{introcomp}

Manual Básico

1 Como compilar seu código?

É necessário compilar seu código para testar seu programa e saber se a lógica utilizada está correta para solucionar todas as possíveis entradas na hora da execução. A compilação de um código no Linux é feita pelo terminal. Para conseguir compilar o seu código, é preciso acessar o local do sistema onde ele se encontra, para isso vamos apresentar os comandos.

2 Comandos de terminal:

No Linux, o terminal é a porta de entrada para o aprendizado em programação de computadores, pois é no terminal onde se compila e executa programas de baixa a alta complexidade de diversas linguagens de programação. No terminal, é possível navegar nos diretórios presentes no sistema, executar comandos do sistema operacional, comandos de programas e comandos do próprio terminal. Desta forma, é fundamental que o programador domine bem os comandos de terminal para que aumente sua produtividade e otimização de tempo.

• cd diretório

Abre um diretório. Por exemplo, para abrir a pasta /mnt, basta digitar "cd /mnt". Para ir ao diretório raiz a partir de qualquer outro, ou seja, o diretório /, digite apenas cd;

• cd ..

Retorna ao diretório superior;

• ls

Lista os arquivos e diretórios da pasta atual;

- ls diretório Lista os arquivos e diretórios do diretório especificado
- rm arquivo Apaga o arquivo especificado;
- rm -R arquivo Apaga o arquivo ou diretório especificado;
- mkdir

Cria um diretório, por exemplo, "mkdir Musicas" cria uma pasta de nome Musicas;

• clear

Elimina todo o conteúdo visível, deixando a linha de comando no topo, como se o terminal acabasse de ter sido acessado;

• diff arquivo1 arquivo2

Indica as diferenças entre os dois arquivos especificados. Este comando é muito útil em programação para verificar se a saída do seu programa está correta.

3 Compilando programas em C com GCC

Para compilar um programa basta executar o seguinte comando no terminal:

• gcc [arquivo.c]

Esse comando irá gerar um arquivo executável chamado a.out. Para executar o arquivo basta utilizar o seguinte comando:

• ./a.out

Você pode especificar o nome do se arquivo executável usando a opção -o da seguinte forma:

• gcc [nomeDoArquivo.c] -o [nomeDoArquivoExecutável] E para executar:

• ./[nomeDoArquivoExecutável]

Caso exista algum **erro** o compilador indicará e o arquivo executável **NÃO** será gerado. Caso exista algum **warning (aviso)** o compilador irá indicar, porém o arquivo executável será gerado. Warnings servem para indicar ou sugerir que você fez algo no seu código que possivelmente pode ser um erro.

4 Algumas opções úteis

A flag -Wall trata warnings como erros e exibe alguns warnings que normalmente seriam ignorados.

O compilador da plataforma run.codes (falaremos dela mais pra frente) utiliza a flag -Wall. Caso seu codigo esteja funcionando mas não esteja sendo aceito pelo run.codes tente roda-lo utilizando essa flag. Exemplo de utilização:

$gcc \ [nomeDoArquivo.c] \ -o \ [nomeDoArquivoExecutável] \ -Wall$

5 GCC no Windows

Para compilar seus códigos no windows utilizaremos o Dev C++, que é um ambiente de programação (IDE - sigla em inglês que significa Ambiente Integrado de Desenvolvimento). Ele torna a atividade de programar mais simples.

Figura 1: Diretorio do Dev-C++



Indicamos criar um diretório para armazenar todos os códigos que serão criados no Dev C++ assim como a figura 1. Com o diretório criado, vamos criar um projeto no Dev-C++. Clique no botão para criar um novo projeto, assim como na figura 2.

Figura 2:	Menu inicial	do Dev-C++	e início d	la criação	do Projeto
E Dev-C+	++ 5.11				

Arqu	oviu	Editar	Localizar	Exibir	Projeto	Executar	Ferramentas
	5				h +		
ÎΩ	Arq	uivo Fon	te Ctrl+	N	14		
-100	Proj	jeto					
	Mo	delo					
	Clas	sse					
_							

Selecione o tipo de aplicação a ser desenvolvida, Console Application, a linguagem de programação, Projeto C, e o nome do projeto. O menu deverá estar parecido com o exposto na figura 3.

Dasic N	lultimedia Win	32 Console				
Windows Application	Console Application	Static Library	dil.	Empty Proj	ect	
Acons	le application (I	MS-DOS window)	۲	Projeto C	O Projeto C+	+
Nome:				Linguagem	Faulao	

Figura 3: Opções de novo Projeto

Finalmente, escolha o diretório criado para ser onde os arquivos do projeto serão armazenados. Você poderá organizar os arquivos como quiser. Sugerimos esta como uma forma padrão.

🐨 Salvar como					\times
Salvar em:	Codigos DEV	/C++ ~	G 🤌 📂 🛄 -		
Acesso rápido	Nome	^ Nenhum item corr	Data de modificaç esponde à pesquisa.	Тіро	Tan
Área de Trabalho					
Bibliotecas					
Este Computador					
	<				>
I	Nome:	Projeto 1 dev		~	Salvar
Dede					0.1

o arquivo do projeto na posta griad D: 4. Calanda

O Dev-C++ preenche já um código padrão para você testar o DEV-C++ e começar a

programar. Seu trabalho agora é programar o seu código.



Quando você terminar seu código, você deve compila-lo e depois rodá-lo. Você pode fazer isso de uma vez automaticamente ou fazer as duas etapas separadamente. Utilizaremos um programa "Hello World" simples para exemplificar.

	Figura 6: Código "Hello World"
] 🔶	* G. G. 2 + 4 4 2 8 🗆 🗉 🕬 🛛
	~
[*] main.c	
1 #	include <stdio.h></stdio.h>
2 #	include <stdlib.h></stdlib.h>
3	
4 /	imes run this program using the consol
5	nt main(int area obar yarou[1) (
	nt main(int argc, char ×argo[]) (
ſ	printf("Hello World!\n");
8	return 0;
9 L }	

Basta clicar nos botões que estão localizados na parte superior direita da figura 6. Temos quatro botões que têm as seguintes funções:



Figura 7: Botões de execução do Dev-C++

Em alguns casos, ao terminar de executar o programa, o Dev-C++ pode fechar o terminal onde seu programa está rodando e assim atrapalhar a visualização da saída de alguma forma. Caso seja necessário resolver isso, inclua a linha "system("pause");" acima do "return 0;". Com essa alteração nosso código "Hello World" ficará como na figura 8:

Figura 8: Código "Hello World" com o comando "system("pause");"@

1	<pre>#include <stdio.h></stdio.h></pre>
2	<pre>#include <stdlib.h></stdlib.h></pre>
3	
-4	<pre>/* run this program using the console</pre>
5	
6	int main(int argc, char ×argv[]) {
7	<pre>printf("Hello World!\n");</pre>
8	system("pause");
9	return 0;
10	L }

Observação: não coloque o comando system("pause"); no linux.

- Podem ser formados por letras, números e o caractere sublinhado;
- Os números não podem aparecer na primeira posição;

Exemplos: cor, n1, n2, roda_1,roda_2.

Uma outra opção no Windows é a compilação online. Existem sites como http://ideone.com cujo você escolhe a linguagem e pode escrever o código, compilar e até testar com entradas.

6 AVALIAÇÃO

Para a sua aprovação no Introcomp, é necessário manter uma presença de 75'%' nas aulas e uma média final de 7 pontos.

Além disso, se o aluno tiver acima de duas faltas seguidas ele será considerado desistente. As faltas podem ser justificadas com antecedência, então sempre entre em contato com a nossa equipe em caso de necessidades.

A nota é calculada da seguinte forma:

- Workings(W): as notas obtidas nos workings representam 60'%' da nota final do curso.
- Prova(P): haverá uma prova no final curso, valendo 20'%' da nota final.
- Trabalho(T): é proposto para os alunos um trabalho valendo também 20'%' da nota final que aborda todo o conteúdo do curso.

A nota de cada working é feita calculando a média aritmética de cada praticando, onde a nota máxima de cada working é 10. No final, é realizada também uma média com a nota de cada working, totalizando no máximo 10 pontos.

Média = 0.6*W+0.2*T+0.2*P

7 Referências:

http://fig.if.usp.br/~esdobay/c/gcc.html