Introdução
Shell Script

João Pedro de França Lourenço
Thiago Bruchmann Carnaiba

Objetivo do curso

Apresentar as utilizações mais importantes do shell script;

Fixar a importância de ter conhecimento sobre shell e Linux para o mercado de trabalho;

Aprender a criar scripts de automatização.

Módulos abordados

01

Shell e linha de comando

O que é o shell?

02

Conceitos básicos

O que é um script?

03

Variáveis e entradas

Como funcionam as variáveis?

04

Estruturas de Decisão

If-else, switch-case...

Módulos abordados

05

Estruturas de Repetição

for, while, until, select, break...

06

Funções e Boas Práticas

Criando funções com argumentos

07

Exemplos Práticos

Exemplos e exercícios

Introdução

Por que aprender shell script?

Eu preciso ser um usuário avançado?

Unix? BSD? Servidores?

01

Shell e linha de comando

Vamos entender mais sobre o mundo unix

Shell

O shell é uma interface de linha de comando (CLI) que permite aos usuários interagirem com um sistema operacional.

É uma camada de software que atua como intermediário entre o usuário e o kernel do sistema operacional.

Shell

Bash

Bourne-Again Shell

Zsh

Z Shell

Tcsh

Tenex C Shell

Ksh

Korn Shell

Os shells fornecem um ambiente de linha de comando para a execução de comandos e scripts, automatizando tarefas e realizando operações no sistema.

Diferença entre Shell e Terminal

Shell

O "shell" é o interpretador de comandos, é o componente que processa os comandos, executa programas e gerencia processos.

Terminal

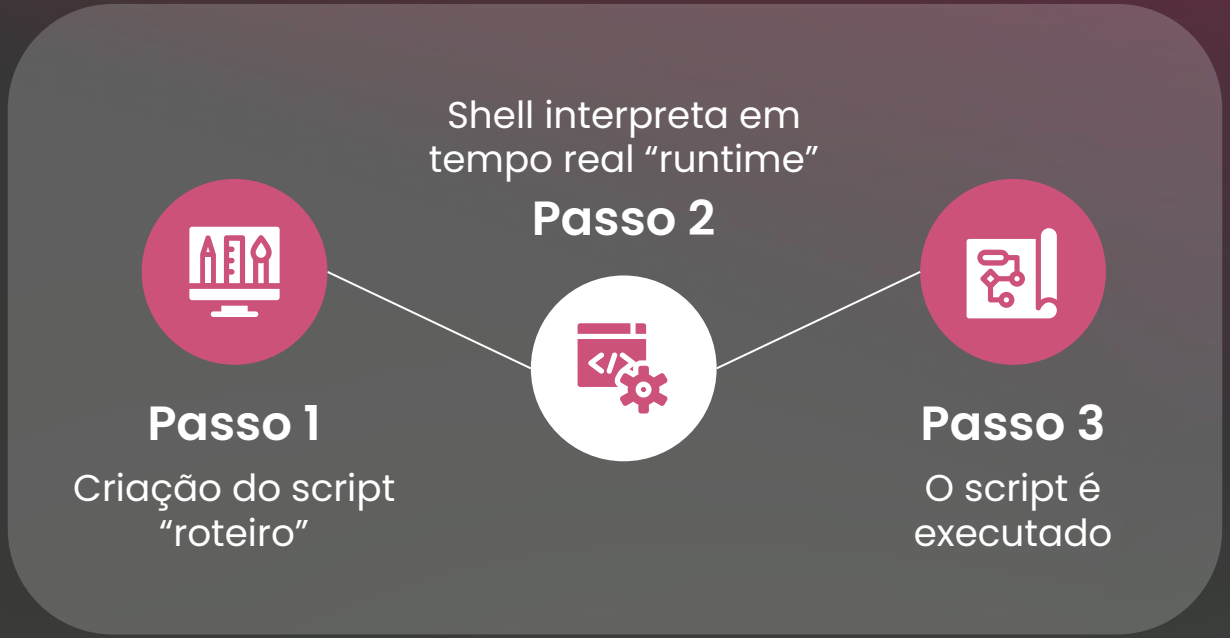
O "terminal" é a janela ou programa que permite ao usuário interagir por uma interface gráfica ou de texto com o shell.

Linguagem interpretada

Shell

PHP

Ruby

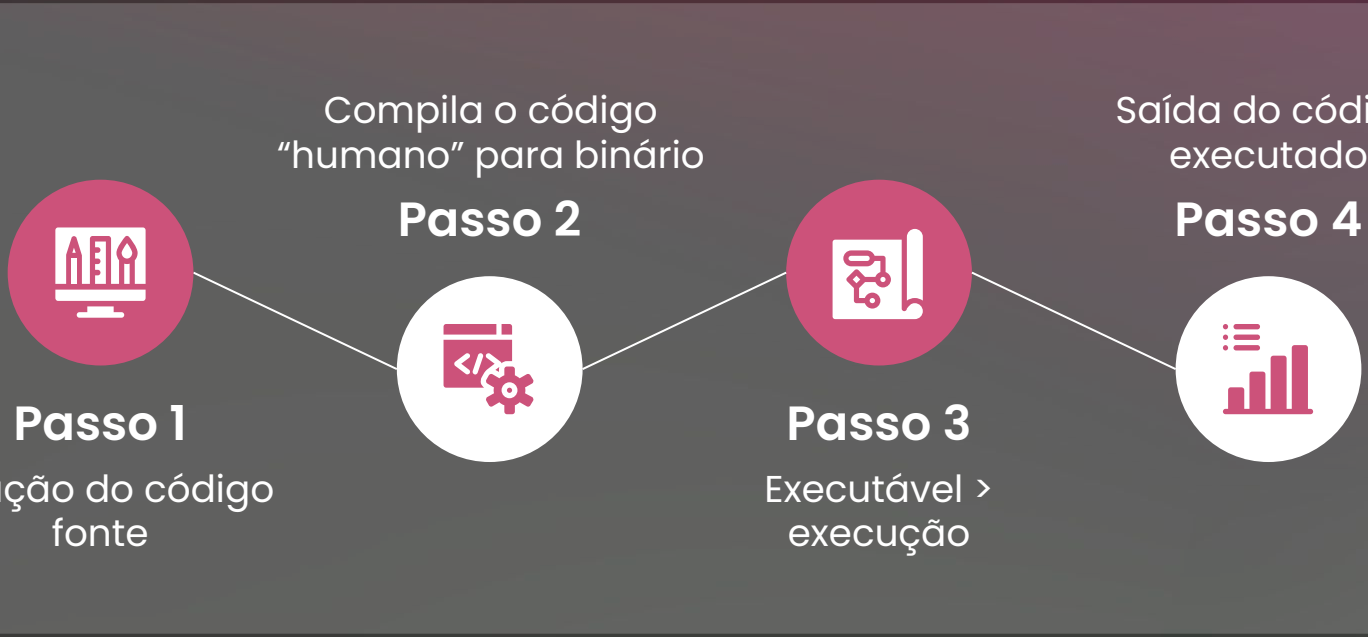


Perl

Python

Javascript

Linguagem compilada



Comandos básicos no shell

sudo	cd	pwd
executa um comando como superusuário	permite navegar até determinado diretório	exibe o caminho completo do diretório atual
Exemplo	**Exemplo**	**Exemplo**
sudo apt-get install	cd /home/imagens	/home/imagens

Comandos básicos no shell

rm	cp	mv
remove arquivos ou diretórios no diretório atual	copia arquivos ou diretórios do diretório atual	move ou renomeia arquivos ou diretórios
Exemplo	**Exemplo**	**Exemplo**
rm img	cp img /home/downloads/	mv img img2

Comandos básicos no shell

ls	mkdir	touch
lista arquivos e diretórios no diretório atual	cria um novo diretório	cria um novo arquivo
Exemplo	**Exemplo**	**Exemplo**
	mkdir /home/teste	touch teste.py

Comandos básicos no shell

cat	find	chmod
exibe o conteúdo de um arquivo de texto	busca por arquivos em diretórios	altera as permissões de arquivos e diretórios
Exemplo	**Exemplo**	**Exemplo**
cat teste.py	find teste.py	chmod +x teste.py

02

Conceitos básicos

Script, .sh, permissões e execuções

Script

Um script é um arquivo de texto que contém uma sequência de comandos que podem ser executados em ordem.

No contexto de Shell Script, esses comandos são instruções que podem ser interpretadas e executadas pelo shell.

Scripts são usados para automatizar tarefas repetitivas, executar operações complexas ou simplificar a execução de comandos em lotes.

Shell scripts geralmente possuem extensões de arquivo como .sh (para Bash) e podem ser executados como programas.

A shebang, também conhecida como "hashbang" ou "inter pontuação", é uma sequência de caracteres que aparece no início de um arquivo de texto executável.

#!/caminho/para/interpretador

A shebang é uma parte crítica de um script executável porque diz ao sistema operacional qual shell ou interpretador deve ser usado para interpretar o código contido no arquivo.

#!/bin/bash: Bash.
#!/bin/sh: shell padrão.
#!/usr/bin/python3: Python 3.

Permissões de execução

Para executar um script, é necessário atribuir permissões de execução ao arquivo. Isso é feito usando o comando chmod para modificar as permissões do arquivo.

As permissões de execução são representadas por "x" nos bits de permissão do arquivo. Para adicionar permissão de execução a um arquivo, use o comando chmod +x arquivo.sh.

Sem as permissões de execução, o sistema não permitirá que o script seja executado.

Escolhendo o Shell

Diferentes shells podem ser usados para executar scripts, e a escolha do shell depende das necessidades do script e da compatibilidade.

O Bash (Bourne-Again Shell) é um shell comumente usado para scripts no ambiente Unix/Linux e é amplamente compatível com a maioria dos sistemas Unix.

O "sh" é um shell genérico que pode variar de sistema para sistema, mas muitas vezes é um link simbólico para o shell padrão do sistema. No entanto, é mais seguro especificar o shell exato no cabeçalho do script, como #!/bin/bash, para garantir a consistência.

Escrevendo um script simples

Para escrever um script, crie um arquivo de texto usando um editor de texto, como o nano, vim ou gedit (gnome-text-editor). Lembre-se de adicionar a linha de cabeçalho indicando qual shell deve ser usado (no nosso caso, #!/bin/bash).

Em seguida, insira os comandos que deseja executar no script. Certifique-se de que cada comando esteja em uma nova linha.

Salve o arquivo com a extensão correta (no nosso caso, .sh) e atribua permissões de execução com chmod +x arquivo.sh.

Execute o script usando ./arquivo.sh no terminal. Certifique-se de estar no diretório em que o arquivo está localizado.

Vamos criar nosso primeiro script

03

Variáveis e entradas

Bora praticar com variáveis, entradas e saídas

Como funcionam

01

Declarando

```
minha_variavel="Olá, mundo!"
```

02

Read

```
read nome
```

03

Echo

```
echo "Qual é o seu nome?"
echo "Olá, $nome!"
```

04

Concatenando

```
string_completa="$a $b"
```

Declarando variáveis

Em shell script, as variáveis são usadas para armazenar informações e valores. Para declarar uma variável, basta atribuir um valor a ela. O nome da variável não pode começar com um número e não deve conter espaços em branco. Os valores das variáveis são tratados como strings por padrão, não sendo necessário especificar o tipo de dado. Aqui está um exemplo:

```
minha_variavel="Olá, mundo!"
```

Neste caso, minha_variavel é o nome da variável e "Olá, mundo!" é o valor atribuído a ela. Você pode acessar o valor da variável usando o $ seguido do nome da variável, por exemplo, $minha_variavel.

Entrada do usuário

O comando read é usado para obter entradas do usuário no shell. Ele permite que você solicite que o usuário insira dados, que serão armazenados em uma variável. Veja um exemplo de como usar o read:

```
echo "Qual é o seu nome?"
read nome
echo "Olá, $nome! Bem-vindo ao shell script."
```

Exibindo a saída

O comando echo é usado para imprimir mensagens na tela. Ele é muito útil para exibir informações ou resultados de um script. Aqui está um exemplo de uso do echo:

```
minha_variavel="Olá, mundo!"
echo $minha_variavel
```

Concatenando variáveis

A concatenação de variáveis é usada para combinar o conteúdo de duas ou mais variáveis em uma única string. Em shell script, você pode atribuir variáveis já existentes para unir variáveis ou strings. Aqui está um exemplo:

```
primeiro_nome="Thiago"
sobrenome="Bruchmann"
nome_completo="$primeiro_nome $sobrenome"
echo "Nome completo: $nome_completo"
```

04

Estruturas de decisão

if then else elif fi, case esac

Como funcionam

01

if then fi

```
if [ condição ]; then
# Seu codigo
fi
```

02

if then else fi

```
if [ condição ]; then
# Seu codigo
else
# Seu codigo
fi
```

03

if then elif fi

```
if [ condição1 ]; then
# Seu codigo
elif [ condição2 ]; then
# Seu codigo
fi
```

04

case esac

```
case expressao in
        padrao1)
                ;;
        )*
esac
```

05

Estruturas de repetição

For, while, until.

Como funcionam

01

For

```
for [ condição ];
do
#Seu codigo
done
```

02

For (estilo C)

```
for ((i=0; i<=5; i++))
do
echo "Executando o $i"
done
```

03

While

```
while [ condição ];
do
#Seu codigo
done
```

04

case esac

```
until [ condição ];
do
#Seu codigo
done
```

06

Funções e Boas Práticas

Criando funções em shell com passagem de argumentos

Como funcionam

01

Função

```
funcao_aqui(){
   comandos;
}
função aqui #chamando
```

02

Argumentos

```
minhaFuncao(){
  arg1=$1
  arg2=$2
  comando no $arg1
}
```

07

Exemplos Práticos

Exemplos de códigos e exercícios práticos

Acesse o GitHub

https://github.com/bThiiago/Shell-Script

Obrigado!

Você tem alguma dúvida?

joao.franca@aluno.ifsp.edu.br
thiago.bruchmann@aluno.ifsp.edu.br

https://www.linkedin.com/in/joao-franca1/
https://www.linkedin.com/in/thiagobruchmann/