

fedora 



 ubuntu

 Mandriva

Curso de Formação LPIC-1

Exame 101



Curso Linux: formação

- › Fluxos de texto, pipes e Redireccionamento
 - › Redireccionamento de fluxos
 - › stdout, stdin, stderr
 - › pipes

Redirecionamentos

Mentalidade de Linux e Unix

- › Unix e Linux têm uma mentalidade de comandos que precisa de ser cultivada
- › Precisa de ser aprendida para um trabalho de sucesso
- › Tudo no âmbito da linha de comandos em Linux precisa de três coisas:
 - › Deve fazer uma coisa muito bem
 - › Deve aceitar input standard
 - › Deve produzir output standard
- › Torna-se possível conseguir muitas coisas com apenas um comando ou vários comandos ligados entre eles

Redireccionamentos

Input / Output

- › Suporte de fluxos separados para manipular dados na linha de comandos
- › São chamados *file descriptors* e são usados principalmente para:
 - › Enviar dados para e de programas
 - › Enviar dados de e para ficheiros
 - › Manipular erros

File descriptors

Nome	número fd	Ficheiro associado
/dev/stdin	0	/proc/self/fd/0
/dev/stdout	1	/proc/self/fd/1
/dev/stderr	2	/proc/self/fd/2

Redireccionamentos

Input / Output

- Entrada standard, ou *stdin* é tudo o que os programas aceitam, ou é assumido que aceitem
- Saída standard, ou *stdout*, é o texto ou dados produzido por um comando e aparece no ecrã ou na consola
- Erro standard, ou *stderr* é um fluxo paralelo para *stdout*, e por defeito aparece misturado no fluxo do *stdout* enquanto ocorrem erros.

Redireccionamentos

Input / Output

```
nl nomes.txt
```

```
cat /etc/fstab
```

```
find / -iname *.txt
```

```
Find / -iname *.txt 2> /dev/null
```

Redireccionamentos

Redireccionamentos de fluxos

- › Na missão por dados bons, ser possível de redireccionar ou alterar o destino de stdout e stderr, e adicionar alguma nitidez a stdin, é essencial às nossas tarefas.
- › < redirecciona o conteúdo de um ficheiro a um comando
 - › O descritor do ficheiro para < é 0
- › > redirecciona o fluxo de stdout para o ficheiro à direita do símbolo O descritor do ficheiro para > é 1
- › >> redirecciona stdout para um ficheiro, adicionando o fluxo corrente ao final do ficheiro, ao contrário de o reescrever. É um modificador de >

Redireccionamentos

Redireccionamentos de fluxos: stdin

- Redireccionar stdin consiste em enviar o conteúdo de um ficheiro para o fluxo stdin de um programa.

```
sort < numeros2.txt
```

```
cat numeros2.txt | sort
```


Redireccionamentos

Redireccionamentos de fluxos: stdout

- Redireccionar stdout consiste em, ou redirecionamento único (>) ou (>>) para adicionar.

```
cat nomes.txt > nomes4.txt
```

```
cat nomes.txt >> nomes2.txt
```

```
find / -iname *.txt > ficheiros_encontrados
```

Redireccionamentos

Redireccionamentos de fluxos: stderr

- Redireccionar stderr consiste em perceber que, por defeito, stderr, aparece na saída standard (stdout), misturados, mas separáveis

```
find / -iname *.txt 2> erros
```

- A secção 2> erros no comando redirecciona o stderr e coloca-o num ficheiro, deixando o fluxo stdout livremente aparecer no alvo por defeito (consola) ou mesmo para um outro ficheiro.

Redireccionamentos

Redireccionamentos de fluxos: stderr



- › O caracter > “agarra” o fluxo stdout e coloca dentro do ficheiro dados_bons, e stderr não é afectado e é enviado para a consola para visualização
- › “agarrar” ambos os fluxos e colocá-los em ficheiros separados é simples como adicionar o > com o descritor do ficheiro de stderr (2)

```
find / -iname *.txt > dados_bons 2> erros
```

Redireccionamentos

Redireccionamentos de fluxos

- Às vezes, toda a saída possível de um comando tem que ser “agarrada” porque irá causar problemas, tal como quando um comando é corrido em plano de fundo e estamos a usar um programa baseado na consola.
- Ter stderr a aparecer no ecrã durante uma sessão em que estamos a editar um ficheiro é desconcertante, no mínimo, e se estamos a configurar ficheiros importantes, é muito perigoso.

```
find / -iname *.txt > /dev/null 2>&1
```

Redireccionamentos

Pipes

- Um pipe (|) é usado para agregar dois ou mais programas, usando a saída de um como entrada para o segundo, tipicamente filtrando e alterando a saída a cada programa para onde os dados passam.

```
cat numeros2.txt | nl
```

- É essencial perceber a diferença do símbolo de redirecionamento e um pipe (|)

```
cat ficheiro1 | nl > pr
```

```
sort < nomes.txt | nl
```

Redireccionamentos

Pipes

- › Dois comandos que funcionam bem e complementam os pipes são *tee* e *xargs*
- › *tee*
 - › Desenhado para aceitar um fluxo de stdin e produzir dois pares completamente idênticos e que se podem utilizar
- › *xargs*
 - › Agarra numa lista de resultados de outro programa e usa-os, um por um num outro programa mais simples ou menos capaz

Redireccionamentos

Pipes: tee

- › Quando precisamos de colocar a saída de um programa num ficheiro, mas ao mesmo tempo queremos ver o que se está a passar.

```
find / -iname *.txt | tee findit.out
```

- › Desenhado para criar um log da saída de um fluxo para um ficheiro e passar outro fluxo para a consola.

Redireccionamientos

Pipes: xargs

```
locate -i readme | cat > mongofile.txt
```

```
less mongofile.txt
```

```
locate -i readme | xargs cat > mongofile.txt
```


Curso Linux

bibliografia

- › LPIC I, Exam Cram 2, Brunson - QUE Certification
- › LPI Linux Certification In a Nutshell, Pritchard, Pessanha, Langfeldt, Stranger & Dean – O REILLY
- › Linux Administration Handbook, Second edition, Nemeth Snyder Hein – Prentice Hall