



**USP - ICMC - SSC
SSC 0501 - 1o. Semestre 2010**

**Disciplina de
Introdução à Ciência da Computação
ICC 1 - Teoria**

Prof. Fernando Santos Osório

Email: fosorio [at] { icmc. usp. br , gmail. com }

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Web - WIKI ICMC: <http://wiki.icmc.usp.br/index.php/SSC-501>

PAE: Gustavo Pessin (Dout. CCMC) / <Http://pessin.googlepages.com/>

Email: pessin [at] { icmc.usp.br , gmail.com }

Monitor: Matheus Lin Alvarenga (EC) / <Http://matheuslin.wordpress.com/>

Email: matheus.lin [at] gmail.com

Aula 05t

1

Abril 2010

Linguagem de Programação “C”

Agenda:

- Programa com Laço: além do FOR em “C”
- **Laços:**
 - > While
 - > Do - While
- **Controle de Fluxo**
 - > Break, Continue
 - > Exit, Return
 - > FLAGS (sinalizadores)

Informações Complementares e Atualizadas:

Consulte **REGULARMENTE**

O material disponível na COTEIA

2

Abril 2010

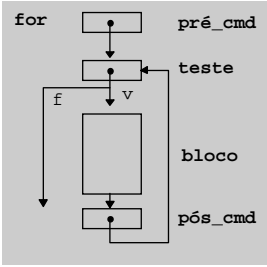
Comando de Laço FOR

Comando de Repetição: FOR

```
for (<pré_cmd>; <teste condição> ; <pós_cmd>)  
{  
  comandos;  
}
```

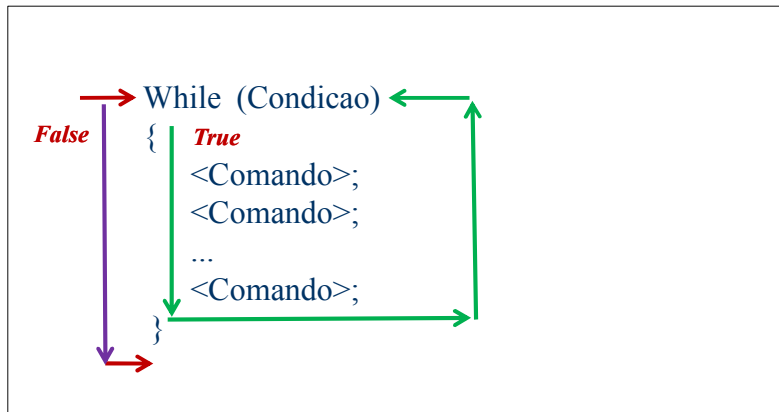
```
for (A = 1; A < 10 ; A++)  
{  
  printf ("%d \n", A);  
}  
printf ("FIM!\n");
```

```
A = 1;  
if (A < 10)  
{  
  printf ("%d \n", A);  
  A = A + 1;  
  <Laço: Continue no IF>  
}  
printf ("FIM!\n");
```



Fluxo de um Programa em "C": WHILE

Fluxo de Execução de um Programa => Repetição WHILE

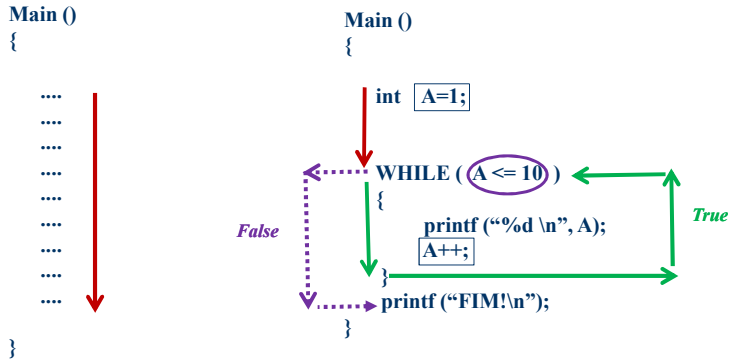


```
while (<condicao_verdadeira>)  
{  
  comandos;  
}
```

WHILE: Enquanto a condição for verdadeira, repita os comandos

Fluxo de um Programa em "C": WHILE

Fluxo de Execução de um Programa => **COM Repetição (LAÇOS)**



Fluxo Seqüencial: SEM DESVIOS!

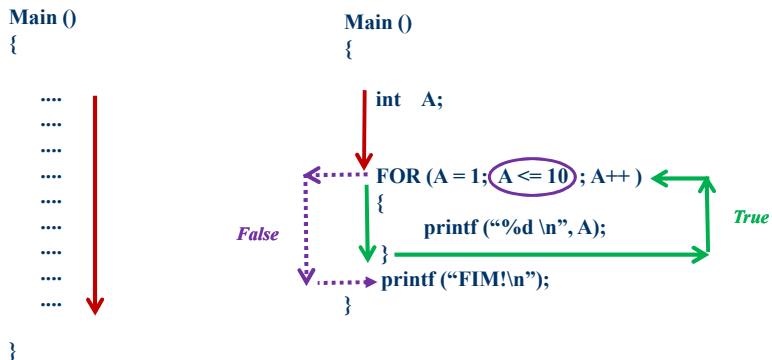
Fluxo com Laços: COM DESVIOS!

5

Abril 2010

Fluxo de um Programa em "C"

Fluxo de Execução de um Programa => **COM Repetição (LAÇOS)**



Fluxo Seqüencial: SEM DESVIOS!

Fluxo com Laços: COM DESVIOS!

6

Abril 2010

Comando de Laço WHILE

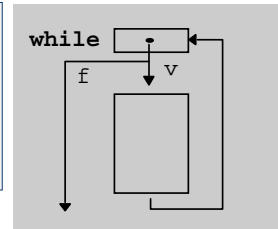
Comando de Repetição: WHILE

```
while (<teste condição> )
```

```
{  
  comandos;  
}
```

```
A = 1;  
while ( A <= 10 )  
{  
  printf ("%d \n", A);  
  A = A + 1;  
}  
printf ("FIM!\n");
```

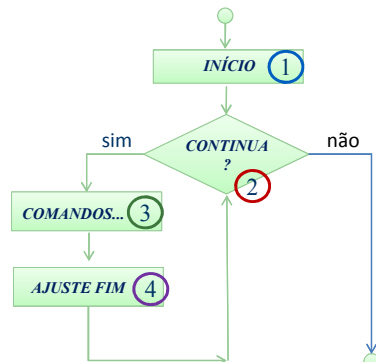
```
A = 1;  
if ( A <= 10 )  
{  
  printf ("%d \n", A);  
  A=A+1;  
  <Laço: Continue no IF>  
}  
printf ("FIM!\n");
```



Fluxo de Execução: Laço de Repetição

Comando de Laço: WHILE

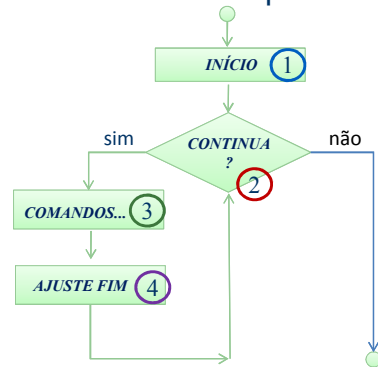
- Estrutura de laço que permite realizar uma repetição até que uma condição seja satisfeita. Usado para fazer contadores e outras repetições



```
A = 1; (1)  
while ( A <= 10 ) (2)  
{  
  printf ("%d \n", A); (3)  
  A = A+1; (4)  
}  
printf ("FIM!\n");
```

Comando de Laço: FOR

- Estrutura de laço que permite realizar uma repetição até que uma condição seja satisfeita. Usado para fazer contadores.



```
for (A = 1; A < 10 ; A++)
{
    printf ("%d \n", A);
}
printf ("FIM!\n");
```

WHILE

```
while ( <expr_inicial> )
    <comando>;
```

```
contador = 0;
while ( contador < nro_vezes )
    printf("Contando... %d \n", contador++);
```

```
contador = 10;
printf("Contagem regressiva...\n");
while ( contador != 0 )
{
    printf("Falta: %d \n", contador);
    contador--;
}
```

CUIDADO: while (contador < fim) ~~contador++;~~

<expr_inicial> é
uma expressão lógica [true, false]

<comando> pode ser único ou
substituído por um bloco { ... }

Comando de Laço **WHILE**

WHILE + **BREAK**

BREAK:
Comando para sair da execução
do bloco atual (“quebra fluxo”)
{ ... break; ... }

```
double Nota;  
  
while (1)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
  
    if (Nota >= 0.0 && Nota <= 10.0)  
        break;  
    else  
        printf("Nota Inválida! Digite novamente... \n");  
}
```

While (1)
“Infinito”

Comando de Laço **WHILE**

WHILE + **BREAK**

```
double Nota;  
  
while (1)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
  
    if (Nota >= 0.0 && Nota <= 10.0)  
        break;  
    else  
        printf("Erro! Digite novamente... \n");  
}
```

```
double Nota = -1.0;  
  
while (Nota < 0.0 || Nota > 10.0)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
    if (Nota < 0.0 || Nota > 10.0)  
        printf("Erro! Digite novamente... \n");  
}
```

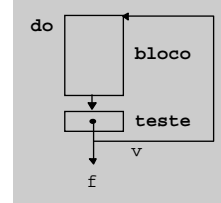
Comando de Laço **DO-WHILE**

DO-WHILE

```
do <comando>  
while ( <expr_inicial> );
```

```
contador = 0;  
do printf ("Contando... %d \n", contador++);  
while ( contador < nro_vezes );
```

```
contador = 10;  
printf("Contagem regressiva...\n");  
do {  
    printf ("Falta: %d \n", contador);  
    contador--;  
}  
while ( contador != 0 );
```



Comando de Laço **WHILE**

WHILE

DO-WHILE

BREAK

```
double Nota;  
  
while (1)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
  
    if (Nota >= 0.0 && Nota <= 10.0)  
        break;  
    else  
        printf("Erro! Digite novamente... \n");  
}
```

```
double Nota = -1.0;  
  
while (Nota < 0.0 || Nota > 10.0)  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
    if (Nota < 0.0 || Nota > 10.0)  
        printf("Erro! Digite novamente... \n");  
}
```

```
double Nota;  
  
do  
{  
    printf("Digite uma nota [0..10]: ");  
    scanf("%lf", &Nota);  
    if (Nota >= 0.0 && Nota <= 10.0) break;  
    printf("Erro! Digite novamente... \n");  
} while (1)
```

Comando de Laço: controle da repetição

BREAK

break; => Força o término da execução de um laço (for, while)

CONTINUE

continue; => Força o recomeço da execução de um laço (for, while)

```
while (1)
{
    tecla = le_tecla();
    if (tecla == 'x')      /* Tecla 'x' para sair do laço */
        break;
    if (tecla == 'r')      /* Tecla 'r' para recomeçar o laço */
        continue;
    ...
}
```

Comando de Laço: controle da repetição

EXIT

exit(0); => Término forçado da execução de TODO o programa.

RETURN

return; => Término da execução da ROTINA atualmente em
return(valor); execução pelo programa. Retorna da sub-rotina.
Se terminar a rotina *main()*, finaliza o programa.

```
main ()
{
    tecla = le_tecla();
    if (tecla == 'x') exit(0);      /* Tecla 'x' para sair do programa */
    if (tecla == 'r') return(0);    /* Tecla 'r' para retornar desta rotina */
    ...
}
```


Comando de Laço: controle da repetição FLAG

VETORES

Vetores Numéricos:

```
double N[10];
```

```
int Continuar = 1;  
int Qtde_Notas = 0;
```

```
while ( Continuar && Qtde_Notas < 10 )  
{  
    printf ("Entre com a nota %d ou -1 para sair: ", Qtde_Notas);  
    scanf ("%lf", &N[Qtde_Notas] );  
    if (N[Qtde_Notas] != -1)  
        Qtde_Notas ++;  
    else Continuar = 0; /* Se nota for igual a -1, sai do laço */  
}
```

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
0.0	1.0	2.0	3.0	4.0					

↑
Qtde_Notas

Continuar: esta variável é um “*senalizador*”
FLAGS ou Sinalizadores: são variáveis que servem para indicar um estado ou condição. Neste caso Continuar indica o controle do laço.

EXERCÍCIOS: USANDO WHILE e VETORES

Faça um programa que calcule a média de toda uma turma de alunos, baseado na fórmula abaixo...

$$\text{Media} = \frac{\sum_{i=1}^N \text{Nota}_i}{N}$$

N = Número de alunos
Nota_i = Nota do aluno i
Media = Média dos N alunos

Atenção: Não sabemos quantos alunos tem a turma. O programa deve ir lendo as notas até que o usuário indique que não deseja fornecer mais dados (usando um **flag** ou uma confirmação explícita).

Exercícios – Laço de Repetição

EXERCÍCIOS: USANDO WHILE (equivalentes aos do uso do FOR)

- Faça um programa que exiba na tela uma contagem regressiva de 10 até 0.
- Faça um programa onde o usuário determine os valores inicial e final de uma contagem (ler os valores inicial e final) e depois realizar uma contagem exibindo os valores na tela.
Nota: O valor inicial pode ser menor que o final (contagem crescente) ou pode ser maior que o final (contagem decrescente).
- Faça um programa que calcule o fatorial de um número fornecido pelo usuário.
Nota: Fatorial de N é $N * (N-1) * (N-2) * \dots * 1 \Rightarrow N!$
Fatorial de 0 é 1 $\Rightarrow 0!$ é igual a 1
- Faça um programa que leia 2 notas de cada aluno nas provas P1 e P2. As 2 notas devem ser válidas, ou seja, ter valores entre 0 e 10. Calcule a média ponderada destas 2 notas, onde a nota da P1 tem peso 1 e a nota da P2 tem peso 2, exibindo a média na tela. Repita este procedimento para uma turma de 10 alunos.



INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO

Web institucional: <http://www.icmc.usp.br/ssc/>

Página pessoal: <http://www.icmc.usp.br/~fosorio/>

E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

PAE Gustavo Pessin – E-mail: pessin [at] gmail .com

Monitor Matheus Lin – E-mail: matheus.lin [at] gmail.com

Disciplina de Introdução a Ciência da Computação

Web disciplina: Wiki ICMC - [Http://wiki.icmc.usp.br](http://wiki.icmc.usp.br)

> Programa, Material de Aulas, Critérios de Avaliação,

> Trabalhos Práticos, Datas das Provas, Notas