



# Programação de Computadores I

## Operadores de Condição

*if() / else if() / else{} / switch()*

**Gil Eduardo de Andrade**





# Operadores de Condição

## Introdução

- Operadores de condição permitem, ao programador, efetuar testes que determinam quais ações devem ser executadas pelo código-fonte escrito;
- Torna-se possível definir, em tempo de execução, qual fluxo de código deve ser seguido de acordo, normalmente com os dados de entrada informados pelo usuário;





# Operadores de Condição

## Exemplo Prático:

- Imagine que um aplicativo simples de cadastro de produtos está sendo codificado, onde o usuário pode: inserir, alterar, remover e visualizar produtos;
- É provável que um menu inicial com essas opções seja apresentado ao usuário;





# Operadores de Condição

## Exemplo Prático:

- Então a partir do menu inicial, o usuário escolherá uma das 4 opções;
- O trecho de código-fonte que deve ser executado, só será identificado em tempo em execução, quando o usuário escolher uma das opção;





# Operadores de Condição

## Exemplo Prático:

- A partir da opção escolhida, seu código deve ser capaz de testar, e assim identificar, qual funcionalidade deve ser executada naquele momento;
- Estes testes ou condições, em tempo de execução são feitos pelos operadores ou comandos de condição;





# Operadores de Condição

## Operador *if()*:

- O primeiro operador de condição, e também o mais básico, presente em praticamente todas as linguagens de programação, é o *if()*;
- O *if* ou *se* (tradução para o português) permite especificar, dentro dos parênteses, condições (verificações) que desejamos efetuar;





# Operadores de Condição

## Operador *if()*:

- O operador *if* possui como estrutura básica:
  - *if(condição) {*  
*}*
    - *if* → é nome do comando, assim como *printf* e *scanf*;
    - *()* → delimita o início e término da condição;
    - *condição* → é a verificação que será efetuada;
    - *{}* → delimita o início e o término do código que será executado caso a condição seja verdadeira;



# Operadores de Condição

## Exemplo Básico:

```
#include <stdio.h>
```

```
int main() {
```

```
    int a = 2;
```

```
    if(a == 2) {
```

```
        printf("A variável 'a' armazena um valor par!");
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

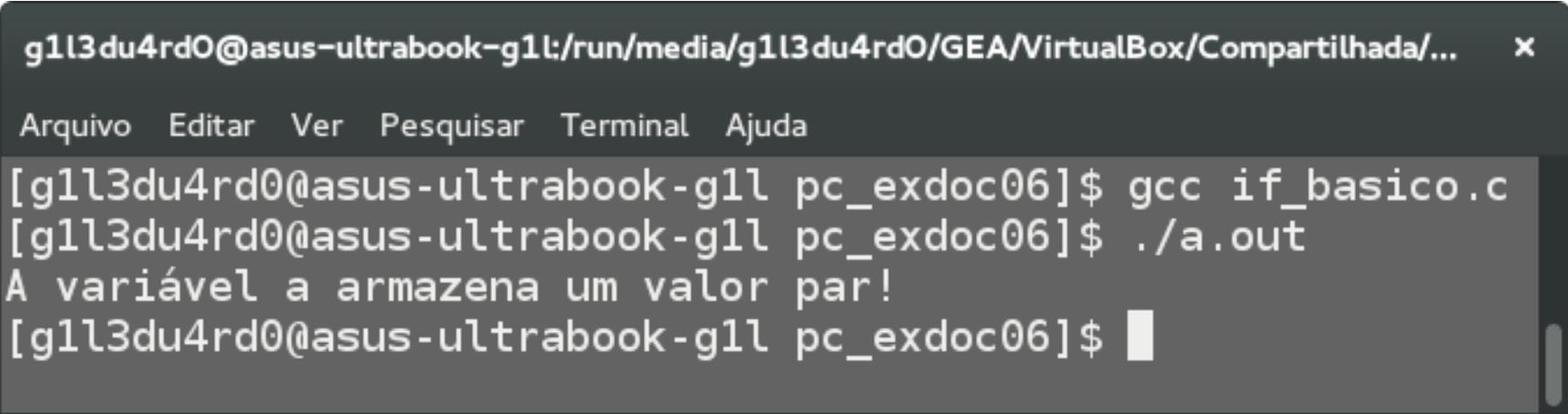
```
}
```

Observe que a condição colocada dentro do comando *if* foi “*a == 2*”, que juntos significam: *se “a” for igual a 2*, logo após o término da condição, indicado pelo *)*, temos o “abre chaves” *{*, que indica o início do trecho de código que deve ser executado se a condição (*a==2*) for verdadeira, e deve ser lido como *então*. O “fecha chaves” *}* indica o final do trecho de código a ser executado quando a condição do *if* for verdadeira. A seguir são apresentados os principais operadores aritméticos de comparação utilizados juntamente com o comando *if*, além do “*==*” visto no exemplo;



# Operadores de Condição

## Exemplo Básico:



```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc if_basico.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
A variável a armazena um valor par!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```





# Operadores de Comparação

## Aritméticos:

- São utilizados para comparação entre valores:
  - $a==b$  → compara se '*a*' é igual a '*b*' →  $if(a==a);$
  - $a!=b$  → compara se '*a*' é diferente de '*b*' →  $if(a!=b);$
  - $a>b$  → compara se '*a*' é maior que '*b*' →  $if(a>b);$
  - $a>=b$  → compara se '*a*' é maior ou igual a '*b*' →  $if(a>=b);$
  - $a<b$  → compara se '*a*' é menor que '*b*' →  $if(a<b);$
  - $a<=b$  → compara se '*a*' é menor ou igual a '*b*' →  $if(a<=b);$



# Operadores de Condição

## Exemplo Básico – Operador Aritmético “!=“:

```
#include <stdio.h>

int main() {
    int a = 2;
    if(a != 2) {
        printf("O valor armazenado em 'a' é diferente de 2");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição definida : “*a != 2*” (‘*a*’ diferente de 2) não é verdadeira, por isso o trecho de código definido entre as chaves {} (comando *printf()*) não é executado.

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc if_diferente.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out

[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```

# Operadores de Condição

## Exemplo Básico – Operador Aritmético “>”:

```
#include <stdio.h>

int main() {
    int a = 2;

    if(a > 0) {
        printf("0 valor armazenado em 'a' é maior que 0");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição definida : “*a > 0*” (*‘a’ maior que 2*) é verdadeira, por isso o trecho de código definido entre as chaves {} (*comando printf()*) é executado.

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc if_maior.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
0 valor armazenado em 'a' é maior que 0
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```

# Operadores de Condição

## Exemplo Básico – Operador Aritmético “>=”:

```
#include <stdio.h>
```

```
int main() {
```

```
    int a = 2;
```

```
    if(a >= 2) {
```

```
        printf("O valor armazenado em 'a' é maior ou igual a 0");
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

Observe que a condição definida : “*a* >= 2” (“*a* maior ou igual a 2”) é verdadeira, por isso o trecho de código definido entre as chaves {} (comando *printf()*) é executado.

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
```

```
Arquivo Editar Ver Pesquisar Terminal Ajuda
```

```
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc if_maior_igual.c
```

```
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
```

```
0 valor armazenado em 'a' é maior ou igual a 2
```

```
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```

# Operadores de Condição

## Exemplo Básico – Operador Aritmético “<”:

```
#include <stdio.h>

int main() {
    int a = 2;

    if(a < 2) {
        printf("O valor armazenado em 'a' é menor que 2");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição definida : “*a < 2*” (‘*a*’ menor que 2) não é verdadeira, por isso o trecho de código definido entre as chaves {} (comando *printf()*) não é executado.

```
g1l3du4rd0@asus-ultrabook-g1l/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc if_menor.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out

[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```

# Operadores de Condição

## Exemplo Básico – Operador Aritmético “<=”:

```
#include <stdio.h>

int main() {
    int a = 2;
    if(a <= 2) {
        printf("O valor armazenado em 'a' é menor ou igual a 2");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição definida : “*a <=2*” (*‘a’ menor ou igual a 2*) é verdadeira, por isso o trecho de código definido entre as chaves {} (comando *printf()*) é executado.

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc if_menor_igual.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
0 valor armazenado em 'a' é menor ou igual a 2
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```



# Operadores de Condição

## Operador *else*:

- O operador de condição ***else*** ou ***senão*** (tradução para o português) é utilizado em contrapartida a um operador ***if()***, sendo assim, ele só pode ser usado se antes houver um ***if()***;
- Portanto, não existe operador ***else*** sem antes existir um operador ***if()*** visto que o ***else*** é a negação para condição especificada no ***if()***;





# Operadores de Condição

## Operador *else*:

- O operador ***else*** possui como estrutura básica:
  - ***else*** { *(obs.: o operador “else” não possui condição)*  
}
  - ***else*** → é nome do comando, assim como *printf* e *scanf*;
  - **{}** → indica o início e término do código que será executado caso a condição do ***if*** anterior ao ***else*** e ao qual ele está atrelado for ***falsa***;



# Operadores de Condição

## Exemplo Básico:

```
#include <stdio.h>

int main() {

    int a = 1;

    if(a == 2) {
        printf("A variável 'a' armazena o valor 2!");
    }
    else {
        printf("A variável 'a' não armazena o valor 2!");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição colocada dentro do comando *if* (*a == 2*), não é verdadeira, sendo assim o trecho de código especificado entre as chaves *{}* do *if* não é executado. Logo, como o operador *else* foi definido para o *if* (*a==2*) e a condição é falsa, o trecho de código definido entre as chaves *{}* do *else* é executado!



# Operadores de Condição

## Exemplo Básico:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc else_basico.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
A variável 'a' não armazena o valor 2!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```





# Operadores de Condição

## Operador *else if()*:

- O operador de condição ***else if()*** ou ***senão se*** (tradução para o português) é utilizado em contrapartida a um operador ***if()***, sendo assim, ele só pode ser usado se antes houver um ***if()***;
- Portanto, não existe operador ***else if()*** sem antes existir um operador ***if()***;
- Ao contrário do ***else***, o operador ***else if()*** possibilita que uma segunda condição seja verificada em contrapartida a já especificada no ***if()***;





# Operadores de Condição

## Operador *else if*():

- O operador *else if* possui como estrutura básica:
  - *else if(condição) {*  
*}*
    - *else if* → é nome do comando, assim como *printf*;
    - *()* → delimita o início e término da condição;
    - *condição* → é a verificação que será efetuada;
    - *{}* → delimita o início e o término do código que será executado caso a condição seja verdadeira;



# Operadores de Condição

## Exemplo Básico:

```
#include <stdio.h>

int main() {
    int a = 1;

    if(a == 2) {
        printf("A variável 'a' armazena o valor 2!");
    }
    else if(a == 1) {
        printf("A variável 'a' armazena o valor 1!");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição colocada dentro do comando *if (a == 2)*, não é verdadeira, sendo assim o trecho de código especificado entre as chaves *{}* do *if* não é executado. Logo, como o operador *else if* foi definido para a *if (a==2)* e possui como condição "*a ==1*" que é verdadeira, o trecho de código definido entre as chaves *{}* do *else if* é executado!



# Operadores de Condição

## Exemplo Básico:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc elseif_basico.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
A variável 'a' armazena o valor 1!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ █
```



# Operadores de Condição

## Implementando Exemplo Prático: (Cadastro de Produtos)

```
#include <stdio.h>

int main() {

    int op;

    printf("=====");
    printf("\n== CADASTRO DE PRODUTOS ==");
    printf("\n=====");
    printf("\n1) Cadastrar Produto");
    printf("\n2) Alterar Produto");
    printf("\n3) Remover Produto");
    printf("\n4) Visualizar Produtos");
    printf("\n5) Sair");
    printf("\n=====");
    printf("\n> ");
    __fpurge(stdin);
    scanf("%i", &op);
```

A primeira parte do código-fonte apresenta um menu de opções ao usuário, possibilitando ao mesmo escolher a funcionalidade que deseja utilizar, a sua escolha será identificada a seguir.

Continuação do código-fonte no próximo slide.

# Operadores de Condição

## Implementando Exemplo Prático: (Cadastro de Produtos)

```
if(op == 1) { // Cadastrar Produto
    printf("\nCADASTRAR");
}
else if(op == 2) { // Alterar Produto
    printf("\nALTERAR");
}
else if (op == 3) { // Remover Produto
    printf("\nREMOVER");
}
else if (op == 4) { // Visualizar Produtos
    printf("\nVISUALIZAR");
}
else if (op == 5) { // Sair
    printf("\nSAIR");
}
else { // Opção Inválida
    printf("\nOPÇÃO INVÁLIDA");
}
```

Considerando que a aplicação desenvolvida disponibiliza ao usuário várias opções ou funcionalidades, e, o mesmo precisa selecionar uma delas para ser executada de cada vez, a utilização dos operadores de condição: ***if(){} else if(){} else{}***  se faz necessária no intuito de identificar qual das opções disponibilizadas foi selecionada, e então executar o trecho de código que implementa tal função.

# Operadores de Condição

## Implementando Exemplo Prático: (Cadastro de Produtos)

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/Vir... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
=====
== CADASTRO DE PRODUTOS ==
=====
1) Cadastrar Produto
2) Alterar Produto
3) Remover Produto
4) Visualizar Produtos
5) Sair
=====
> 2

ALTERAR
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```

Observe que durante a execução do código exemplo apresentado nos *slides* anteriores, o usuário escolheu a opção "2", sendo verdadeira a condição colocada no operador *else if(op==2)*, sendo executado o código especificado entre as suas chaves *{ printf("ALTERAR"); }*.



# Operadores de Combinação

## Lógicos (&& / ||):

- São utilizados para combinar uma ou mais condições nos operadores *if()* e *else if()*;
- Existem dois operadores de combinação lógicos, básicos, que podem ser utilizados na linguagem C:
  - **E**: representado pelos caracteres **&&**;
  - **Ou**: representado pelos caracteres **||**;





# Operadores de Combinação

## Lógicos:

- Quando utilizamos o operador lógico **&& (e)** para combinar várias condições, o resultado desta combinação só será verdadeiro se todas as condições que foram combinadas também forem verdadeiras;
  - **Exemplo:** `if(a==0 && b==1) { }`





# Operadores de Combinação

## Lógicos:

- Quando utilizamos o operador lógico ***// (ou)*** para combinar várias condições, o resultado desta combinação será verdadeiro se uma ou mais das condições combinadas também for verdadeira;
  - ***Exemplo: if(a==0 || b==1) { }***



# Operadores de Combinação

## Lógicos – Tabela Verdade:

- Operadores lógicos podem ser representados por tabelas, denominadas, tabela verdade:
  - Considere que **A** representa a condição: “**a==0**”;
  - Considere que **B** representa a condição: “**b==0**”;

A	B	A && B
F	F	F
V	F	F
F	V	F
V	V	V

A	B	A    B
F	F	F
V	F	V
F	V	V
V	V	V

# Operadores de Condição

## Exemplo Básico:

```
#include <stdio.h>
```

```
int main() {
```

```
    int a = 1, b = 2;
```

```
    if(a == 1 && b == 1) {  
        printf("As duas condições são verdadeiras!");
```

```
    }
```

```
    else {
```

```
        printf("Ao menos uma das condições é falsa!");
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

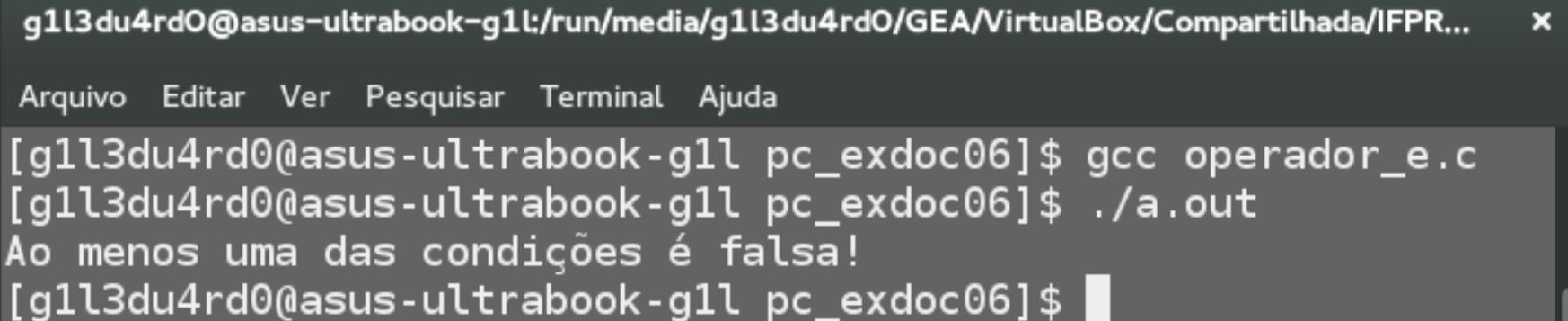
```
}
```

Observe que a condição colocada dentro do comando *if* (*a == 1 && b == 1*), não é totalmente verdadeira, ou seja, uma ou mais condições são falsas, considerando que o operador lógico de combinação utilizado foi o “e” (&&) a condição final é falsa e o trecho de código especificado entre as chaves *{}* do *if* não é executado.



# Operadores de Condição

## Exemplo Básico:



```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc operador_e.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
Ao menos uma das condições é falsa!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```



# Operadores de Condição

## Exemplo Básico:

```
#include <stdio.h>

int main() {

    int a = 1, b = 2;

    if(a == 1 || b == 1) {
        printf("Uma das, ou, as duas condições são verdadeiras!");
    }
    else {
        printf("As duas condições são falsas!");
    }

    printf("\n");
    return 0;
}
```

Observe que a condição colocada dentro do comando *if* ( $a == 1 || b == 1$ ), não é totalmente verdadeira, ou seja, uma ou mais condições são falsas, porém ao menos uma condição é verdadeira. Considerando que o operador lógico de combinação utilizado foi o “ou” ( $||$ ) a condição final é verdadeira e o trecho de código especificado entre as chaves  $\{\}$  do *if* é executado.



# Operadores de Condição

## Exemplo Básico:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IFPR... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc operador_ou.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
Uma das, ou, as duas condições são verdadeiras!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$
```





# Operadores de Condição

## Operador *switch()*:

- Na linguagem C, assim como na maioria das outras linguagens, também encontra-se disponível o operador de condição ***switch()***;
- O comando ***switch()*** possui a mesma funcionalidade dos comandos ***if(){} else if(){} else{}*** , contudo possui uma sintaxe e organização distinta destes;





# Operadores de Condição

## Operador *switch()*:

- O operador ***switch*** possui como estrutura básica:
  - ***switch(variável) {***
    - case 1:***
      - break;***
    - case 4***
      - break;***
    - default:***
      - break;***
  - }***





# Operadores de Condição

## Operador *switch()*:

- ***switch*** → é nome do comando, assim como *printf* e *scanf*;
- ***variável*** → indica a variável do qual o valor será testado, ou seja, o qual será igual;
- ***case*** → indica uma possibilidade que deseja-se testar para o valor da variável;
- ***1, 4*** → indica o valor a ser testado para variável, funciona de forma análoga ao *if(){} e else if(){};*
- ***break*** → indica o término do código a ser executado caso a condição seja verdadeira, análogo ao fecho chave *};*
- ***default*** → funciona de forma análoga ao *else{};*, ou seja, o código que deve ser executado se nenhum case anterior for verdadeiro;



# Operadores de Condição

## Exemplo *switch()*:

```
int var;

printf("Digite um valor para a: ");
__fpurge(stdin); //fflush(stdin) - Windows;
scanf("%i", &var);

switch(var) {

    case 1:
        printf("'var' armazenada 1!");
        break;

    case 2:
        printf("'var' armazenada 2!");
        break;

    default:
        printf("'var' armazenada valor desconhecido!");
        break;

}
```

O comando *switch()* deve ser interpretado da seguinte forma:

- 1) *switch(var)* → verifique o valor armazenado em 'var'
- 2) *case 1:* → caso o valor seja igual a um, então execute até encontrar o *break*;
- 3) *case 2:* → caso o valor seja igual a dois, então execute até encontrar o *break*;
- 4) *default:* → por padrão, caso o valor não seja igual a nenhum dos *cases* anteriores, então execute até encontrar o *break*;

# Operadores de Condição

## Exemplo *switch()*:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/l... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ gcc switch.c
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
Digite um valor para a: 1
'var' armazena 1!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
Digite um valor para a: 2
'var' armazena 2!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ ./a.out
Digite um valor para a: 3
'var' armazena valor desconhecido!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc06]$ █
```



# Operadores de Condição

## Exemplos Utilizados no Documento

[http://www.gileduardo.com.br/ifpr/pci/downloads/pc\\_exdoc06.zip](http://www.gileduardo.com.br/ifpr/pci/downloads/pc_exdoc06.zip)

## Mais Exemplos sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/pci/downloads/pc\\_ex06.zip](http://www.gileduardo.com.br/ifpr/pci/downloads/pc_ex06.zip)

## Exercícios sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/pci/downloads/pc\\_pratica06.pdf](http://www.gileduardo.com.br/ifpr/pci/downloads/pc_pratica06.pdf)

