

Trabalho Prático no. 5

Sistemas Operativos 2011/2012

Eng.º Vítor Manuel Ferreira

Escola Superior de Tecnologia e Gestão do Politécnico de Viana do Castelo

11 de Abril de 2012

Conteúdo

1	Objectivos	1
2	Revisão Teórica	2
2.1	O que é a "shell"	2
2.1.1	Objectivos de uma "Shell"	2
2.1.2	Tipos de "Shell"	2
2.2	Variáveis de <i>Ambiente</i>	2
2.2.1	Definição de variáveis	2
2.3	Caminho de pesquisa de comandos	3
2.4	A prompt	3
2.5	História de comandos	4
2.6	Redefinição de comandos	4
2.7	Ficheiros de inicialização	4
2.8	<i>Shell Script</i>	5
3	Exercícios propostos	5

1 Objectivos

No final deste trabalho prático deverá ser capaz de [?] [?] [?]:

- Definir e utilizar variáveis de ambiente da "Bourne Again Shell" (bash)
- Definir o caminho de pesquisa de comandos
- Definir a prompt da shell
- Utilizar a história de comandos
- Redefinir comandos
- Conhecer e alterar os ficheiros de inicialização
- Saber identificar uma "shell script".

2 Revisão Teórica

2.1 O que é a "shell"

A shell é o **programa** que serve de interface entre o utilizador e o sistema operativo. De um modo mais simplista, a shell é o programa usado para lançar todos os outros programas. Também se costuma designar a shell como interpretador de comandos UNIX capaz de ser usada para programação simples, sendo este o tema a estudar ao longo deste trabalho no. 5.

2.1.1 Objectivos de uma "Shell"

Utilização interactiva - A shell espera pela introdução de comandos, interpreta-os e executa-os.

Personalização de uma sessão Linux - A shell define variáveis para controlar a sessão interactiva (algumas são previamente definidas pelo sistema). As variáveis podem ser definidas em ficheiros de inicialização e ficam definidas após o login. Estes ficheiros de inicialização podem conter comandos Linux ou de shell, para execução imediata após o login.

Programação - A uma série de comandos individuais combinados num programa é chamada de "shell script" (semelhante às "batch files" do MS-DOS).

2.1.2 Tipos de "Shell"

bash - Bourne Again Shell (Linux standard). Apenas esta será estudada nas aulas.

```
/bin/sh  
/bin/bash
```

csh - C Shell (utiliza sintaxe semelhante à linguagem C).

```
/bin/csh
```

tcsh - Extensão da C Shell.

```
/bin/tcsh
```

2.2 Variáveis de *Ambiente*

Um das propriedades mais importantes da shell são as variáveis. Com estas variáveis podemos controlar como a shell e os programas, por ela executados, se comportam. Para consultar as variáveis que existem basta usar o comando **set**.

2.2.1 Definição de variáveis

É de notar que, as diferentes shells têm diferentes variáveis de ambiente. A forma de definir as variáveis também muda ligeiramente. Como só vamos utilizar a bash apenas focaremos o nosso estudo nas variáveis de ambiente desta shell.

O nome da variável de ambiente pode ser qualquer conjunto de caracteres alfanuméricos, podendo incluir "-" e números (um número nunca pode iniciar o nome de uma variável) mas sendo, sempre, apenas uma palavra (sem espaços). Isto porque, a shell, como bem sabe, usa os espaços na linha de comando para distinguir os diferentes componentes de um comando (opções, argumentos e o nome do comando).

Para definir uma nova variável ou alterar o valor de uma variável já existente usamos o operador "=" sem espaços entre o nome da variável, o operador e o valor da variável.

Por exemplo:

```
$ nome = Camilo      ;dará mensagem de erro
$ nome=Camilo        ;atribui à variável ''nome'' o valor ''Camilo''
```

Para usar o valor da variável deve preceder-se o seu nome pelo operador "\$". Sempre que o "\$" é colocado antes do nome da variável, este é substituído pelo valor da mesma.

Por exemplo:

```
$ echo $nome          ;imprimirá no ecrã Camilo, enquanto
$ echo nome           ;imprimirá no ecrã nome
```

Os valores atribuídos às variáveis são sequências de caracteres. Estes caracteres podem ser explicitamente escritos ou ser o resultado obtido de um comando Linux. Na maior parte das vezes é necessário limitar os valores a ter em conta com aspas (" - *double quotes*), plicas(' - *single quotes*) ou apostrofes (' - *back quotes*). Cada uma destas formas de limitação de *strings* tem a sua função:

- as aspas e as plicas (" - *double quotes*, ' - *single quotes*) permitem a interpretação de várias palavras como um único parâmetro. No entanto, as aspas não retiram o significado ao carácter \$, ou seja, quando estiver colocado antes do nome de uma variável esse conjunto é substituído pelo valor da variável.
- Os apostrofes (' - *back quotes*) permitem executar um comando Linux e o seu resultado ser visto como um parâmetro da linha de comando.

Por exemplo:

```
$ echo "O nome é $nome"      ; imprimirá: O nome é Camilo
$ echo 'O nome é $nome'      ; imprimirá: O nome é $nome
$ echo `pwd`                 ; imprimirá: /home/aluno
$ echo `pwd`                 ; imprimirá: pwd
```

2.3 Caminho de pesquisa de comandos

A variável mais importante da *shell* chama-se **PATH**. Esta variável indica à *shell* as directorias onde deve procurar os programas executáveis.

Sempre que mandamos executar um comando com a *shell*, esta vai percorrer todas as directorias listadas na variável **PATH**, procurando um ficheiro executável com o mesmo nome. As directorias habitualmente listadas na variável **PATH** são:

```
/bin          ; comandos usados durante o arranque do sistema
/usr/bin      ; comandos standard do sistema
/usr/X11R6/bin ; programas relacionados com o sistema de janelas X
/usr/local/bin ; programas opcionais, instalados e compilados localmente
```

2.4 A prompt

A segunda variável mais importante da *shell* define a **prompt**, ou seja, o texto que aparece antes do cursor para introduzir os comandos.

A *bash* utiliza as variáveis **PS1**, **PS2** e **PS3**. A primeira é aquela que a *shell* usa normalmente. As restantes são utilizadas para continuar comandos que o utilizador possa deixar incompletos.

Para estas variáveis existem símbolos próprios. Os mais usados estão ilustrados na figura 1:

Existem outras variáveis importantes da *shell* como, por exemplo, **HOME** - directoria pessoal do utilizador (home directory), **USER** - nome do utilizador, **PWD** - directoria de trabalho actual.

Símbolo	Significado
\h	Nome do computador até ao primeiro '.'
\H	Nome completo do computador
\s	Nome da <i>shell</i>
\t	Hora actual no formato HH:MM:SS (24 horas)
\A	Hora actual no formato HH:MM (24 horas)
\u	Nome do utilizador
\w	<i>Working directory</i> actual
\W	Nome do <i>working directory</i> actual
!	Número do comando (dentro da história dos comandos do utilizador)
#	Número do comando na <i>shell</i>

Figura 1: Símbolos mais usados para as variáveis de ambiente da bash.

2.5 História de comandos

Esta é uma funcionalidade muito interessante da *shell*. A *shell* memoriza todos os comandos que o utilizador vai executando e, depois, permite repetir, editar e voltar a executar esses comandos. Com as teclas direccionais podemos recuperar os comandos anteriores e editar o seu conteúdo com as teclas de edição.

O comando **history** permite consultar a lista dos comandos já executados. Estes apresentam-se numerados, podendo ser repetidos utilizando o prefixo "!" seguido do respectivo número. Por exemplo:

```
$ !195      ; repete o comando 195
$ !!       ; repete o último comando
$ !cat     ; repete o último comando que começava pela palavra cat
$ !?sor.c ; repete o último comando que continha uma ocorrência dos
              ; caracteres "sor.c".
```

As variáveis **HISTSIZE** e **HISTFILE** da bash definem, respectivamente, o número de comandos que fica guardado no histórico e o ficheiro onde os mesmos são guardados quando fazemos *logout*.

2.6 Redefinição de comandos

Podemos criar os nossos próprios comandos ou alterar as funcionalidades dos comandos existentes, através do comando **alias**. Por exemplo:

```
$ alias ll="ls -l"      ; quando o utilizador escrever "ll, será
                          ; executado o comando "ls -l"
$ alias hist="history" ; permite invocar o histórico dos comandos
                          ; apenas com "hist"
```

2.7 Ficheiros de inicialização

Qualquer *shell* possui ficheiros de inicialização que são executados quando entramos no sistema (fazemos *login*) ou quando invocamos uma nova *shell*. Estes ficheiros permitem personalizar o ambiente de trabalho ao gosto do utilizador.

Quando é feito *login*, a *bash* lê e executa os comandos do ficheiro **/etc/profile**, se o ficheiro existir. Seguidamente procura os ficheiros **/.bash_profile**, **/.bash_login** e **/.profile**, por esta ordem, e lê e executa os comandos do primeiro que existir e for legível.

Quando é feito o *logout*, a *bash* lê e executa os comandos do ficheiro `/.bash_logout`, se existir. Quando é invocada uma nova *shell* são executados os comandos do ficheiro `/.bashrc`, se este existir. O conteúdo destes ficheiros é apenas uma sequência de instruções, que são executadas como se tivessem sido escritas na linha de comandos. Estes ficheiros só são executados se o dono for o próprio utilizador e estiver protegido contra escrita dos restantes utilizadores.

2.8 Shell Script

Podemos colocar instruções para a *shell* num ficheiro e dizer à *shell* para ler os comandos e executá-los. Neste sentido o ficheiro funciona como um programa de *shell*, executando os comandos como se fossem instruções de um programa. Um ficheiro que contenha comandos de *shell* intitula-se *shell script*. Esta ferramenta permite automatizar a maior parte das tarefas de administração do sistema e muitas tarefas de uso diário.

Para executar os comandos de um ficheiro *shell script* utilizam-se os comandos *sh* ou *.* seguidos do nome do ficheiro que contém o *script*.

3 Exercícios propostos

1. Na sua área de trabalho, dentro da directoria "PL"(Práticas Laboratoriais) já existente, crie uma nova directoria "trab5", onde passará a trabalhar ao longo de todo este trabalho prático. Neste momento deverá ter a seguinte "prompt":

```
username@machinename:~/PL/trab5$
```

2. Defina uma variável de nome **VAR_1** com o valor "*Sou a primeira variável definida*".
3. Leia o valor da **VAR_1**.
4. Lance uma nova *shell* e verifique se a variável **VAR_1** está definida.
5. Saia da *shell* e exporte agora a variável **VAR_1** para o ambiente.
6. Repita agora o exercício 4.
7. Na nova *shell* defina uma variável **VAR_2** com um valor introduzido no teclado.
8. Verifique o valor de **VAR_2** e exporte-a para o ambiente.
9. Saia da nova *shell* e verifique agora se **VAR_2** está definida.
10. Verifique o valor da variável **PATH** e acrescente, se necessário, a *working directory*.
11. Altere a *prompt* para incluir o número do comando e a hora no formato HH:MM.
12. Verifique os últimos comandos dados nesta sessão.
13. Defina o comando **listar** como sendo o comando "ls -la".
14. Coloque o comando **dir** a listar os ficheiros com cores.
15. Verifique o conteúdo dos ficheiros de inicialização do seu computador.
16. Verifique o conteúdo do ficheiro de *logout* do seu computador.
17. Defina a variável **DATA** com o resultado da data actual no formato AAAA-MM-DD.
18. Defina a variável **HORA** com o resultado da hora actual no formato HH:MM:SS.
19. Faça um *backup* do ficheiro **.bashrc**.

20. Faça com que sempre que se lança uma nova consola sejam definidas as variáveis **DATA** e **HORA** e sejam mostradas na consola.
21. Reponha o ficheiro **.bashrc** na sua forma original.

Referências