

PROGRAMAÇÃO SHELL SCRIPT: O BÁSICO PARA INICIAR E CRIAR SCRIPTS NO BASH DO LINUX

por [pedrodelfino](#) • 0 comments



Programação Shell Script é útil para quem trabalha com sistemas Linux? Seu foco é automatizar tarefas? Então veja aqui o básico para você iniciar do zero.

O sistema Linux apresenta as melhores soluções para a área de TI, descomplicando a vida dos administradores de sistemas. E, por esse motivo, ele vem ganhando a preferência em relação a outros sistemas no mundo corporativo.

Porém, alguns [profissionais de TI](#) desconhecem as vantagens do sistema, enquanto outros ainda têm dúvidas sobre as soluções que ele oferece.



O fato é que o sistema Linux possui ferramentas que otimizam os serviços de TI de uma empresa, pois o programa é constantemente aprimorado por especialistas em desenvolvimento de software. Assim, o profissional de TI tem em suas mãos um sistema que ajudará na [boa gestão da área de TI](#).

Neste artigo, vamos dar um exemplo da [eficiência e praticidade do sistema Linux](#): nós ensinaremos como fazer a programação shell script no software por meio do bash. Você com certeza ficará impressionado com a facilidade oferecida pelo programa. Acompanhe o tutorial!

PROGRAMAÇÃO SHELL SCRIPT: INTRODUÇÃO

Utilizando o interpretador de comandos bash ou shell, podemos [criar códigos ou scripts](#) no sistema Linux. Esses códigos criados facilitam muito a vida do programador ao economizar tempo, pois permitem automatizar tarefas de rotina e também tornam mais simples ações como: abrir programas e esvaziar diretórios temporários e lixeira, entre outras tarefas.

Antes de iniciar a programação, verifique se entrou no programa como usuário comum ou root. Se estiver no modo root, mude para usuário comum. Por que a mudança? Como root qualquer ação descuidada do programador será aceita pelo programa e poderá causar danos ao sistema operacional.

Para fazer essa verificação, basta ao abrir o terminal e verificar o qual é o símbolo ao lado do cursor. O símbolo (\$) significa usuário comum, enquanto o símbolo (#) é o modo root. Para mudar para o usuário comum, digite exit.

CRIAR UM ARQUIVO

Em um primeiro momento, é preciso criar um arquivo para digitar o script. Há dois modos de realizar essa ação: via modo gráfico ou via terminal. No primeiro caso, você deve clicar com o botão direito do mouse no diretório que desejar e, em seguida, pode escolher entre criar um novo documento ou arquivo de texto.

No modo via terminal, utilize o comando vi. Abra o arquivo e digite:

```
vi exemplo1.sh,
```

Será criado e aberto um arquivo de leitura e escrita. Outra opção é digitar o comando touch:

```
touch exemplo1.sh,
```

Essa ação cria um arquivo sem abri-lo. Se desejar abri-lo, você poderá usar o editor que preferir.

PERMISSÃO AO ARQUIVO

Para começar a editar o arquivo, é preciso conceder a permissão de escrita a ele. Se você digitou o comando vi, pressione ESC para que apareça o caractere ":". Então digite:

```
!chmod 777 %
```

Caso tenha optado pelo comando touch, escreva:

```
chmod 777 exemplo1.sh
```

Vamos entender o que foi feito: o comando `chmod` é usado para conceder permissões em diretórios e arquivos, enquanto o valor `777` permite que o usuário total liberdade para editar o arquivo. O caractere `!` obriga o comando `vi` a executar o `chmod`, enquanto o símbolo `%` refere-se ao arquivo atual.

EDITAR E EXECUTAR O ARQUIVO

Depois da permissão do arquivo, iniciaremos a parte de edição do arquivo e usaremos o comando `vi` para essa tarefa. Abra o arquivo e digite:

```
vi exemplo1.sh
```

É necessário digitar “`i`” para colocar o `vi` no modo inserção.

Nesse ponto, você precisa definir qual Shell será usado — vamos escolher o `bash`. Porém, você pode escolher outro interpretador de comandos, como `sh`, `ksh` ou `csh`. Para usar o `bash`, digite:

```
#!/bin/bash
```

INSERIR COMENTÁRIOS

Esse item é fundamental na programação do sistema, porque, ao inserir comentários que expliquem a função do código e que ações estão sendo feitas no programa, o programador não perde a linha de execuções.

Inicie com o caractere “`#`” para inserir o comentário, como mostra o exemplo abaixo:

```
#!/bin/bash
```

```
#comentários
```

USO DAS VARIÁVEIS

Deixar de declarar as variáveis pode prejudicar o funcionamento dos comandos, visto que são elas que armazenam informações importantes que serão utilizadas pelos códigos. Por isso essa ação é tão importante na linguagem da programação.

Para declarar a variável, siga o exemplo:

```
nome_da_variavel=valor
```

Vale lembrar que o valor pode ser strings ou números. Para utilizar o valor, devemos colocar o símbolo `$` na frente do nome que será inserido.

ENTRADA DE DADOS DO USUÁRIO

Se o script pedir ao usuário o fornecimento de dados para a entrada do processamento, o programador precisará digitar um comando para que o ele leia os dados escritos. Nesse caso, digite:

```
read nome_da_variavel_para_armazenar_o_valor_a_ser_lido
```

Validação da execução dos comandos

Um script precisa validar suas tarefas para cumpri-las com eficiência. Para isso, acompanhe sempre os logs, para que possa perceber erros ou melhorias. Existem recursos de monitoramento que podem ser utilizados caso você deseje.

COMANDO DE SELEÇÃO

Um comando simples que permite a execução de uma tarefa baseada na ação de um usuário que esteja usando o seu sistema é o condicional. Faça o seguinte:

```
if [ CONDICAO ] ;  
then  
AÇÕES  
fi
```

É importante lembrar que para cada “if” aberto, deve haver um “fi” que feche a sequência — a mesma regra serve para os colchetes.

FUNÇÕES

Para organizar, separar e estruturar um algoritmo, é necessário o uso de funções — não importa a linguagem de programação utilizada. Veja o que você pode fazer:

```
nome-funcao()  
{  
  AÇÕES  
}
```

INSERÇÃO DE ARGUMENTOS

O shell script recebe dados fornecidos por outro programa ou por um usuário com a finalidade de produzir saídas — esses dados são os argumentos. Veja alguns dos nomes que qualificam os argumentos:

- › \$# — total de argumentos que foram passados
- › \$* — retorno aos argumentos
- › \$0 — nome do script executado

```
#!/bin/bash  
echo Foram digitados $# parâmetros. São eles: $*.  
echo O primeiro parâmetro foi: $1  
echo O nome do script é: $0
```

COMO CONTINUAR AGORA NA PROGRAMAÇÃO SHELL SCRIPT?

Como você pôde notar, o sistema Linux é completo, e muitos programadores decidem fazer um [curso de especialização no sistema](#) para aproveitar bem todas as suas ferramentas e funções. O profissional de TI aprende como usar os diversos comandos, detalhes técnicos e como montar o sistema de acordo com as necessidades de sua empresa.

Dessa forma, o [administrador de sistemas torna-se um profissional capacitado](#), pois terá a melhor solução de TI para seu negócio — o sistema Linux.

Este artigo ajudou você a entender como usar a programação shell script no bash de seu Linux? Deixe seu comentário!