FATEC - SP

Faculdade de Tecnologia de São Paulo

Prof° Luiz Paulo Zanetti

E-mail: luizpaulozanetti@hotmail.com

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina
Linguagem de Programação

Switch = interruptor

Switch

Embora o if-else-if possa executar vários tipos de testes, o código pode ficar muito difícil de ser seguido. C possui um comando de vários desvios chamado switch.

No switch, o computador testa uma variável sucessivamente contra uma lista de constantes inteiras ou de caracteres e executa um comando ou bloco de comandos quando encontrar uma coincidência.

Switch = interruptor

Forma geral do switch

```
switch (variável)
  {
   case constante1:
   seqüência de comandos
   break;
   case constante2:
   seqüência de comandos
   break;
   default:
   seqüência de comandos
}
```

Switch = interruptor

O comando default dentro do switch

O comando default será executado se não for encontrada nenhuma coincidência na lista de constantes.

Caso não seja colocado um comando default e não haja coincidência, nenhum comando será executado.

Switch = interruptor

O comando break

Quando o computador encontra alguma coincidência, ele executa os comandos associados àquele case até encontrar break ou o fim do comando switch.

É um erro muito comum dos programadores esquecerem de colocar o break após os comandos.

Switch = interruptor

O switch difere do if, já que o primeiro só pode testar igualdade e a expressão condicional if pode ser de qualquer tipo.

Não pode haver duas constantes case com valores iguais no mesmo switch.

Podem ser colocados comandos switch dentro de comandos switch.

Pode ser deixado um case vazio quando mais de uma condição usa o mesmo código.

Switch = interruptor

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
 printf ("Digite um valor de 1 a 3: ");
scanf("%d",&a);
switch(a)
{
      case 1:
      printf( “OK – 01”);
      break;
      case 2:
      printf( “OK – 02”);
      break;
      case 3:
      printf( “OK – 03”);
      break;
      default :
      printf ("Valor invalido!\n");
}
getch();
}
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a;
 printf ("Digite uma letra de A a C: ");
scanf("%s",&a);
switch(a)
{
      case ‘A’:
      printf( “OK – A”);
      break;
      case ‘B’:
      printf( “OK – B”);
      break;
      case ‘C’:
      printf( “OK – C”);
      break;
      default :
      printf ("Valor invalido!\n");
}
getch();
}
```

While = Enquanto

While

A execução do while executa um comando (ou bloco de comandos) enquanto uma condição for verdadeira.

A forma geral do while é:

```
while (condição)
  comando;
```

While = Enquanto

Exemplo - 01

```
pausa()
{
    char tecla = '\0';
    printf("Tecle ESPAÇO para continuar...");
    while (tecla != ' ')
    tecla = getche();
}
```

While = Enquanto

Exemplo - 02

O exemplo anterior pode ser reescrito sem utilizar variável.

Veja:

```
main()
{
   printf("Tecle ESPAÇO para continuar...");
   while (getche() != ' ');
}
```

Do While = Repita até

Ao contrário do loop for e do loop while, que testam a condição no começo, o loop do while verifica somente no final. Desta forma, o loop será executado pelo menos uma vez.

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

Do While = Repita até

Exemplo - 02

```
main() /* Lê um número maior que 100 */
{
   int num;
   do
   {
    printf("Digite um número maior que 100");  scanf("%d",
    &num);
   }
   while (num <= 100);
}
```

While e Do While

Loops aninhados

Quando um loop está dentro do outro, dizemos que o loop interior está aninhado.

Loops aninhados permite que sejam solucionados vários problemas de programação.

Em C podemos aninhar qualquer tipo de loop.

While e Do While

Estrutura de Repetição
Comparativo Enquanto e Repita
Os dois laços podem ser executados “N” vezes ou uma quantidade previamente determinada na programação.

Laço **DO WHILE**

O laço **DO WHILE** é executado obrigatoriamente uma vez ! ;

A avaliação do laço **DO WHILE** é feita no final do laço;

Uma nova execução ocorreno no laço **DO WHILE** quando a avaliação da condição for falsa ;

Laço **WHILE**

O laço **WHILE** pode não ser executado;

A avaliação do laço **WHILE** é realizada no início do laço;

Uma nova execução ocorre no laço **WHILE** quando a avaliação da condição for verdadeira.

while

Verifica para executar

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
lnt a=0
clrscr();
      while(a<=10)
      {
      printf(“FATEC");
      delay(1000);
      a++;
      }
getch();
}
```

do while

Executa para verificar

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
lnt a=0
clrscr();
      do
      {
      printf(“FATEC");
      delay(1000);
      a++;
      }
       while(a<=10);
getch();
}
```