Tecnologia em Análise e Desenvolvimento de Sistemas - TADS

Estrutura de Dados

TADS / IFRS - 2022-2

Prof. Luciano Vargas Gonçalves

E-mail: luciano.goncalves@riogrande.ifrs.edu.br





Linguagem C e Ambiente de Programação

Linguagem C e Compilador C

- Tutoriais para Leitura e Consulta sobre a linguagem C
 - http://www.inf.ufrgs.br/~binsely/tutorialc.pdf
 - https://www.cenapad.unicamp.br/servicos/treinamentos/apostila s/apostila_C.pdf

Linguagem C e Compilador C

- Instalação e Configuração Linux e Windows
 - Linux
 - Link 1 Ubuntu 18.04
 - Link 2 Vídeo
 - Windows
 - Link 1 Tutorial para Windows 10
 - Link 2 Vídeo para instalar o MinGW

Editar e compilar um código C

- Código mínimo Hello World!!
 - Editor VsCode
 - Arquivo Teste.c
 - #Include < >
 - Função main()

```
Programas C > Aula1 > C Teste1.c > ...

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6  printf("\n Hello World!! \n");
7  }
8
```

Editar e compilar um código C

- Código mínimo Hello World!!
 - Compilar no Terminal.
 - Acessar a pasta "Aula1" e o arquivo fonte "Teste.c".
 - Compilar o arquivo com o compilador <u>qcc</u> do linux, executar o arquivo com "./"

Olá Mundo em C

Linguagem C é Compilada (Linguagem máquina)

- Para compilar: (verifica a sintaxe)
 >> gcc fonte.c <ENTER>
- Para compilar e gerar executável:
 >> gcc fonte.c -o nomeexec <ENTER>
- Executando:>> ./nomeexec <ENTER>

OBS: Se o nome do executável não for informado o default é a.out

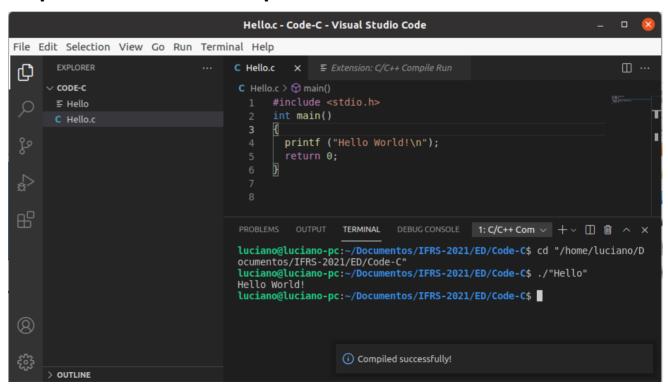
Instalação do VsCode:

- Download VsCode
 - https://code.visualstudio.com/
- Tutorial de instalação
 - Windows
 - https://code.visualstudio.com/docs/languages/cpp
 - Linux
 - https://sempreupdate.com.br/como-instalar-o-visual-studio-code-no-ubuntu-20-04/



Instalação do VsCode:

- Precisa criar uma pasta e um arquivo Hello.c
- Tecla F6
 - Compilar



Introdução a Programação em C

- Faça uma revisão de algoritmos e implementações usando a linguagem C;
 - Avalie a Sintaxe da linguagem (estruturas);
 - Avalie a Semântica da linguagem (operações);
 - Livro online:
 - http://www.inf.ufpr.br/lesoliveira/download/c-completo-total.pdf
 - Desenvolva algoritmos com C
 - Exemplos e exercícios Online
 - Revisar até a aula de Strings
 - https://programacaodescomplicada.wordpress.com/indice/linguagem-c/
 - https://www.youtube.com/c/excriptvideo/search?query=programa%C3%A7%C3%A30%20C

Revisão da Linguagem C

- Variável e Declaração;
 - Espaço de memória de um certo tipo de dado associado a um nome para referenciar seu conteúdo.
 - Em C a **Tipagem forte**, necessita declaração.

```
int idade; Tipo e nome idade = 30; printf (" A idade é : %d", idade);
```

Linguagem C

Nome de Variáveis

- Quantos caracteres quiser (32);
- Comece com letras ou sublinhado:
 - Seguidos de letras, números ou sublinhados
- Linguagem C é sensível ao caixa:

```
peso # Peso # pEso
```

- Não podemos definir um identificador com o mesmo nome que uma palavra-chave
 - Auto, static, extern, int, long, if, while,

Linguagem C

Declaração de Variáveis - Tipos primitivos

tipo	bytes	escala
char	1	-128 a 127
int	2	-32.768 a 32.767
float	4	3.4e-38 a 3.4e+38
double	8	1.7e-308 a 1.7e+308

Long ou Long int (4 bytes)
Unsigned Char (0 a 255)
Unsigned int (0 a 65.535)

Linguagem C

Declaração de Variáveis - Tipos primitivos

```
#include <stdio.h>
         main()
            int soma=10;
float money=2.21;

declaração

char letra= 'A';

double pi=2.01E6;
            printf ("valor da soma = %d\n'', soma);
            printf ("Valor de Money = %f\n", money);
            printf("Valor de Letra = %c\n", letra);
            printf("Valor de Pi = %e\n", pi);
                           Introdução Linguagem C
```

Entrada de Dados

- Leitura de dados tipados via teclado
 - Scanf ("string de controle", lista de argumentos);

```
Exemplo:
scanf(``%d",&idade);
```

OBS: Para sequência de caracteres (%s), o caracter & não deverá ser usado.

```
printf("\nInforme sua idade:");
//entrada de dados com scanf
scanf("%d",&idade);
printf("Sua idade é: %d \n",idade);
}
```

Entrada de Dados

- Leitura de dados tipados via teclado
 - Scanf ("string de controle", lista de argumentos);

```
int main(){
  int idade;
  printf("\nInforme sua idade:");
  //entrada de dados com scanf
  scanf("%d",&idade);
  printf("Sua idade é: %d \n",idade);
}
```

Saída no Terminal

• luciano@luciano-pc:~/Documentos/IFRS2022/Disciplinas/Primeiro/Ed1/Exemplos/Aula1\$./"Exemplo1"

```
Informe sua idade:45
Sua idade é: 45
```

Entrada e Saída de Dados – Caracteres de formatação

Caracteres especiais e de formatação texto

- %c → caracter
- %d → inteiro
- %e → número ou notação científica
- %f → ponto flutuante
- % $o \rightarrow octal$
- %x → hexadecimal
- %s → string (cadeia de caracteres)
- %If → double

Entrada e Saída de Dados

Exemplos de usos dos caracteres especiais

```
int main(){
    char ch:
    printf("Digite um caracter:");
    scanf("%c",&ch);
    printf(" %c = %d em decimal Asc ",ch,ch);
    printf("\n %c = %o em octal, %x em hexadecimal\n ",ch,ch,ch);
   Digite um caracter:A
    A = 65 em decimal Asc
    A = 101 em octal, 41 em hexadecimal
     Saída no Terminal
```

Endereço de memória

- Nome e o Tipo de uma variável é necessário para o programador solicitar ao computador(compilador) a reserva de memória para armazenar uma ou mais informações;
- O nome da variável só é importante para o programador localizar a informação em seu programa;
- O computador usa para identificar uma variável ou espaço da memória o seu endereço de memória;
- Toda variável ocupa uma porção de memória conforme seu tipo de dados. Para cada porção de memória o primeiro byte ocupado por ela é o seu endereço na memória (RAM);

Endereço de Memória

- Exemplo: Variavel idade (int) ocupa 4bytes
 - CC4453F4 é o seu endereço

```
Endereço
                                                                          Informação
       int main(){
           int idade;
                                                              cc4453f2
           printf("\nInforme sua idade:");
           scanf("%d",&idade);
                                                              cc4453f3
           printf("Sua idade é: %d \n",idade);
                                                              cc4453f4
                                                              cc4453f5
Informe sua idade:45
                                                                              45
Sua idade é: 45
                                                              cc4453f6
Variável idade está armazenada na posição de memória cc4453f4
                                                              cc4453f7
   Saída no terminal
                                                              cc4453f8
```

Endereço de Memória

• Exemplo: idade, idade2 ocupam diferentes posições de memória, espaçadas 4Bytes

```
Endereço
                                                                                              Informação
int main(){
   int idade,idade2; —
   printf("\nInforme sua idade:"):
                                                                                d2559960
   scanf("%d",&idade);
                                                                                 d2559961
   printf("\nInforme idade seu irmão:");
                                                                                                  45
   scanf("%d",&idade2);
                                                                                 d2559962
   printf("Sua idade é: %d seu irmão é: %d \n",idade,idade2);
                                                                                 d2559963
   printf("Variável idade está armazenada na posição de memória %x \n",&idade);
                                                                                 d2559964
   printf("Variável idade2 está armazenada na posição de memória %x \n",&idade2);
                                                                                 d2559965
    Informe sua idade: 45
                                                                                                  38
                                                                                 d2559966
    Informe idade seu irmão:38
    Sua idade é: 45 seu irmão é: 38
                                                                                 d2559967
    Variável idade está armazenada na posição de memória d2559960
    Variável idade2 está armazenada na posição de memória d2559964
```

Atribuição de Endereço de memória "&"

- Quando usamos o operador "&" para acesso e atribuição à endereços de memória;
- Exemplo:
 - Usado no Scanf, para atribuir uma valor a uma posição de memória

```
scanf("%d",&idade);
```

Atribuição de Endereço de memória "&"

- Quando usamos o operador "&" para acesso e atribuição à endereços de memória;
- Exemplo:
 - Usado para consultar o endereço de uma variável:

```
printf("Variável idade está armazenada na posição de memória %x \n",&idade);
```

Comando printf formatação para REAIS

```
vint main(){
      int total = 350;
      //saida de dados "printf()"
      printf("o n° de aluno são %d \n",total);
      printf("o n° de aluno são %2d \n",total);
      printf("o n° de aluno são %4d \n",total);
      printf("o n° de aluno são %10d \n".total);
                      o n° de aluno são 350
                      o n° de aluno são 350
                      o n° de aluno são 350
                      o n° de aluno são
                                             350
```

int main(){

Comando printf formatação para REAIS

```
double total = 35.123456789;
//saida de dados "printf()"
printf("o n° de aluno são %f \n",total);
printf("o n° de aluno são %2.2f \n",total);
printf("o n° de aluno são %2.4f \n",total);
printf("o n° de aluno são %4.10f \n".total);
              o n° de aluno são 35.123457
              o n° de aluno são 35.12
              o n° de aluno são 35.1235
              o n° de aluno são 35.1234567890
                   Ocupa 10 posições após a vírgula
```

Comando printf formatação para REAIS

```
vint main(){
      float total = 35.123456789;
     //saida de dados "printf()"
      printf("o n° de aluno são %f \n",total);
      printf("o n° de aluno são %2.2f \n",total);
      printf("o n° de aluno são %2.4f \n",total);
      printf("o n° de aluno são %4.10f \n",total);
                o n° de aluno são 35.123455
                o n° de aluno são 35.12
                o n° de aluno são 35.1235
                o n° de aluno são 35.1234550476
```

Ocupa 10 posições após a vírgula

Float vs Double precisão de representação

```
int main(){
    float a = 35.111111111111:
    float b = 35.111111111111;
    float c=a+b;
    //saida de dados "printf()"
    printf("A %4.20f \n",a);
    printf("B %4.20f \n",b);
    printf("C %4.20f \n",c);
      A 35.11111068725585937500
      B 35.11111068725585937500
      C 70.22222137451171875000
```

```
int main(){
    double a = 35.111111111111:
    double b = 35.111111111111;
    double c=a+b;
    //saida de dados "printf()"
    printf("A %4.20f \n",a);
    printf("B %4.20f \n",b);
    printf("C %4.20f \n",c);
 A 35.111111111111100058224
 B 35.111111111111100058224
 C 70.222222222200116448
```

Exemplo – Array de Char (String)

Programa para ler o nome e a idade de uma pessoa, Após mostrar na tela os valores:

```
C Hello.c (deleted)
                   C Exemplo1.c ●
Aula1 > C Exemplo1.c > 分 main()
      #include <stdio.h>
                                                               String e Vetor não usam "&"
      int main()
        int idade:
        char nome[30];
        printf("Digite 0 seu nome:");
        scanf("%s", nome);
        printf("Informe sua Idade:");
        scanf("%d",&idade);
        printf("Pessoa %s possui %d anos de idade. \n",nome,idade);
        return 0:
 12
                  TERMINAL
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$ ./"Exemplo1"
Digite o seu nome:Joao
Informe sua Idade:34
Pessoa Joao possui 34 anos de idade.
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$
```

Operadores Aritméticos

Operador	Ação
+	Adição
*	Multiplicação
/	Divisão
%	Resto da divisão inteira
-	Subtração (unário)
	Decremento
++	Incremento

Operadores relacionais e lógicos

Operador	Ação
>	Maior que
>=	Maior ou igual que
<	Menor que
<=	Menor ou igual que
==	Igual a
!=	Diferente de
&&	Condição "E"
П	Condição "OU"
!	Não

Operadores relacionais e lógicos

Em C: ZERO representa FALSO UM representa VERDADE

```
int testel, teste2;
       teste1 = 12 > 10:
       teste2 = 12 < 10:
       printf("Testes 1 resultado %d\n", teste1);
       printf("Testes 2 resultado %d\n",teste2);
       if(testel)
         printf("Verdade = %d \n", testel);
       else
         printf("Falso = %d \n", testel);
       if(teste2)
         printf("Verdade = %d \n", teste2);
       else
         printf("Falso = %d \n", teste2);
20
       return 0;
                                       PROBLEMS
         OUTPUT
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$ ./"Exemplo2"
Testes 1 resultado 1
Testes 2 resultado 0
Verdade = 1
Falso = 0
```

Comparação

Pré Incremento ++X;

Pós incremento X++;

```
Aula1 > C Exemplo3.c > 分 main()
        int c = 0, d = 0;
         printf("Valor de C= %d\n",++c);
         //incremente apos atribui
        printf("Valor de C= %d\n",c);
        printf("Valor de D= %d\n",d++);
 11
        //atribui e apos incrementa
         printf("Valor de D= %d\n",d);
 12
        return 0;
 14
PROBLEMS
           OUTPUT
                             DEBUG CONSOLE
                   TERMINAL
luciano@luciano-pc:~/Documentos/IFRS-2021/E
Valor de C= 1
Valor de C= 1
Valor de D= 0
Valor de D= 1
luciano@luciano-pc:~/Documentos/IFRS-2021/E
```

Comparação

Operador SIZEOF

 Este operador retorna o tamanho da variável ou tipo que está em seu operando.

```
Aula1 > C Exemplo4.c > 分 main()
      #include <stdio.h>
      int main()
        int a = 0:
        float b = 10.04f;
        char c = 'D':
        double d = 12.565:
        printf("Inteiro A= %d \t\t0cupa %d Bytes\n",a, sizeof(a));
        printf("Float B= %f \t0cupa %d Bytes \n",b, sizeof(b));
 10
        printf("Char C= %c \t\t0cupa %d Bytes \n",c, sizeof(c));
        printf("Double D= %f \t0cupa %d Bytes \n",d, sizeof(d));
        return 0:
PROBLEMS
          OUTPUT
                  TERMINAL
                            DEBUG CONSOLE
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$ ./"Exemplo4"
Inteiro A= 0
                       Ocupa 4 Bytes
Float B= 10.040000
                       Ocupa 4 Bytes
Char C= D
                       Ocupa 1 Bytes
Double D= 12.565000
                       Ocupa 8 Bytes
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$
```

Operadores de Seleção - IF

- Permitir testes para decidir ações alternativas:
 - if
 - if else
 - switch
 - (?:) Operador Condicional

Operadores de Seleção - IF

IF/ELSE cascata

```
int main(){
   int a = 2;
   char op;
   if( a==0)
       printf("\nA igual a ZERO\n");
   else if( a==1)
       printf("\nA igual a UM\n");
   else
       printf("\nA diferente de Zero e UM\n");
}
```

Operadores de Seleção - Switch

 Comando getchar() captura um caracter;

Comando Switch / Case

```
int main(){
    int a = 2;
    switch (a){
        case 0:
            printf("\nA igual a ZERO\n");
        break;
        case 1:
            printf("\nA igual a UM\n");
        break;
        default:
            printf("\nA diferente de Zero e UM\n");
```

Operadores de Seleção – Ternário ?

- Operador Ternário ?
 - Forma compacta de expressar uma instrução if eles
 - (condição) ? expressão1 : expressão2;
 - Exemplo:

Estrutura de Repetição - For

for (<início>;<condição>;<incremento>) <comando>;

Na forma mais simples:

- Inicialização:
 - expressão de atribuição
 - sempre executada uma única vez
- Teste:
 - condição que controla a execução do laço
 - é sempre avaliada a cada execução
 - verdadeiro → continua a execução
 - falso → para a execução

Estrutura de Repetição - For

```
for (x=0,y=0;x+y<100;++x,y=y+x)
printf("%d",x+y);
```

Exemplo com duas variáveis de condições

```
Aula1 > C Exemplo7.c > 🕅 main()
      #include <stdio.h>
      int main()
        int x,y;
        for (x=0, y=0; x+y<100; ++x, y=x+y)
          printf("X=%d Y=%d = X+Y=%d\n",x,y,x+y);
        return 0;
PROBLEMS
                   TERMINAL
                             DEBUG CONSOLE
          OUTPUT
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$ ./"E
X=0 Y=0 = X+Y=0
X=1 Y=1 = X+Y=2
X=2 Y=3 = X+Y=5
X=3 Y=6 = X+Y=9
X=4 Y=10 = X+Y=14
X=5 Y=15 = X+Y=20
X=6 Y=21 = X+Y=27
X=7 Y=28 = X+Y=35
X=8 Y=36 = X+Y=44
X=9 Y=45 = X+Y=54
X=10 Y=55 = X+Y=65
X=11 Y=66 = X+Y=77
X=12 Y=78 = X+Y=90
luciano@luciano-pc:~/Documentos/IFRS-2021/ED/Code-C/Aula1$
```

Estrutura de Repetição – For - LOOP

```
for(;;)
printf("Este loop rodará eternamente!\n");
```

Exemplos

```
for(i=0;i<10;i++);
```

 A presença do ponto e vírgula finalizando o comando, força a execução do loop sem que seja executado qualquer outro comando.

Estrutura de Repetição – While

```
while <condição> <comando>;
```

```
Exemplo: Contagem
#include <stdio.h>
main()
{
int i=0;
while (i < 10) {
        printf("%d",i);
        i++;
    }
}</pre>
```

 O loop se repete, enquanto a condição for verdadeira

Estrutura de Repetição – do / while

 Ao contrário das estruturas "for" e "while" que testam a condição no começo do loop, "do / while" sempre a testa no final, garantido a execução ao menos uma vez da estrutura.

```
do {
            <comandos>;
} while <condição>;
```

```
Exemplo: Término determinado pelo usuário.
#include < stdio.h>
main()
{
int num;
do {
   scanf("%d",&num);
} while (num < 100);
}
```

Próxima Aula:

- Ponteiros e Structs
- Leiam e testem os exemplo da material até a seção 7
 - http://www.inf.ufrgs.br/~binsely/tutorialc.pdf