

Variáveis e Vetores

João Marcelo Uchôa de Alencar

Universidade Federal do Ceará - Quixadá

27 de Setembro de 2019

O Subshell

- ▶ Cada execução de um *script* inicia um novo processo *shell* que vai interpretar as instruções nele contidas.
- ▶ As variáveis definidas no *shell* pai não são copiadas automaticamente para o *shell* filho.
- ▶ Para copiar as variáveis, precisamos usar o comando **export**.
 - ▶ Qualquer variável não exportada é ignorada pelos *subshells*.
 - ▶ O *subshell* acessa uma **cópia** das variáveis.
 - ▶ Um *subshell* pode exportar uma variável para seus *subshells*.
- ▶ Uma maneira de retornar um valor de um *subshell* é atribuir o retorno a uma variável.

O Comando *source*

- ▶ Através do *source*, você pode executar os comandos de um arquivo no próprio *shell* atual.
- ▶ Dessa forma, suas variáveis são atualizadas.
- ▶ Os arquivos *.profile* ou *.bashrc* são invocados com *source* toda vez que você faz *login*.

Principais Variáveis do Sistema

HOME	Caminho do seu diretório
PATH	Caminhos que guardam programas
PWD	Diretório corrente
LOGNAME	Seu nome de usuário
PS1	Caracteres do <i>prompt</i> primário
IFS	Separador padrão
TERM	Tipo de terminal
PROMPT_COMMAND	Executado ao final de cada comando

Exemplo

- ▶ Vamos criar uma variável no arquivo *.bashrc* chamada **LOGIN_DATE**. Ela deve armazenar a data que fazemos *login* no sistema.
- ▶ Vamos criar um diretório **bin** na sua pasta de usuário. Dentro dele, um *script* chamado *showLoginDate* (sem a extensão *.sh*) deve exibir na tela o conteúdo da variável **LOGIN_DATE**. Esse diretório deve ser inserido na variável **PATH**.

Usando as `{ }` para Controlar Variáveis

- ▶ `${!variavel}`: indireção.
- ▶ `${!padrao*}`: nome de variáveis de acordo com o padrão.
- ▶ `${variavel:-default}`: valor *default*.
- ▶ `${variavel:=default}`: atualiza a variável com valor *default*.
- ▶ `${variavel/de/para}` substitui uma ocorrência do padrão.
- ▶ `${variavel//de/para}` substitui todas as ocorrências.

Usando as { } para Gerar *Strings*

- ▶ {lista}
- ▶ {inicio..fim}
- ▶ prefixo{lista} ou prefixo{inicio..fim}
- ▶ {lista}sufixo ou {inicio..fim}sufixo
- ▶ prefixo{lista}sufixo ou prefixo{inicio..fim}sufixo
- ▶ {inicio..fim..incr}

Vetores ou *Arrays*

- ▶ `vet=(elemento0 elemento1 elemento2 ... elementoN)`
- ▶ `vet[n]=valor`
- ▶ declare -a vet (sem alterar caso exista)
- ▶ declare -A vet (vetor associativo)
- ▶ read -a vetor

Acessando Vetores

- ▶ `${vet[0]}`
- ▶ `${!vet[@]}`
- ▶ `${#vet[@]}`
- ▶ `${vet[@]:0:2}`

Exemplo

- ▶ Fazemos um *script* chamado *contadorVetor.sh*
- ▶ O usuário deve informar, sem utilizar parâmetros, um número por vez.
- ▶ O número deve ser armazenado em um vetor.
- ▶ Quando o usuário informar o caractere *q* o *script* deve informar quantos números foram inseridos no vetor e terminar. Devemos usar vetores!

Atividade

Faça um *script* chamado *contaPalavras.sh* na pasta *atividades/atividades08* que pergunte ao usuário o nome de um arquivo de texto e para cada palavra do arquivo diga quantas vezes ela aparece no texto.

```
$ cat arquivo.txt
a casa que vivo é boa.
boa casa é.
$ ./contaPalavras.sh
Informe o arquivo: arquivo.txt
casa: 2
boa: 2
é: 2
a: 1
que: 1
vivo: 1
```