão é verdadeira )  
    execute comando ou bloco de comandos ;  
else /* se expressão é falsa */  
    execute comando ou bloco de comandos ;
```

Ex:

```
if ( count > 9 )  
    count = 0;  
else  
    count++;
```

Linguagem C

Ricardo Jacobi

Aninhamento de *if*

é possível aninhar construções do tipo *if-else* em diversos níveis

```
if (cond1)          /* if1 */
{
  if (cond2)        /* if2 */
  {
    comando if2;
  }
  else              /* else2 */
  {
    comando else2;
  }
}
else                /* else1 */
{
  if (cond3)        /* if3 */
  {
    if (cond4)      /* if4 */
    {
      comando if4;
    }
    else            /* else4 */
    {
      comando else4;
    }
  }
  else              /* else3 */
  {
    comando else3;
  }
}
```

Linguagem C

Ricardo Jacobi

Aninhamento de *if*

cuidado com a correspondência entre *if*'s e *else*'s:

```
if (cond1)
{
  if (cond2)
  {
    comando if2;
  }
  else
  {
    comando if1;          /* atenção: else2! */
                          /* erro: comando do if2 */
  }
}
```

modo correto:

```
if (cond1) {
  if (cond2)
  {
    comando if2;
  }
}
else
{
  comando if1;
}
```

Linguagem C

Ricardo Jacobi

Exemplo

- Converter um *string* tipo "10" para um valor inteiro

```
...
char str[] = "10";
if (str[0]!='0')
    if (str[1]!='0') printf("Zero");
    else printf("Um");
else /* str[0] == '1' */
    if (str[1]!='0') printf("Dois");
    else printf("Tres");
```

Linguagem C

Ricardo Jacobi

Encadeamento *if-else-if*

```
if (teste_1) <comando_1>;
else if (teste_2) <comando_2>;
else if (teste_3) <comando_3>;
```

...

```
else <comando_n>;
```

- No encadeamento apenas *um* dos *n* comandos será executado: o primeiro cujo teste for verdadeiro

Linguagem C

Ricardo Jacobi

Encadeamento *if-else-if*

- Ex: escrever o nome de um dígito
'0' -> "zero", '1' -> "um", etc.

...

```
if (ch == '0') printf("Zero");  
else if (ch=='1') printf("Um");  
else if (ch=='2') printf("Dois");  
else if ...  
else if (ch=='0') printf("Nove");  
else printf("Nao era um digito!");
```

Linguagem C

Ricardo Jacobi

Expressão Condicional ?

- A expressão condicional "? : ." é uma simplificação do if-else utilizada tipicamente para atribuições condicionais:

```
exp1?exp2:exp3 ≈ if (exp1)?exp2; else exp3;
```

- Ex: implementando $z = \max(x, y)$ com:

```
if: if (x > y) z=x; else z=y;
```

```
?: z = (x > y) ? x : y;
```

Linguagem C

Ricardo Jacobi

Exemplo

- O que faz o trecho de código abaixo ?

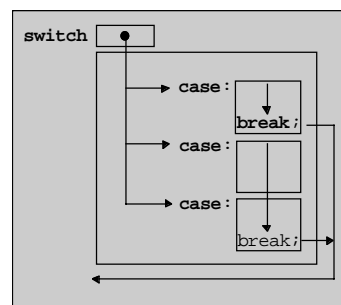
```
...
conta = 0;
for (index=0; index < 1000; index++)
{
    printf("%d", conta);
    conta = (conta==8) ? 0 : conta+1;
}
```

Linguagem C

Ricardo Jacobi

O comando *switch*

```
switch ( valor ) {
    case valor1:
        comandos1;
        break;
    case valor2:
        comandos2;
        break;
    default:
        comandos default;
        break;
}
```



Linguagem C

Ricardo Jacobi

O comando *switch*

- a expressão *valor* é avaliada e o valor obtido é comparado com os valores associados às cláusulas *case* em sequência.
- quando o valor associado a uma cláusula é igual ao valor do *switch* os respectivos comandos são executados até encontrar um *break*.
- se não existir um *break* na cláusula selecionada, os comandos das cláusulas seguintes são executados em ordem até encontrar um *break* ou esgotarem-se as cláusulas do *switch*
- se nenhuma das cláusulas contém o valor de seleção, a cláusula *default*, se existir, é executada

Linguagem C

Ricardo Jacobi

Exemplo *switch*

```
switch( char_in ) {  
  case '.': printf( "Ponto.\n" );  
            break;  
  case ',': printf( "Virgula.\n" );  
            break;  
  case ' ': printf( "Dois pontos.\n" );  
            break;  
  case ';': printf( "Ponto e virgula.\n" );  
            break;  
  default : printf( "Nao eh pontuacao.\n" );  
}
```

Linguagem C

Ricardo Jacobi

Exemplo *switch*

```
switch( char_in ) {
  case '0': putchar('0');      /* 0123456789 */
  case '1': putchar('1');      /* 123456789 */
  case '2': putchar('2');      /* 23456789 */
  case '3': putchar('3');      /* 3456789 */
  case '4': putchar('4');      /* 456789 */
  case '5': putchar('5');      /* 56789 */
  case '6': putchar('6');      /* 6789 */
  case '7': putchar('7');      /* 789 */
  case '8': putchar('8');      /* 89 */
  case '9': putchar('9');      /* 9 */
          break; }|
```

Linguagem C

Ricardo Jacobi

Exercício *switch*

- Contar o número de ocorrências de dígitos decimais em uma sequência de caracteres digitados pelo usuário utilizando o comando `switch`

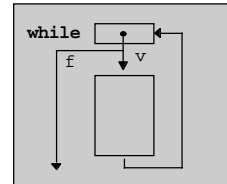
```
main() {
  char ch;  int ch_count = 0;
  printf("- Entre caracteres ('F' para terminar) -\n" );
  do {     ch = getchar();
          /*
           *  usar switch para contar os digitos
           */
          } while (ch != 'F');
  printf ("\n\Lidos: %d\n", ch_count);
}
```

Linguagem C

Ricardo Jacobi

Comando *while*

```
while (condição)
{
    comandos;
}
```



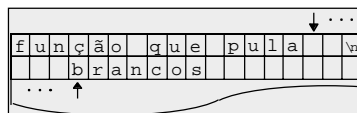
- 1º avalia condição
- se condição é verdadeira, executa comandos do bloco
- ao término do bloco, volta a avaliar condição
- repete o processo até que condição seja falsa

Linguagem C

Ricardo Jacobi

Exemplo *while*

```
void pula_branços () {
int ch;
while ((ch = getchar()) == ' ' || /* brancos */
ch == '\n' || /* newline */
ch == '\t' ) /* tabs */
; /* não faz nada */
}
```



Linguagem C

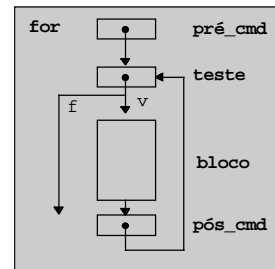
Ricardo Jacobi

Comando *for*

```
for (pré_cmd; teste; pós_cmd) {  
  comandos;  
}
```

• em termos de *while*, equivale a:

```
pré_cmd;  
while (teste) {  
  comandos;  
  pós_cmd;  
}
```



Comando *for*

- 1º executa *pré_cmd*, que permite iniciar variáveis
- 2º avalia teste: se verdadeiro, executa comandos do bloco, senão encerra laço
- ao término do bloco, executa *pós_cmd*
- reavalia teste
- repete o processo até que teste seja falso

Exemplo *for*

- Ex: imprimir o conteúdo de um array:

```
void main() {
    int i;
    for ( i=0; i < size_array; i++)
        printf ("%d ", array[i]);
}
```

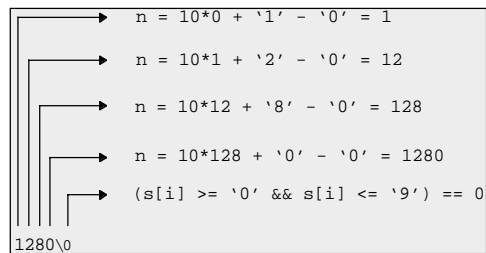
Linguagem C

Ricardo Jacobi

Exemplo *for*

- Ex: conversão de string para inteiro

```
int atoi (char s[]) { /* asc to integer */
    int i, n;
    for (n=0, i=0; s[i] >= '0' && s[i] <= '9'; ++i)
        n = 10 * n + s[i] - '0';
    return (n);
}
```



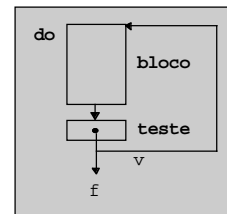
Linguagem C

Ricardo Jacobi

Comando *do-while*

- `do-while` é utilizado sempre que o bloco de comandos deve ser executado ao menos uma vez

```
do  
{  
  comandos;  
}  
while (condição);
```



Comando *do-while*

- 1º executa comandos
- 2º avalia condição:
 - se verdadeiro, reexecuta comandos do bloco
 - senão encerra laço

Exemplo *do-while*

```
void itoa (int num, char s[]) {
int i = 0; int sinal;
  if ((sinal = num) < 0) /* armazena sinal */
    num = - num; /* faz num positivo */
  do {
    s[i] = num % 10 + '0'; /* unidades */
    ++i;
  } while ((num /= 10) > 0); /* quociente */
  if (sinal < 0) s[i] = '-';
  ++i;
  s[i] = '\\0';
  reverse(s);
}
```

Linguagem C

Ricardo Jacobi

Comando *break*

- o comando *break* permite interromper a execução de um laço ou de um *switch*

- Ex:

```
main () {
int i, j;
for (i = 0; i < 4; i++)
  for (j = 0; j < 2; j++)
    if (i == 1) break;
    else printf("i: %d j: %d\\n", i, j);
}
```

```
i: 0 j: 0
i: 0 j: 1
i: 2 j: 0
i: 2 j: 1
i: 3 j: 0
i: 3 j: 1
```

Linguagem C

Ricardo Jacobi

Comando *continue*

- o comando *continue* leva a execução do próximo passo de uma iteração. Os comandos que sucedem *continue* no bloco não são executados

- Ex:

```
main() {  
  int i;  
  for (i = 0; i < 5; i++)  
    if (i == 1) continue;  
    else printf("i: %d \n", i);  
}
```

```
i: 0  
i: 2  
i: 3  
i: 4
```

Goto's e labels

- C suporta os comandos *goto*, que permitem o desvio do fluxo de execução do programa para uma posição indicada por um rótulo (*label*)
- apesar de banido da prática de programação estrutura, *goto's* podem ser úteis em determinadas circunstâncias, como sair de dentro de laços aninhados

Exemplo *goto*

```
for ( ... )
for ( ... ) {
...
if ( desastre )
goto erro;
}
...
/* o label deve estar na mesm função */
erro:
    dah_um_jeitinho();
```

Linguagem C

Ricardo Jacobi

Exemplo com *goto*

```
/* usando goto */
for ( i=0; i < n; i++ )
    for ( j=0; j < m; j++ )
        if ( A[i] == B[j] )
            goto achei;
/* trata outro caso: não achou */
achei:
/* tratamento do achado */
```

Linguagem C

Ricardo Jacobi

Exemplo sem *goto*

```
/* sem goto */
int achei = 0;
for ( i=0; i < n; i++ )
    for ( j=0; j < m; j++ )
        if ( A[i] == B[j] )
            achei = 1;
if (achei) /* tratamento do achado */
else /* trata outro caso: não achou */
```

Linguagem C

Ricardo Jacobi

Exercícios

1. Usando o comando *for*, faça um algoritmo que conte o número de 1's que aparecem em um *string*
ex: 0011001 => 3
2. Usando o comando *while*, escreva um programa que substitui as ocorrências de um caracter *ch0* em um *string* por um outro caracter *ch1*
3. Utilizando o comando *do-while*, implemente um programa que converte um *string* contendo um número binário positivo em um inteiro.
ex: "001101" => 13

Linguagem C

Ricardo Jacobi