

1

Configuração do Interpretador de Comandos (Bash Shell)

Os objectivos deste módulo são os seguintes:

- Configuração do Bash Shell
- Variáveis de Ambiente
- Path
- Alias
- Ficheiros de Configuração

Introdução

O programa do shell (normalmente diz-se apenas shell) é um **interpretador de comandos** de linha. Existe muitos interpretadores ou programas shell para Linux como o csh, sh, tcsh etc o seu shell por defeito faz parte da definição do utilizador (ver ficheiro /etc/passwd).

Utilizaremos neste curso o "bash" Shell. Isto é apenas um ficheiro executável, cujo localização habitualmente é /bin/bash

O bash shell é muito popular, o seu nome é uma brincadeira das palavras inglesas *born again shell* (nascer novamente) e *bourne* (nome do autor do primeiro shell para Unix).

O bash shell utilize um conjunto de ficheiros de iniciação para criar um ambiente de trabalho e execução. Os ficheiros globais de configuração estão tipicamente no directório /etc ,a seguir ficheiros no seu directório raiz (*home directory*) podem depois adicionar ou reconfigurar o ambiente. Também o tipo de configuração varie conforme se o shell é do tipo *login* ou não e se o shell aceite interagir com o utilizador ou não (*interactive*).

Uma sessão "interactive login shell" é iniciado depois de fazer um login. Este tipo de Shell normalmente leia o ficheiro ~/.bash_profile ou o ficheiro o ~/.bashrc

Uma sessão "interactive non-login shell" é normalmente inicializada na linha de comando invocando o próprio programa do shell (escrevendo, /bin/bash) ou pelo comando **su** ou através dum xterm ou console iniciado dentro duma ambiente gráfica. Este tipo de Shell normalmente leia o ficheiro ~/.bashrc do utilizador para mais configurações do ambiente.

Uma sessão "non-interactive shell" pode ser iniciado, por exemplo executando um script (ficheiro de comandos). Não é "interactive" porque o script execute sem necessidade de comunicar com o utilizador, neste caso o ambiente do shell do progenitor é herdado sem alterações.

Ficheiros de Configuração

Ficheiros de configuração do shell são em primeiro lugar os ficheiros configurados pelo administrador do system "generic system-wide" e depois os ficheiros de configuração controlado por cada utilizador .

- /etc/profile : generic system-wide profile
- /etc/bashrc : generic system-wide .bashrc file for interactive bash shells.
- .bash_profile : user profile - processado no login
- .bashrc : setup-file processado cada vez se-inicia um interactive non-login shell
- .bash_login : processado no login
- .bash_logout : processado no logout

Nota: Ambiente das salas de D.I.- o ficheiro .bashrc é lido a abrir um terminal ?

Símbolos Especiais (wildcards)

Alguns símbolos são interpretados duma maneira especial pelo Shell e também pelo seu contexto de utilização. Exemplos são:

- no contexto dum nome dum ficheiro
 - * → qualquer string.
 - ? → qualquer carácter.
 - . → se o nome começa com este simbolo é um ficheiro escondido
- no contexto dum directório
 - . → o directório actual,
 - .. → o directório em cima e
 - ~ → o directório raiz.

Variáveis do Shell

Variáveis são identificadores (símbolos) que possuem valores que podem mudar/variá!

Algumas variáveis são definidas quando se faz o login e por norma são escritas com maiúsculas e persistem duma instância de Shell para outra (Variáveis do Ambiente) enquanto outras tem um âmbito da validade (scope) que é o Shell actual.

Exemplos de variáveis da ambiente são o HOME (directório raiz) o TERM (tipo de terminal) o SHELL (programa de shell a utilizar por defeito).

Para declara e inicializar uma variável utilize o sintaxe NOME=VALOR e para aceder ao seu valor utilize-se o sintaxe \$NOME

Para que as variáveis são validas em sub-shells criados a partir dum shell original é necessário declarar as variáveis "global" usando o comando "export".

Se tiver variáveis, definições etc. definidos num ficheiro pode carregar este ficheiro dentro da ambiente do shows actual com o comando **source**

Exemplos : source ~/.alias source ~/.bashrc

A Variável do Ambiente (Environment Variable) PATH

A variável PATH contém uma lista de directórios separados por ":". (Nota. Em Windows ";")

Para ver o seu path pode utilizar o comando **echo**.

```
alunos:~ crocker$ echo $PATH
/bin:/sbin:/usr/bin:/usr/sbin:/Network/Servers/docentes/crocker/bin:
```

A escrever o nome dum comando o sistema operativo pesquisará esta lista pela ordem definido até encontrar um ficheiro executável com o nome indicado. Depois executará o comando. Por exemplo a escrever o comando "ls" estamos a executar o ficheiro localizado no directório /bin que normalmente constará como um dos directórios no PATH.

Para ver se um comando específico está no path utilize-se o comando **which**

```
alunos:~crocker$ which ls
/bin/ls
```

Deverás inspeccionar o tipo e as propriedades do ficheiro **/bin/ls** com os comandos **file** e o próprio **ls**

```
alunos:~crocker$ ls -l /bin/ls
alunos:~crocker$ file /bin/ls
```

Um símbolo especial o "." . Este indique o "directório actual". Se este directório não estiver incluído no seu path estará condenado a escrever ./a.out para executar o ficheiro a.out localizado no directório actual porque a escrever apenas a.out o bash shell do sistema operativo não localizará o ficheiro e responderá com "bash: a.out command not found" Para rectificar esta situação altera o path assim :

```
PATH=$PATH:.
export PATH
```

Modificação do prompt:

Num shell interactive o **prompt** é que se vê na ecrã quando o programa está a esperar do próximo input/comando para interpretar. O valor do **prompt** está guardado na variável PS1.

Para modificar o prompt fazer PS1='value' (entre plicas ou aspas)

Exemplos PS1='>' PS1='\ls`\nOla>'

Alguns códigos que possam ser incluídos no prompt

- \d the date (day-of-the-week month day)
- \h the hostname (without the domain)
- \n start a new line
- \u the current username
- \w the working directory
- `comando` estes tipo de plicas implique que o comando é executado !

Por defeito PS1='\h:\w \u\\$ ' e está definido no ficheiro /etc/bashrc.

Agora verifique e altere o seu prompt !

Aliases or Shortcuts:

Para definir um alias o sintaxe é `alias comand=valor`.

```
Exemplos    >alias adeus='logout'
            >alias more='less'
            >alias dir='ls -lt'
            >alias cdjoao='cd ~/sisops/a12345 ; ls'
```

- Para cancelar um alias usar o comando `unalias`. Por exemplo `>unalias adeus`
- Para visualizar os aliases definidos basta escrever o comando sem opções: `>alias`

Os aliases podem ser inseridos num ficheiro de configuração (`.bashrc` etc) ou muitas vezes poderão ser inseridos num ficheiro `“.alias”` ou `“.bash_aliases”` que contém uma lista de atalhos subsequentemente lidos usando o comando `source` pelo um dos ficheiros de configuração.

Bash Shell Options - NoClobber:

Existe muitas opções do shell que possam ser configurados. Um exemplo é a opção “noclobber”. Se a opção `noclobber` estiver ligado não se pode, por engano, apagar um ficheiro utilizando redireccionamento. O comando para ligar a opção `noclobber` é:

```
set -o noclobber e para desligar set +o noclobber
```

Exemplo:

```
[crocker@penhas]$ PS1='Ola bash>'
Ola bash> set -o noclobber
Ola bash> touch trash
Ola bash> cat ficheiro > trash
bash: trash: cannot overwrite existing file
Ola bash> set +o noclobber
Ola bash> cat fich
```

Exercícios:

1. Criar a pasta `~/sisops` `>mkdir ~/sisops`
2. Criar um ficheiro `~/.alias` com alguns dos seguintes atalhos

```
alias lst='ls -alt | less'
alias dir='ls -a'
alias ll='ls -l'
alias rm='rm -i'
alias casa='cd;pwd;date'
alias cdso='cd ~/sisops; ls ; pwd'
alias indent='indent -npsl -bl'
alias cc='cc -Wall'
alias stdc='cd /usr/include; ls -l std*.h'
```

3. ler o ficheiro com o comando `source ~/.alias` e experimente!

```
> gedit ~/.alias & (Perguntar: que faz o & ?)
>source ~/.alias
>dir
>casa
```

4. Crie um ficheiro chamado “*ola.c*” com seguinte código usando um editora simples.

```
main() { printf("\nBem Vindo\n") ; }
```

```
formatar o ficheiro usando o comando “indent”    > indent ola.c
e voltar a ver as diferenças no ficheiro         > cat ola.c
```

5. Compile com as opções seguintes: `cc -Wall -o ola ola.c` e a seguir alterar o código fonte para eliminar os avisos da compilação do programa.
- Tem de inserir a linha `#include <stdio.h>` no início do ficheiro
 - Tratar do tipo e retorno da função `main`
6. Tentar executar o comando “*ola*” escrevendo
- `> ola`
 - `> ./ola`
7. Se tiver de escrever `./ola` modifique a variável `PATH` para incluir o directório actual `PATH=$PATH:.`

E tentar executar novamente usando `> ola`

8. Criar o directório `~/bin` com o comando `>mkdir ~/bin`
9. Mover o executável `ola` para o directório `~/bin` com o comando

```
>mv ola ~/bin
```

e executar

```
>ola
```

10. Consegue executar o comando `ola` sem escrevendo o caminho completo ? (deverá ver qual o valor do seu **path**?)
11. Alterar o `path` para incluir o directório `~/bin` e execute o comando novamente `ola` !

Um Ficheiro Típico `~/.bash_profile` `~/.bashrc`

```
PS1='\h \w \u sisops> '
PATH=$PATH:$HOME/bin:.
export PATH

if [ -f ~/.bash_aliases ]; then
  source ~/.bash_aliases
fi

set -o noclobber
```