

Introdução à programação C/C++ aula prática 0¹

1 Introdução

No Linux não existe uma interface padrão. Você pode escolher, alterar, tirar e muito mais com diversas interfaces diferentes. Cada distribuição traz a sua, mas a escolha é sua.....

Por esta razão vamos ter de nos adaptar a instalação que estamos usando e alguns comandos abaixo não serão exatamente como descritos.

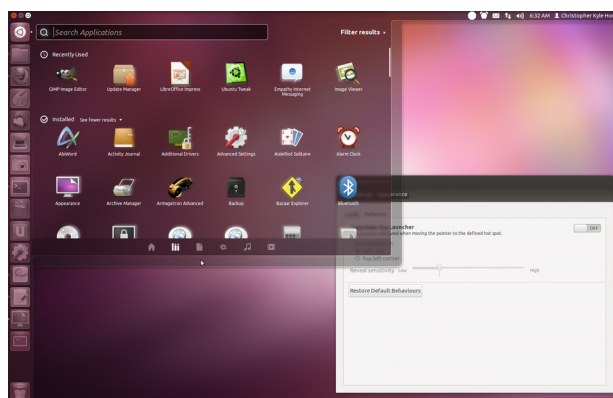


Figura 1: Ubuntu Unity

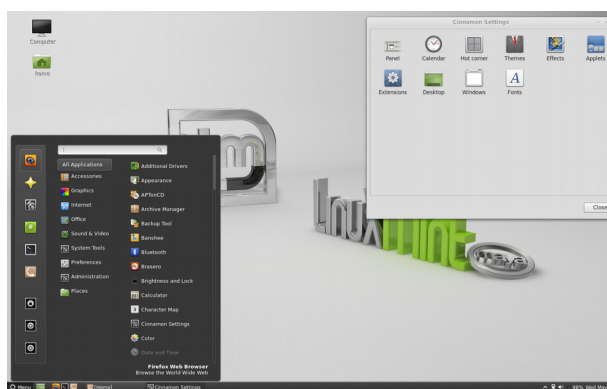


Figura 2: Mint Cinnamon

2 Observações

Sublinhados: os sublinhados no texto são apenas para destaque de informações. Não é preciso sublinhar nada.

Arquivos do usuário: considere que o nome de usuário de sua conta no sistema Linux é, por exemplo, aluno. Normalmente o diretório onde ficam os documentos e dados dos usuários é chamada de /home/aluno. Esse diretório pode variar para contas de usuários em ambientes controlados por servidores de autenticação, ou por outros motivos.

1 Este material é praticamente uma reprodução do produzido pelo Prof. Adriano Joaquim de Oliveira Cruz, disponível em <http://equipe.nce.ufrj.br/adriano/c/praticas.htm>

3 Exercícios

3.1 Exercício 1 – terminal & comandos

Sabemos que vamos usar a maior parte do tempo interfaces gráficas, mas neste exercício é importante usar o terminal.

1. Abra um `terminal` para digitar os comandos. Para isso procure nos menus disponíveis essa opção ou experimente usar a combinação de teclas `Ctrl+Alt+T`.
2. Ao abrir um `terminal` você vai direto para o diretório onde ficam os seus documentos.
3. Para confirmar esta informação digite no *prompt* do terminal o comando `pwd` (guarde esse como um dos comandos da sua vida).



Figura 3: Não tenha medo do terminal

4. Crie um diretório para guardar seus programas. Chame esse diretório de `MeusProgramas`. Para criar o diretório digite o seguinte comando `mkdir MeusProgramas`.
5. Vá para o diretório recém criado. Digite `cd MeusProgramas`.
6. Confirme essas informações usando a interface gráfica. Existem várias formas de realizar essa tarefa. Geralmente, o diretório raiz dos arquivos do usuário é chamado nas interfaces gráficas de `Home` ou `Home Folder`.
7. Usando o terminal, liste os arquivos guardados no diretório através do comando `ls`.

Para o futuro

- * você pode navegar diretamente para o diretório `Home` no terminal digitando o comando `cd ~`.
- * para subir um nível na hierarquia de diretórios use o comando `cd ..`.

3.2 Exercício 2 – primeiro programa

1. Abra qualquer editor de arquivos de texto dos sistema (Word, OpenOffice/LibreOffice Writer **não servem**). A opção mais simples é o gedit. Existem outros disponíveis.
2. Digite o programa da listagem 1, **EXATAMENTE COMO ESTÁ MOSTRADO**.
3. Salve o programa como AloMundo.c no diretório /home/aluno/MeusProgramas do computador. De tempos em tempos procure salvar o seu trabalho para evitar perdas.

Listagem 1: Programa do exercício 2.

```
#include <stdio.h>
int main ( void ) {
    printf("Alo Mundo.\n");
    return 0;
}
```

4. Agora vamos verificar se o seu programa contém algum erro de compilação.
5. Navegue no terminal até o seu diretório MeusProgramas e digite o comando seguinte, exatamente como indicado:

```
gcc -o AloMundo AloMundo.c -Wall
```

Este comando pede que o compilador gcc seja executado. As letras precedidas por sinais de menos indicam opções de compilação. A primeira opção `-o` indica que a palavra logo a seguir é o nome desejado para o código executável do programa, no caso AloMundo. Observe que o não é necessário que o nome termine em .exe.

AloMundo.c indica o nome do programa que deve ser compilado.

A opção `-Wall` serve para pedir ao compilador para indicar não somente erros, mas também todo tipo de avisos de problemas.



Figura 4: Imagem meramente ilustrativa ;)

6. Se você digitou o programa corretamente não haverá nenhum erro e o próximo passo é executá-lo. Para isto digite na janela de comando o nome do executável.

```
./AloMundo
```

O `./` antes de AloMundo indica que o programa está no diretório atual. Em alguns sistemas Linux estes símbolos devem ser usados ou então o programa não é encontrado.

3.3 Exercício 3 – Resolvendo uma equação

1. Digite o programa mostrado na listagem 2. Salve o programa como `primeiro.c`.
2. De tempos em tempos lembre de salvar o seu trabalho para evitar perdas.

Listagem 2: Programa do exercício 3.

```
/*
  Programa:   primeiro.c
  Autor:      Nome do aluno
  Data:       08/09/2016
  Descrição:  Este programa resolve a equação do primeiro
              grau  $ax + b = 0$ 
*/

#include<stdio.h>
int main ( void ){

    float a; // coeficiente da equação de primeiro grau
    float b; // termo independente da equação
    float x; // resultado da equação de primeiro grau

    printf("Este programa resolve uma equacao de primeiro grau. \n");

    printf("Entre como o coeficiente a, por favor. ");
    scanf("%f", &a);

    printf("Entre com o coeficiente b, por favor. ");
    scanf("%f", &b);

    if ( a == 0 ){
        printf("Esta equacao nao tem solução \n");
    } else {
        x = -b/a;
        printf("A solução da equação vale %1.3f\n", x);
    }
    return 0;
}
```

3. Agora vamos verificar se o seu programa contém algum erro de compilação. Digite no terminal comando seguinte.

```
gcc primeiro.c -o primeiro -Wall
```

4. Quantos erros de compilação foram indicados?
5. Executo o programa e teste vários casos.
6. Agora vamos introduzir alguns erros no texto do programa para vermos como o compilador indica erros.
7. Retire o `;` final de um comando `printf`
8. Recompile o programa e observe as mensagens que serão mostradas. Procure corrigi-los baseando-se nas indicações dadas sobre a linha onde ocorre o erro. Observe que a linha indicada é após a linha em que o erro ocorreu.
9. Uma vez corrigido o erro, compile e execute o programa novamente.

3.4 Exercício 4 – Desafio

Escreva um programa que leia duas notas de um aluno, calcule e imprima a sua média. O programa deve imprimir se o aluno foi aprovado ou reprovado. O critério para aprovação é média maior ou igual a 5.0.