

Ola

Iniciando

A IDE CS50 é uma **Interface Integrada de Desenvolvimento** (do inglês, *Integrated Development Environment*, IDE) que permite você programar "na nuvem" sem instalar qualquer software localmente. De fato, a IDE CS50 provê seu próprio **ambiente de trabalho** (do inglês, *workspace*) onde você pode salvar arquivos e pastas (diretórios). Além disso, cada vez que você salva seu arquivo, um ponto de restauração é criado e você pode acompanhar as mudanças feitas.

Logging In

Acesse <https://ide.cs50.io> e clique em **"Sign in with GitHub"** para acessar sua IDE CS50. Uma vez que sua IDE tenha sido iniciada, você verá que (por padrão) ela é dividida em três partes. Na parte superior está o "editor de textos" onde você escreverá seus programas. Na parte de baixo você tem um "terminal de comandos" ou CLI onde você poderá listar o conteúdo dos diretórios, compilar e executar programas, e eventualmente instalar novos programas. Na lateral esquerda você encontra o "navegador de arquivos", que mostra todas as pastas e arquivos na sua IDE.

Inicie clicando no terminal. E você verá o *prompt* abaixo

```
~/ $
```

Em seguida digite

```
mkdir ~/pset1
```

e tecla ENTER. Será criado um diretório chamado **pset1** dentro do seu diretório pessoal.

Entre no diretório com o comando abaixo

```
cd ~/pset1
```

e o *prompt* mudará para

```
~/pset1/ $
```

Caso contrário, reveja seus passos para descobrir o que houve de errado.

Crie um novo diretório chamado **hello** e entre no diretório

```
mkdir hello  
cd ~/pset1/hello
```

e o *prompt* mudará para

```
~/pset1/hello/ $
```

Agora, crie um arquivo chamado **hello.c** com o comando

```
~/pset1/hello/ $ touch hello.c
```

em seguida abra o arquivo no editor de textos e escreva as linhas abaixo:

```
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello, world\n");  
}
```

Perceba que a IDE CS50 irá destacar (colorindo) palavras-chave da linguagem C. As cores não fazem parte da linguagem C e servem apenas para que você identifique mais rapidamente palavras com funções diferentes dentro do seu código.

Para ver a lista de arquivos, digite **ls** e tecla ENTER no seu terminal de comandos.

Compilando seu código

Depois de ter digitado seu código, use o comando

```
clang hello.c
```

ou o comando

```
gcc hello.c
```

para compilar seu código. Será criado um arquivo chamado **a.out**. Para executar o arquivo digite

```
./a.out
```

no seu terminal e tecle ENTER. Deverá aparecer a mensagem

```
Hello, world
```

no terminal.

Nomeando programas

Toda vez que você usa os comandos `clang` ou `gcc`, o nome do arquivo criado será sempre `a.out`. Para dar um nome diferente para seu arquivo, utilize os comandos como segue

```
clang -o hello hello.c
```

ou

```
gcc -o hello hello.c
```

Nos dois casos, o nome do programa será `hello` em vez de `a.out` e você poderá executar o programa digitando

```
./hello
```

no terminal e teclando ENTER.

Para facilitar a compilação, a IDE CS50 dispõe do comando `make`. Assim, basta digitar

```
make hello
```

e será criado o arquivo `hello`. Note que o nome depois do comando `make` é apenas `hello` sem a extensão `.c` e que o arquivo `hello.c` precisa existir no diretório onde `make` está sendo executado.

Obtendo dados do usuário

Em breve, usaremos a função `scanf` que serve para solicitar dados ao usuário via teclado. Mas por ora evitaremos as complexidades desta função e, principalmente, a complexidade de tratamentos de erros. Por exemplo, quando solicitamos para o usuário digitar um número e ele digita uma letra.

Para tanto, utilizaremos as funções

```
get_int
get_char
get_float
get_double
get_string
```

Para que seja possível utilizar tais funções, devemos incluir a biblioteca `cs50.h` em nossos programas.

Assim, podemos solicitar um inteiro e armazenar em uma variável com a seguinte linha

```
int n = get_int("Digite um numero: ");
```

ou uma `string` (lembre-se que em C não existem *strings*, mas voltaremos a este assunto posteriormente) usando

```
string nome = get_string("Nome: ");
```

E podemos utilizar os marcadores de **lugares reservados** para exibir o conteúdo de uma variável por meio da função `printf` como em

```
printf("O valor digitado foi %d\n", n);
```

ou

```
printf("Ola, %s\n", nome);
```

É possível que você obtenha erros como em

```
error: use of undeclared identifier 'string'; did you mean 'stdin'?
```

Neste caso, verifique se você incluiu a biblioteca `cs50.h` no início do seu programa.

Como testar seu código

Execute o comando `check50` como abaixo

```
check50 cs50/problems/2020/x/hello
```

Para verificar se seu código está devidamente **estilizado**, utilize o comando `style50`

```
style50 hello.c
```

Note que devemos informar o nome do arquivo de código-fonte.

Enviando seu programa

Uma vez que você tenha verificado o funcionamento, `check50` e a estilização do código, `style50`, execute o comando abaixo, logando com sua conta do GitHub através de **nome de usuário** e **senha**. Por questões de segurança, você verá asteriscos (*) em vez dos caracteres da sua senha.

```
submit50 cs50/problems/2020/x/hello
```