

Laboratório 1

Introdução à Programação C (CMT012)

Prof. Ronald Souza

IC/UFRJ — 16/08/2023

Google Classroom: 4v5wlvf

Objetivo

Familiarizar-se com as ferramentas de apoio para o desenvolvimento de programas de computador usando a linguagem C.

Apresentação do LEP-2 e ferramentas de apoio

Esse é nosso primeiro contato com o **Laboratório de Ensino de Programação 2 (LEP-2)**, que iremos usar ao longo do curso de *Introdução à Programação C*.

O roteiro a seguir mostra como criar e executar nosso primeiro programa em C. Siga-o **sem pressa** para entender bem todas as etapas. Precisaremos delas ao longo de todo o curso. **Qualquer dúvida, consulte os monitores ou o professor.**

Primeira etapa: iniciar o computador - Linux

O Linux possui diferentes distribuições, cada uma com sua interface gráfica própria, a qual pode ainda ser alterada pelo usuário. Por isso, teremos que nos adaptar à instalação disponível no LEP-2 e alguns comandos citados abaixo poderão ser diferentes. Siga o **algoritmo** abaixo.

1. Ligue o computador e escolha o sistema operacional Linux.
2. Caso ainda não tenha conta de usuário cadastrada, use para **login** e **senha** a palavra “convidado”.
3. Abra um **Terminal** para digitar comandos (procure nos menus disponíveis essa opção).
4. Ao abrir a janela do terminal, você estará dentro do diretório (ou pasta) raiz do seu usuário (ex., `/home/convidado`). Digite **pwd** para checar essa informação.
5. Crie um diretório para guardar seus programas: **mkdir meusProgramas**
6. Vá para o diretório criado digitando **cd meusProgramas**
7. Verifique quais arquivos o diretório contém, digitando **ls**

Segunda etapa: escrever um programa em C

Para escrever um programa básico em C, siga o algoritmo abaixo.

1. Abra um editor de texto simples qualquer (o **gedit** e o **kate** são algumas opções).
2. Digite (ou copie) o programa abaixo e salve-o dentro do diretório recém criado **/home/convidado/meusProgramas/** dando ao arquivo o nome **aloMundo.c**

```
/*
Descrição:  exhibe mensagem na tela
Entrada:   nenhuma
Saída:     mensagem padrão exibida na tela
Defesa:    não se aplica (não há restrições de entrada)
*/

#include <stdio.h>

int main() {
    printf("\n");
    printf("Alo Mundo!\n");
    printf("\n");
    return 0;
}
```

Terceira etapa: executar um programa C

Para executar o programa **aloMundo.c**, siga o algoritmo abaixo.

1. Volte à janela do Terminal, no diretório **/home/convidado/meusProgramas/** e compile o seu programa digitando:
gcc aloMundo.c -o aloMundo.out -Wall
2. “Compilar o programa” inclui várias etapas, dentre elas verificação de erros sintáticos e geração do arquivo executável. Esse arquivo será chamado **aloMundo.out** (como definimos após a opção **-o**). A opção **-Wall** deve ser **sempre** acrescentada. Nela, ‘W’ se refere a *warnings* e ‘all’ indica que queremos todos eles. Portanto, o compilador exibirá **todas** as mensagens de alertas e erros do seu programa, **caso existam problemas**.
3. Finalmente, **execute** o programa digitando:
./aloMundo.out
4. Os caracteres **./** ao início são necessários para indicar que o programa se encontra no diretório corrente.

5. Verifique a saída do programa na tela.

Se tudo deu certo, você experimentou o seu primeiro programa em C!

Segundo programa em C

Para praticar, repita as duas etapas anteriores para escrever, compilar e executar o programa em C mostrado abaixo.

```
/*
Descrição: dados três números inteiros, retorna o maior
Entrada: os três números inteiros
Saída: o maior número encontrado
Defesa: não se aplica (não há restrições de entrada)
*/

#include<stdio.h>

int main() {
    //dicionário de dados
    int a, b, c;
    int maior;

    //obtem os três números
    printf("Digite os tres numeros separados por espaco: \n");
    scanf("%d %d %d", &a, &b, &c);

    //encontra o maior número
    if(a > b && a > c) {
        maior = a;
    }
    else if(b > c){
        maior = b;
    }
    else {
        maior = c;
    }
    //exibe o resultado
    printf("O maior numero eh: %d \n", maior);

    return 0;
} //FIM
```

Siga as instruções abaixo:

1. Digite (ou copie) o programa e salve o arquivo no diretório **`/home/convidado/meusProgramas/`** dando ao arquivo o nome **`maior.c`**
2. Compile o programa digitando: `gcc maior.c -o maior.out -Wall`
3. Execute o programa digitando: `./maior.out`
4. **Execute o programa várias vezes para diferentes números de entrada.**
5. Agora vamos “provocar” um erro de sintaxe no nosso código. Para isso, altere a linha `int maior;` apagando o `;` (ponto-e-vírgula) no final dessa linha. Salve o seu arquivo e o compile novamente. Verifique o que aconteceu, **tentando entender as mensagens de erro na tela.**
6. Corrija o erro, salve as alterações, compile e execute o programa novamente.

Terceiro programa em C

Escreva você mesmo um **programa em C** que implemente o **algoritmo** a seguir.

Algoritmo: resolver a equação de primeiro grau $ax+b=0$

Entrada: coeficientes reais a e b da equação $ax + b = 0$

Saída: resultado da equação (valor de x)

Defesa: coef. angular não-nulo (i.e. devemos impedir $a == 0$).

INÍCIO

- 1- Obtenha os coeficientes da equação;
- 2- Verifique se o valor do coeficiente ' a ' é inválido. Caso seja, exiba uma mensagem e termine o programa;
- 3- Calcule o valor de x ;
- 4- Exiba o valor de x na tela.

FIM

Quarto programa em C

Escreva um programa em C para calcular a área da coroa circular (anel) formada por dois círculos de raios r_1 e r_2 . Considere $\pi = 3.14$.

Projete primeiro o algoritmo que resolve esse problema e depois escreva esse algoritmo na linguagem C.

Teste o seu programa pelo menos para os seguintes pares de entrada:

- 2 e 1; resposta esperada: 9.42
- 15 e 5; resposta esperada: 628
- 100 e 0; resposta esperada: 31400