



Trabalho Prático no. 5

Introdução à programação em *shell script*





Revisão Teórica

- O que é a "shell"
- Variáveis de Ambiente
- Caminho de pesquisa de comandos
- A "prompt"
- História de comandos
- Redefinição de comandos
- Ficheiros de inicialização
- "Shell Script"





O que é a "shell"

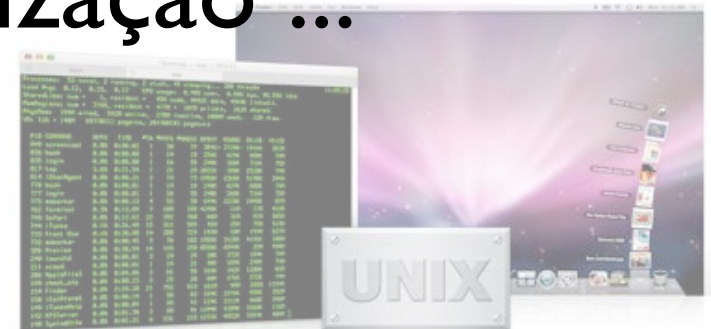
- Programa que serve de interface entre o utilizador e o kernel sistema operativo
 - Programa usado para lançar todos os outros programas
- ➡ Interpretador de comandos UNIX capaz de ser usada para programação simples





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 ...





Objectivos de uma "Shell"

- ...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).





Tipos de "Shell"

- **bash** - "Bourne Again Shell" (Linux standard) - apenas esta será estudada: `/bin/sh`
`/bin/bash`
- **csh** - C Shell (utiliza sintaxe semelhante à linguagem C): `/bin/csh`
- **tcsh** - Extensão da C Shell: `/bin/tcsh`





Revisão Teórica

✓ O que é a "shell"

- Variáveis de Ambiente
- Caminho de pesquisa de comandos
- A "prompt"
- História de comandos
- Redefinição de comandos
- Ficheiros de inicialização
- "Shell Script"





Variáveis de Ambiente

- Uma das propriedades mais importantes da shell
- Configuração: controlo de como a shell e os programas, por ela executados, se comportam
- O comando “**set**”: consulta das variáveis definidas





Definição de Variáveis

- Diferentes shells têm diferentes variáveis de ambiente, assim, a forma de definir variáveis também muda ligeiramente
- Nome da variável de ambiente: qualquer conjunto de caracteres alfanuméricos e números mas sendo, sempre, apenas uma palavra (não pode conter espaços)





Definição de Variáveis

- Para definir uma nova variável ou alterar o valor de uma variável já existente:
 - utilizar 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"
```

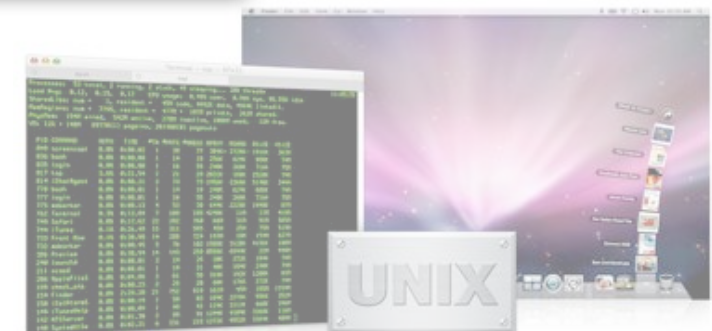




Usar o Valor da Variável

- 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 que:  
$ echo nome ; imprimirá no ecrã nome
```





Formas de Limitação de “Strings”

- Os valores atribuídos às variáveis são sequências de caracteres
- Estes caracteres podem ser explicitamente escritos ou ser o resultado obtido de um comando





Formas de Limitação de “Strings”

Na maior parte das vezes é necessário limitar os valores a ter em conta com:

1. aspas (” - double quotes),
2. plicas(’ - single quotes)
3. ou apostrofes (` - back quotes)





Formas de Limitação de “Strings”

(1, 2) 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.

```
$ echo "O nome é $nome" ; imprimirá: O nome é Camilo  
$ echo 'O nome é $nome' ; imprimirá: O nome é $nome
```





Formas de Limitação de “Strings”

(3) Os apostrofes (‘ - back quotes) permitem executar um comando Linux e o seu resultado ser visto como um parâmetro da linha de comando

```
$ echo ‘pwd’ ; imprimirá: /home/aluno  
$ echo 'pwd' ; imprimirá: pwd
```





Revisão Teórica

- ✓ O que é a "shell"
- ✓ Variáveis de Ambiente
 - Caminho de pesquisa de comandos
 - A "prompt"
 - História de comandos
 - Redefinição de comandos
 - Ficheiros de inicialização
 - "Shell Script"





Caminho de pesquisa de comandos

- A variável de ambiente muito importante chama-se PATH
- Esta variável indica à shell as directorias onde deve procurar os programas executáveis





Revisão Teórica

- ✓ O que é a "shell"
- ✓ Variáveis de Ambiente
- ✓ Caminho de pesquisa de comandos
 - A "prompt"
 - História de comandos
 - Redefinição de comandos
 - Ficheiros de inicialização
 - "Shell Script"





A “prompt”

- Uma segunda variável de ambiente também muito importante 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
- Para estas variáveis existem símbolos próprios. Os mais usados estão ilustrados na figura 1:

Símbolo	Significado
%h	Nome do computador até ao primeiro '.'
%H	Nome completo do computador
%s	Nome da shell
%t	Hora actual ao formato HH:MM:SS (24 horas)
%A	Hora actual ao formato HH:MM (24 horas)
%u	Nome do utilizador
%w	Working directory actual
%W	Nome do working directory actual
%l	Número de comando (dentro da história dos comandos do utilizador)
%#	Número de comando na shell
%p	Percentagem de espaço em disco
%c	Comandos por segundo
%d	Percentagem de espaço em disco (dentro do projecto)
%e	Percentagem de espaço em disco (dentro do projecto)





A “prompt”

Simbolo	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>





Revisão Teórica

- ✓ O que é a "shell"
- ✓ Variáveis de Ambiente
- ✓ Caminho de pesquisa de comandos
- ✓ A "prompt"
- História de comandos
- Redefinição de comandos
- Ficheiros de inicialização
- "Shell Script"



História de comandos

- 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



História de comandos

- 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:

```
$ !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".
```



História de comandos

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





Revisão Teórica

- ✓ O que é a "shell"
- ✓ Variáveis de Ambiente
- ✓ Caminho de pesquisa de comandos
- ✓ A "prompt"
- ✓ História de comandos
 - Redefinição de comandos
 - Ficheiros de inicialização
 - "Shell Script"



Redefinição de comandos

- Podemos criar os nossos próprios comandos ou alterar as funcionalidades dos comandos existentes, através do comando **alias**

```
$ 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"
```





Revisão Teórica

- ✓ O que é a "shell"
- ✓ Variáveis de Ambiente
- ✓ Caminho de pesquisa de comandos
- ✓ A "prompt"
- ✓ História de comandos
- ✓ Redefinição de comandos
- Ficheiros de inicialização
- "Shell Script"





Objectivos do trabalho no.5

- 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

