UNIX: Uso básico do Shell

O shell

O *shell* é o programa que permite a interação do usuário com o sistema, em modo texto. Em UNIX existem vários tipos de *shell*, com funcionalidades diversas. Os mais conhecidos são:

- *Bourne shell* (sh): o mais antigo, está presente em quase todos os sistemas, pois é necessário para diversas operações administrativas.
- *C shell* (csh): de sintaxe mais simples, é mais fácil de usar e por isso preferido pelos usuários iniciantes. É o *shell* default do sistema *Solaris*.
- *Korn shell* (ksh): mistura características de ambos os anteriores. Muito usado em ambientes IBM e HP.
- *BASH Shell* (bash): *Bourne-Again Shell*, é uma extensão do *sh* e utilizado como padrão nas máquinas Linux.

Neste curso estaremos usando o *shell* chamado **bash**. Para informações específicas sobre esse **shell** consulte sua página de manual (`man bash`).

A linha de comando

O uso do *shell* se baseia em comandos digitados em uma linha de comando. Os comandos têm normalmente a seguinte sintaxe:

`comando opções parâmetros`

Por exemplo:

```
$ ls -l /etc
```

Cada comando possui suas opções, que podem ser conhecidas através da página de manual respectiva. Ao receber um comando digitado pelo usuário, o *shell* executa as seguintes ações, em sequência:

1. ler a linha de comando e quebrá-la em palavras
2. armazenar a linha no histórico
3. resolver variáveis de ambiente
4. resolver aliases de comandos
5. resolver *wildcards* (caracteres curingas ou metacaracteres)
6. lançar cada um dos comandos com seus parâmetros
7. esperar o fim dos comandos
8. voltar ao início

Essas ações são executadas repetidamente até o encerramento do *shell*, seja pela morte do processo ou pelo comando `exit`.

Wildcards

Wildcards são caracteres que permitem designar nomes genéricos para arquivos e diretórios na linha de comando. O Bash permite o uso das seguintes possibilidades de *wildcards*:

* : designa qualquer *string* com zero ou mais caracteres. Para listar todos os arquivos que começam com a letra

c no diretório `/usr/bin`:

```
$ ls -l /usr/bin/c*
```

Idem, arquivos cujo nome contém a *string* coa em qualquer posição:

```
$ ls -l /usr/bin/*coa*
```

Idem, arquivos cujo nome contém as letras c, e e s em sequência:

```
$ ls -l /usr/bin/*c*e*s*
```

? : designa qualquer caractere único. Para listar todos os arquivos cujo nome contém três letras no diretório `/etc`:

```
$ ls -l /etc/???
```

Idem, arquivos cujo nome contém a como segunda letra:

```
$ ls -l /etc/?a*
```

`[]` : designa um conjunto de caracteres. Para listar todos os arquivos que começam com uma vogal minúscula no diretório `/etc`:

```
$ ls -l /etc/[aeiou]*
```

Idem, arquivos contendo dois dígitos consecutivos:

```
$ ls -l /etc/*[0-9][0-9]*
```

`{,}` : permite definir padrões alternativos. Por exemplo, para remover todos os arquivos `.doc` ou `.pdf` do diretório `/tmp`:

```
$ rm /tmp/{*.doc,*.pdf}
```

Idem, para remover arquivos contendo as strings `abc` **ou** `xyz`:

```
$ rm /tmp/*{abc,xyz}*
```

Variáveis

O *shell* permite a criação de variáveis para armazenar valores durante uma sessão de trabalho ou em scripts. Existem dois tipos de variáveis: locais e exportáveis ou variáveis de ambiente.

As variáveis exportáveis ou de ambiente: são acessíveis a todos os comandos lançados pelo *shell* (são transferidas a todos os processos filhos daquele *shell*). Normalmente são escritas em letras maiúsculas. Para criar uma variável de ambiente basta executar `export VAR=valor`. Observe que não pode haver espaços em brancos entre o nome da variável, o sinal de atribuição = e o conteúdo da variável. Para destruir uma variável basta utilizar o comando `unset`. Exemplo:

```
$ export JDKHOME=/usr/local/jdk
$ echo $JDKHOME
$ unset JDKHOME
```

Para acessar o conteúdo de uma variável basta utilizar o caractere $ antes do nome da variável. Um outro exemplo de utilização:

```
$ export JDKHOME=/usr/local/jdk
$ echo ${JDKHOME}
$ unset JDKHOME
```

Algumas variáveis de ambiente exportáveis têm valor pré-definido pelo *shell*, e são importantes para o funcionamento normal da sessão de trabalho. Vejamos as principais (a lista de variáveis exportáveis pode ser consultada através do comando `env`):

- `PATH` : define os diretórios onde podem ser encontrados programas executáveis
- `MANPATH` : define os diretórios contendo páginas de manual
- `USER` : nome de login do usuário da sessão
- `HOME` : diretório *home* do usuário
- `HOST` : nome do *host* (máquina)
- `SHELL` : *shell* ativo
- `TERM` : tipo de terminal
- `MAIL` : localização da caixa de correio do usuário no sistema local

As variáveis locais são acessíveis somente dentro do *shell* onde foram criadas. Para criar uma variável local basta fazer `var=valor`, repare que não pode existir espaços em brancos entre o nome da variável, o sinal de atribuição = e o conteúdo da variável. Para destruir uma variável basta utilizar o comando `unset`.

```
$ nome=joao
$ echo $nome
$ touch $nome.txt
$ unset nome
$ nomes=( joao maria pedro antonio )
$ echo ${nomes[2]}
$ mkdir ${nomes[3]}
$ unset nomes
```

As variáveis locais podem armazenar listas de valores, úteis sobretudo na construção de *scripts*. A forma de definição e uso de variáveis contendo listas segue a seguinte sintaxe:

- `$nome` : o conteúdo da variável nome
- `${nome[x]}` : o conteúdo da variável, na posição x
- `${nome[@]}` : o conteúdo de toda a lista de valores contida na variável
- `${#nome[@]}` : o número de posições (entradas) da variável
- `${#nome}` : o número de caracteres da variável (na entrada indicada, se existir mais de uma)

Vejamos alguns exemplos:

```
$ nome=( joao maria pedro antonio luis )
$ echo $nome
joao
$ echo ${nome[3]}
antonio
$ echo ${#nome[@]}
5
$ echo ${#nome}
4
```

Aliases

O *shell* permite facilmente a criação de novos comandos ou a redefinição de comandos já existentes, através de aliases. Por exemplo, para criar um alias `copy` para o comando `'cp -i'`, basta executar:

```
$ alias copy='cp -i'
```

Assim, `copy` torna-se um comando válido durante a sessão corrente (veremos logo a seguir como criar aliases permanentes). Os aliases podem usar variáveis de ambiente também, como mostra o seguinte exemplo:

```
$ alias ola='echo "Olá $LOGNAME, como vai você?"'
```

Para desativar um alias basta usar o comando `unalias`:

```
$ unalias ola
```

Os aliases existentes na sessão podem ser consultados usando o comando `alias` sem parâmetros.

Histórico

Todos os comandos executados em uma sessão são memorizados pelo *shell*, e podem ser consultados e reexecutados quando desejado. Para consultar o histórico basta executar o comando `history`. Os comandos que são digitados são armazenados dentro do arquivo `.bash_history` que está em seu diretório de trabalho. Os comandos de uso do histórico começam sempre pelo sinal `!` (ponto de exclamação). Vejamos os mais comuns:

- `!132` : reexecuta o comando de número 132 do histórico
- `!-2` : reexecuta o penúltimo comando
- `!!` : reexecuta o comando anterior (equivale a !-1)
- `!gr` : reexecuta o último comando começando com as letras `gr`

Configuração

Ao ser lançado, o *shell* lê arquivos de configuração que contém comandos para definir o ambiente de trabalho de cada usuário. A sintaxe dos arquivos de configuração é bastante simples, similar à sintaxe das linhas de comando. O *Shell Bash* lê os arquivos de configuração, caso existam, na sequência apresentada a seguir:

- `/etc/profile` : contém configurações comuns a todos os usuários; executado no momento do login.
- `~/.profile` : contém configurações específicas ao usuário; executado no momento do login.
- `/etc/bash.bashrc` : contém configurações comuns a todos os usuários; executado durante lançamento de cada shell.
- `~/.bashrc` : contém configurações específicas ao usuário; executado durante lançamento de cada *shell*.

Para criar variáveis e aliases permanentes, ou para executar comandos ao iniciar o *shell*, basta inserir os comandos correspondentes no arquivo desejado. Por exemplo, se inserirmos o comando

```
alias ola='echo "ola $LOGNAME, como vai voce"'
```

no arquivo `~/.bashrc`, no próximo lançamento de *shell* ele será lido e fará parte dos comandos disponíveis ao usuário.

Exercícios

1. Crie aliases para os comandos `copy`, `ren`, `del`, `type`, `dir` e `move` do MS-DOS, e os inclua no arquivo de inicialização do *shell*.
2. Coloque uma mensagem de boas-vindas na inicialização do *shell*, com o conteúdo abaixo. Para as linhas de data e hora, consulte as páginas de manual dos comandos `date` e `echo`.

```
Ola usuário joão
Seja bem-vindo à máquina server1
Data: 07/04/1999
Horário: 19:35:42
```

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=unix:shell_basico

Last update: **2020/08/18 19:16**