

Programação Shell Script

REDES DE COMPUTADORES

Prof. Dr. João Paulo Aramuni

Scripts disponíveis no ambiente

- * Antes de escrevermos novos scripts veremos os scripts que nós já temos **disponíveis** para nossa utilização.

Scripts disponíveis no ambiente

- * Esses scripts foram colocados dentro do seu **diretório de usuário** quando a sua conta foi criada.
- * Eles são utilizados para **configurar** o comportamento das suas **sessões** (quando você realiza o login) no computador.

Scripts disponíveis no ambiente

- * Você pode *editar* esses scripts para alterar alguma coisa de acordo com a sua necessidade.
- * Vamos dar uma olhada em alguns desses scripts mais tarde e aprender novos conceitos importantes sobre o shell.

Scripts disponíveis no ambiente

- * Antes de começar a editar esses scripts vamos relembrar alguns conceitos.

Comandos

- * O que são **comandos** em Linux?

Comandos

- * Comandos podem ser diversas coisas.
- * Alguns comandos **já estão embutidos** no próprio shell.
- * Isso quer dizer que **o shell consegue entender esses comandos** que já estão embutidos nele.
- * **Ex:**

`$ pwd`

`$ cd`

Comandos

- * Como são chamados esses comandos **embutidos** no shell?

Comandos

- * Comandos Builtins – Embutidos no Shell
- * Os comandos que já foram implementados (escritos) dentro do próprio shell são chamados de **builtins**.

Comandos

- * Comandos Builtins – Embutidos no Shell
- * Como **visualizar** estes comandos (builtins) ?

Comandos

- * Comandos Builtins – Embutidos no Shell
- * Para visualizar os comandos que já estão *embutidos no próprio shell* digite o comando abaixo:

\$ help

Comandos

- * Este é um tipo de comando.
- * E os outros tipos?
- * Quais são os demais tipos de comandos em Linux?

Programas Executáveis

- * O **segundo tipo** de comando são os **programas executáveis**.
- * A **maioria** dos **comandos no Linux** fazem parte deste tipo.
- * Os programas executáveis são justamente todos aqueles arquivos **disponíveis** dentro dos **diretórios** que estão incluídos no seu **path** de usuário.

Programas Executáveis

- * Vamos dar uma olhada na variável **path** do nosso usuário para entendermos melhor:

```
$ echo $PATH
```

Tipos de Comando

- * Os demais tipos de comandos Linux...

Tipos de Comando

- * Os **dois últimos** tipos de comandos são aqueles contidos no “runtime environment” (**Ambiente de Execução**) do usuário.
- * Os dois tipos restantes de comandos no ambiente são os **alias** e as **funções shell**.
- * Durante a sua sessão, o sistema mantém um **grande número de informações** na **memória** a seu respeito.

Ambiente do Usuário

- * Esse **grande número de informações** a respeito do usuário é chamado de **ambiente**.
- * O ambiente contém algumas coisas como seu **path**, seu **nome de usuário**, o nome do arquivo onde **seus e-mails** são entregues e muitas **outras informações**.

Ambiente do Usuário

- * Você poderá ver a **lista completa de informações** sobre seu **ambiente** utilizando os comandos abaixo:

\$ env

set

Aliases

- * Antes de saber o que significa um alias, **vamos fazer um!**
- * Assegure-se de estar no **diretório home** do seu usuário.
- * Abra o arquivo **.bash_profile** e verifique se ele utiliza o arquivo **.bashrc**.

Aliases

- * Se o seu **.bash_profile** não realiza uma chamada ao arquivo **.bashrc**
 - * Coloque no final do arquivo o seu alias
- * No RedHat o arquivo **.bash_profile** utiliza o arquivo **.bashrc** para armazenar os aliases e as funções shell
 - * Abra o arquivo **.bashrc** e adicione o alias no local indicado no arquivo.

```
alias meu='ls -la'
```

O .bash_profile

- * O **.bash_profile** é um **shell script**
 - * Ele é executado cada vez que você realiza o seu login
- * Quando **adicionamos um alias** para um comando neste arquivo, estamos criando um **novo comando** chamado “**meu**” que realiza a operação “**ls -la**”.

alias **meu** = ‘**ls -la**’

O .bash_profile

- * Para experimentar o nosso **novo comando** precisamos realizar o **logout/login** na máquina.

Aliases

- * Então para que utilizamos essa **técnica**? (Aliases)

Aliases

- * Utilizando esta técnica você pode criar inúmeros **comandos personalizados** para o seu ambiente.
- * Exercício << 10 min >>
 - * Crie um alias para o comando “date” para exibir a data na tela no seguinte formato:
 - * Dia da semana por extenso, dia mês, Ano

Aliases

- * Exercício

- * **Crie um alias** para o comando “date” para exibir a data na tela no seguinte formato:
 - * Dia da semana por extenso, dia mês, Ano.

Resp:

\$ **alias** hoje =‘date “+%A, %d %B, Ano de %Y” ’

Aliases

- * Obs:
- * 1) O comando alias é um comando **shell builtin**
- * 2) Você pode criar os seus alias **diretamente** no **prompt de comando**.
- * 3) Tome cuidado! Os comandos criados diretamente no prompt de comandos **somente serão válidos** durante aquela sessão do prompt.

Funções Shell

- * Aliases são bons para comandos bem simples.
- * Para criar estruturas mais complexas, nós devemos utilizar as **funções shell**.
- * As funções shell podem ser definidas como pequenos **sub-scripts**.

Funções Shell

- * Exemplo:
- * Abra o arquivo onde você criou o alias utilizando o comando `date` e o substitua pela função shell “hoje”:
- * **function hoje {
 echo “A data de hoje é: ”
 date “%A%, %d %B, Ano de %Y”
}**

Funções Shell

- * O comando **function** também é um comando builtin, ou seja, pode ser executado **diretamente no prompt** de comando:

```
[aluno@aluno alunox]$ function hoje {  
> echo "A data de hoje é:"  
> date "+%A, %d %B, Ano de %Y"  
> }  
[aluno@aluno alunox]$
```

Comando Type

- * Existem muitos tipos de comandos
- * Isso pode gerar inúmeras confusões e você pode não saber se está utilizando um alias, uma função shell ou um comando executável.
- * Para saber que tipo de comando você está utilizando, utilize o comando **type** conforme abaixo:

\$ type <nome_comando>

Exercício

- * O comando Grep é um comando frequentemente utilizado no shell.
- * Com base na tabela a seguir, pesquise a descrição sobre o que cada uma dessas opções do Grep fornecem.

grep

Procura em arquivos ou textos por linhas que contêm determinado padrão de pesquisa. O padrão pode ser uma string ou uma expressão regular.

Opções:

Opção	Lembrete	Descrição
-i	Ignore case	
-v	Invert	
-r	Recursive	
-q	Quiet	
-s	Silent	
-n	Number	
-c	Count	
-l	Filename	
-w	Word	
-x	Full line	
-A	After	
-B	Before	
-C	Context	

Obrigado.

Contato: joaopauloaramuni@gmail.com