

Trabalho Prático no. 3

Sistemas Operativos 2011/2012

Eng.º Vítor Manuel Ferreira

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

14 de Março de 2012

Conteúdo

1	Objectivos	1
2	Revisão Teórica	1
2.1	Filtros	1
2.2	Tabela de Expressões Regulares	2
2.3	Privilégios e Protecções de Ficheiros	3
3	Exercícios propostos	4

1 Objectivos

No final deste trabalho prático deverá ser capaz de [1] [2]: [1] [2]:

- Utilizar o redireccionamento através de *pipes* (`|`) de *stdout* (`>`, `>>`) e *stderr* (`2 >`, `> &`)
- Identificar e aplicar diferentes tipos de filtros e pesquisas sobre ficheiros e directorias
- Saber aplicar os filtros de manipulação de ficheiros **head**, **tail**, **wc**, **sort**, **split**, **cat**, **tee**
- Utilizar operações de pesquisa sobre directorias e ficheiros: **find** e **grep**
- Identificar e aplicar diferentes expressões regulares e caracteres especiais:

`^`, `$`, `*`, `[]`

- Saber aplicar diferentes expressões regulares e caracteres especiais nas operações de pesquisa sobre ficheiros: **grep** e **find**
- Conhecer os privilégios e protecções de ficheiros

2 Revisão Teórica

2.1 Filtros

Os filtros enviam o seu resultado para o *stdout*. Os filtros mais simples enviam para o *stdout* o conteúdo de ficheiros sem alterações: **cat**, **tee**, **head** e **tail**. Existem outros filtros que mostram uma versão alterada do conteúdo de ficheiros, partes seleccionadas dos ficheiros ou apenas informação sobre os ficheiros: **sort**, **split**, **cut** e **wc**. Alguns filtros permitem construir ficheiros com partes, ou a totalidade, de outros ficheiros: **paste**, **patch** e **diff**.

Símbolo	Significado
.	Uma ocorrência de qualquer carácter.
	Operador lógico <i>ou</i> : significa uma palavra ou a outra.
?	Uma ou zero ocorrências do carácter anterior, quer este seja normal ou especial.
*	Zero ou mais (qualquer quantidade) ocorrências repetidas do carácter anterior (normal ou especial).
+	Uma ou mais (qualquer quantidade) ocorrências repetidas do carácter anterior (normal ou especial).
{n}	Quantidade exacta de <i>n</i> ocorrências repetidas do carácter anterior (normal ou especial).
{n1,n2}	Desde <i>n1</i> caracteres até <i>n2</i> ocorrências do carácter anterior.
^	Início de linha.
\$	Final de linha.
[]	Uma ocorrência de qualquer dos caracteres listados dentro dos parênteses rectos. Podem ser indicadas gamas contínuas de valores utilizando o “-” e podem ser negadas listas de valores utilizando “^” no início da lista.
\<palavra>	Os caracteres dentro de “<” e “>” constituem uma palavra completa.
\	Desactiva o significado do carácter seguinte, se for carácter especial.

Figura 1: Conjunto de símbolos e caracteres que possuem um significado especial com o comando **grep**.

2.2 Tabela de Expressões Regulares

O comando **grep** procura no conteúdo de ficheiros alguns padrões, informa em que ficheiros o padrão foi encontrado e apresenta no *stdout* as linhas respectivas. Com este comando podemos procurar mais do que um padrão simultaneamente. Os padrões desejados são introduzidos previamente num ficheiro e posteriormente lidos pelo comando.

No comando **grep** é possível utilizar expressões regulares (figura 1) para procurar padrões específicos.

Para procurar todas as linhas iniciadas por *Anaconda*.

```
username@machine_name:~/PL/trab3$ grep "^Anaconda" nome_do_ficheiro_a_procurar
```

Para procurar todas as linhas que acabam em *trato*.

```
username@machine_name:~/PL/trab3$ grep "trato$" nome_do_ficheiro_a_procurar
```

Para procurar todos os padrões de três caracteres que comecem por *m* e terminem em *e*.

```
username@machine_name:~/PL/trab3$ grep "m.e" nome_do_ficheiro_a_procurar
```

NOTA: O ponto (.) substitui apenas um carácter.

Para procurar 0 ou mais instâncias consecutivas de um carácter.

```
username@machine_name:~/PL/trab3:~/PL/trab3$ grep "c*" nome_do_ficheiro_a_procurar
```

NOTA: O asterisco procura os padrões *c*, *cc*, *ccc*, *cccc*, ...

Para verificar qualquer texto de quatro caracteres que comece por dia e não termine em 5 ou 7, ou seja, todos menos dia5 e dia7.

```
username@machine_name:~/PL/trab3:~/PL/trab3$ grep "dia[~57]" nome_do_ficheiro_a_procurar
```

NOTA: Verifica-se que o carácter `^` dentro de `[]` indica exclusão dos caracteres seguintes e que classes de caracteres são definidas por `[]`, com o carácter `-` a indicar gama de valores.

O comando **find** permite, depois de encontrados os ficheiros desejados, executar outros comandos sobre os ficheiros. A sua sintaxe é ligeiramente diferente dos outros comandos.

2.3 Privilégios e Protecções de Ficheiros

Cada linha do comando **ls -l** representa um ficheiro ou directoria e contém informação organizada em várias colunas:

```
drwxrwxr-x 2 ap linuxusr 4096 Mar 11 17:13 S0_04_05
-rw-rw-r-- 1 ap linuxusr 1023 Mar 09 17:54 ficha3
```

colunas: ---1--- -2- -3- ---4--- -5- --6-- --7--

Lido da esquerda para a direita, as colunas representam:

1. as permissões
2. o número de links
3. o dono ("*owner*") do ficheiro
4. o nome do grupo a que pertence
5. o tamanho do ficheiro (em bytes),
6. a data e hora em que o ficheiro foi modificado
7. o nome do ficheiro ou directoria.

O primeiro campo consiste numa sequência de 10 caracteres que nos indicam as protecções de cada entidade. O primeiro carácter indica o tipo de ficheiro:

- **[-]** Ficheiro normal
- **[d]** Directoria
- **[l]** link para outra directoria ou ficheiro
- **[c]** Dispositivos do tipo carácter(c) ou bloco (b)
- **[b]** Ficheiros virtuais que estão associados a periféricos (portas série, discos, disquetes, terminais, ratos, colunas de som, modems, etc.)
- **[s ou p]** Representam sockets e pipes (ficheiros virtuais) utilizados para comunicação entre diversas aplicações

As permissões estão divididas em três blocos de três caracteres. Dentro de cada bloco, um "r" significa permissão de leitura (*read*); um "w" significa permissão para escrita (*write*) e um "x" significa permissão para executar (*execute*) o programa contido no ficheiro (nas directorias permite visualizar o seu conteúdo).

Os três caracteres do primeiro bloco representam as permissões associadas ao dono de cada ficheiro. **Os três caracteres do segundo bloco** representam as permissões dos utilizadores que pertencem ao grupo a que o dono está associado. **Os três caracteres do terceiro bloco** representam as permissões dos restantes utilizadores.

3 Exercícios propostos

Como todo o trabalho será realizado através da "bash" (consola), click em "consola" (ícon localizado no lado esquerdo do desktop), onde deverá aparecerá a seguinte "prompt":

```
username@machine_name:~$
```

Para registar as suas respostas utilize o documento modelo disponibilizado. Grave o documento sempre que fizer alguma alteração ao ficheiro. No fim, deverá submeter o documento na plataforma Moodle, clicando no link "submeter trabalho prático no. 3".

1. Na sua área de trabalho, dentro da directoria "PL" (Práticas Laboratoriais) já existente, crie uma nova directoria "trab3", onde passará a trabalhar ao longo de todo este trabalho prático (comandos: *pwd*, *mkdir*, *cd* e *clear*). Neste momento deverá ter a seguinte "prompt":

```
username@machine_name:~/PL/trab3$
```

2. Liste as primeiras 24 linhas do ficheiro "/etc/services" e guarde o resultado no ficheiro "p24" (comando: "head" com a opção: "-n").
3. Liste as últimas 35 linhas do mesmo ficheiro e guarde o resultado no ficheiro "u35" (comando: "tail" com a opção "-n").
4. Conte o número de caracteres, palavras e linhas do ficheiro "/etc/termcap" (comando: "wc").
5. Apresente apenas o número de caracteres da linha mais longa do mesmo ficheiro (comando: "wc" com a opção -L).
6. Apresente o número de linhas, palavras e caracteres de todos os ficheiros de configuração "/etc/*conf" (comando: "wc"). Modifique o comando de modo a não visualizar as mensagens de erro.
7. Divida o conteúdo do ficheiro "/etc/services" em ficheiros de 25 linhas com o prefixo "rc_" (comando: "split" com a opção "-l").
8. Reúna novamente todos os ficheiros gerados no exercício anterior no ficheiro "rc.services" (comando: "cat").
9. Procure todos os ficheiros do computador que comecem por "a" e acabem em ".html" (comando: "find"). Modifique o comando de modo a não visualizar as mensagens de erro.
10. Copie todos os ficheiros de configuração do computador "*conf" para a sua directoria (comando: "find" opção "-exec").
11. Procure a palavra "stddef" em todos os ficheiros "header" da biblioteca da linguagem C ("/usr/include") (comando: "grep").
12. Procure todas as linhas que começam por "typedef struct" nos ficheiros "header" da biblioteca da linguagem C.
13. Procure todas as linhas que acabam em ";" nos ficheiros "header" da biblioteca da linguagem C.
14. Conte o número de linhas encontradas no exercício anterior.
15. Conte o número de linhas que contêm ";" nos ficheiros "header" da biblioteca da linguagem C.
16. Conte o número total de linhas existentes nos ficheiros "header" da biblioteca da linguagem C.

17. Construa um ficheiro "myServices" com 30 linhas e 5 delas vazias. Copie agora todas as linhas não vazias desse ficheiro para "myServicesLines".
18. Conte o número de linhas de "/etc/services" que só têm 1 carácter.
19. Liste todos os ficheiros que têm acesso para leitura por todos os utilizadores dentro da directoria "/etc".
20. Liste apenas o nome das directorias existentes em "/dev".
21. Procure todos os ficheiros de código C que contenham ciclos "while", "for" e "until".

Referências

- [1] Fernando Pereira. *Linux Curso Completo - 5ª Edição*. FCA, 2005.
- [2] Paulo Trezentos. *Linux para PCs - Caixa Mágica - O Linux em port.-2ª Ed.Act.* FCA, 2005.