

fedora 



 ubuntu

 Mandriva

# Curso de Formação LPIC-1

## Exame 101



# Curso Linux: formação

- › Vi / Vim
- › Vi básico
- › Vi avançado

# Vi

## Vi: básico

- › 3 modos no vi
- › Comando
- › Insert
- › lastline

- › Inicia sempre no modo comando
- › Edição propriamente dita é no modo de insert

# Vi

## Vi: modo de edição

- › Teclas comuns para invocar o modo de inserção desde o modo de comandos
  - › i – o método mais comum de entrar no modo de edição, deixando o cursor na posição corrente
  - › I – move-se para o início da linha corrente e desde aí age como a tecla i
  - › a – o segundo método mais comum é pressionar a tecla a, movendo o cursor um carácter para a direita, comportando-se como a tecla i a partir daí
  - › A – move-se para o final da linha corrente, comportando-se como a tecla i a partir daí
  - › o – usar esta tecla para iniciar uma nova linha a seguir à linha corrente. Se estamos na linha 3 no modo de comandos, pressionar o move a linha 4 (tornando-a na linha 5) e cria uma linha 4 vazia
  - › O – Cria uma nova linha na linha corrente. No modo de comandos, pressionar O move a linha 3, tornando-a na linha 4 e cria uma linha 3 vazia
  - › Ao pressionar uma destas teclas, no canto inferior esquerdo (esquerda da linha de mensagens) aparece a palavra -- INSERT --

# Vi

## Vi: modo de edição

- Voltar ao modo de comandos é simples, bastando pressionar a tecla ESC pelo menos uma vez, embora muita gente pressione duas vezes só para ter a certeza.
- A qualquer altura pode-se voltar ao modo de comandos desde o modo de inserção, pressionando a tecla ESC

# Vi

## Vi: Linha de mensagens

- › O fundo do ecrã do vi contem mensagens que nos podem ajudar
- › Variam consoante a situação e acção efectuadas
- › Abrimos um ficheiro – vi Lorem1.txt
  - › "Lorem1.txt" [noeol] 5L, 2243C 1,1 Top
- › Os números 1,1 são a linha e coluna corrente
- › O texto top é a posição corrente do cursor
- › Um novo ficheiro – vi novo.txt
  - › "novo.txt" [New File] 0,0-1 All

# Vi

## Vi: Abrir um novo ficheiro para editar

```
vi novoficheiro.txt  
vi /caminho/completo/novoficheiro.txt
```

- Abrir um novo ficheiro com uma string para procurar e colocar o cursor na linha da primeira ocorrência da string

```
vi +/Martins nomes.txt
```

- Ir para uma determinada linha assim que abrir o ficheiro

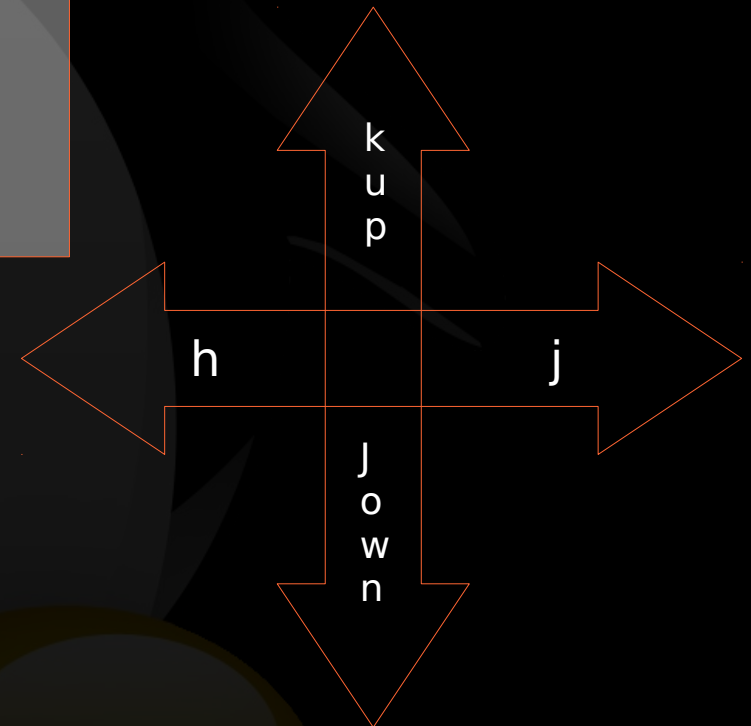
```
Vi +10 nomes.txt
```

# Vi

## Vi: Navegar num ficheiro

- › Teclas para se movimentar dentro de um ficheiro
  - › H – tecla para andar para a esquerda
  - › J – tecla para se movimentar para baixo
  - › K – tecla para se movimentar para cima
  - › L – tecla para andar para a direita

- › Teclas para movimentar páginas
  - › Ctrl + F – Uma página para a frente
  - › Ctrl + B – Uma página para trás
  - › Ctrl + D – Meia página para a frente
  - › Ctrl + U – Meia página para trás





# Vi

## Vi: Multiplicadores

- › Quase qualquer acção ou pressionar de teclas pode ser efectuado um X numero de vezes
  - › Basta carregar num numero antes de pressionar a tecla
  - › Cursor para a linha 5
    - › 5G ou 5gg
  - › Mover 10 palavras
    - › 10W
  - › Mover palavra a palavra
    - › w ou W (modo de comandos)
- 
- › Por vezes, após tanta inserção, tecla de escape deixa a zona de mensagens sem informação relevante
  - › Ctrl + G mostra o nome do ficheiro a editar, numero de linhas e posição corrente

# Vi

## Vi: Anular operações

- › Um numero grande e desconhecido de opções são operações de anulação
- › u em modo de comandos anula uma operação única, ou a ultima numa série de operações
- › Se abrir-mos um ficheiro, fizermos 30 operações e pressionarmos 30 vezes a tecla u, terminamos com o ficheiro exactamente como começámos.
- › Para anular todas as operações, não pressionar U – não é para isso. Em modo de comandos:
  - › :e! para anular tudo desde a ultima escrita em disco

# Vi

## Vi: Sair do ficheiro

- › Quando são efectuadas alterações no buffer, o vi não nos deixa sair normalmente, usado :q
- › Uma das situações mais difíceis do vi é a mensagem: E37: No write since last change (add ! to override)
- › Este erro pode ser corrigido adicionando o !
- › Para sair de um ficheiro que é apenas de leitura, ou ao qual não queremos efectuar alterações - :q!
- › Conhecido como *qbang* ou *quit dammit*

# Vi

## Vi: Guardar ficheiros

- A forma mais simples de guardar um ficheiro é :w no modo de comandos
- Sair e guardar é também bastante simples - :wq no modo de comandos
- Se formos *root* e estivermos a querer guardar um ficheiro de configuração que não conseguimos escrever, digitar :wq! obriga o ficheiro a ser escrito, desde que o que estiver a não deixar escrever o ficheiro seja uma permissão de escrita
- Duas formas adicionais de guardar e sair também estão disponíveis
  - :x – guarda e sai
  - Shift + ZZ (pressionar shift e tecla z duas vezes)

# Vi

## Vi: Inserir e adicionar texto

- › A tarefa mais comum numa sessão de vi é efectuar alterações ao texto num ficheiro
- › Isto envolve usar uma das seguintes teclas para alterar do modo de comandos para o modo de edição:
  - › i – inserir texto à esquerda do cursor
  - › I – inserir texto no início da linha
  - › a – adiciona à direita do cursor
  - › A – adiciona no final da linha
  - › o – começa uma nova linha por baixo da linha corrente
  - › O – começa uma nova linha antes da linha corrente

# Vi

## Vi: Alterar ou substituir texto

- › As seguintes são incrivelmente úteis quando estamos a alterar um ficheiro e precisamos de alterar um caracter, uma linha, uma frase ou apenas uma palavra:
  - › cw – altera uma única palavra a partir da posição do cursor. Para alterar, colocar o cursor no primeiro caracter da palavra
  - › c\$ - altera toda a linha desde a posição do cursor até ao final da linha
  - › r – substitui o caracter onde está o cursor
  - › R – substitui tudo até ser pressionada a tecla ESC
    - › Zona de mensagens aparece -- REPLACE --

# Vi

## Vi: Apagar texto e linhas

- › Um modo mais avançado do vi é apagar caracteres, palavras ou mesmo linhas.
- › ATENÇÃO ÀS LINHAS OU CARACTERES APAGADOS
  - › Pressionar u no caso de algum engano
- › x – apaga o caracter onde está o cursor
- › X – apaga o caracter antes do cursor
- › dw – apaga uma única palavra desde onde está o cursor.
  - › Para apagar a palavra inteira, colocar o cursor no primeiro caracter da palavra
- › dd – apaga completamente a linha corrente, independentemente da posição do cursor
- › D – apaga todo o texto desde a posição do cursor até ao final da linha
- › dL – apaga todo o texto desde o cursor até ao final do ecrã
- › dG – apaga todo o texto desde o cursor até ao final do ficheiro
- › d^ - apaga todo o texto desde o inicio da linha até ao cursor

# Vi

## Vi: Cortar, copiar e colar

- › O processo de mover texto pelo ficheiro no vi é um pouco complexo
- › As seguintes teclas são usadas para cortar, copiar e colar:
  - › yy – copia uma linha de texto para o buffer sem nome
  - › 3yy – copia 3 linhas de texto. Linha corrente do cursor mais as duas seguintes
  - › yw – copia a palavra onde está o cursor
  - › 3yw – copia 3 palavras. Onde está o cursor mais as duas seguintes
  - › p – cola o conteúdo do buffer sem nome à direita do cursor
  - › P – cola o conteúdo do buffer sem nome à esquerda do cursor
- › Copiar e colar linhas são efectuadas por linhas
- › Copiar e colar palavras são efectuadas por palavras
- › Yanking (y)
- › Pasting (p)



# Vi

## Vi: Cortar, copiar e colar

- › Por vezes, queremos apenas seleccionar partes da linha
- › Em modo de comandos
  - › Pressionar v
  - › Movimentar-se com as teclas de movimento até estar seleccionada todas as palavras desejadas
  - › Pressionar y
- › Para colar, as mesmas regras aplicam-se

# Vi

## Vi: buffers

- › O vi tem um total de 27 buffers
  - › 26 com nome (a-z)
  - › 1 sem nome que é re-escrito a cada operação
- › A não ser que tenhamos especificado, todas as operações são efectuadas para o buffer sem nome
- › Duas operações seguidas fazem com que o conteúdo da primeira operação seja re-escrito pela segunda
- › Operações nos buffers com nome são sempre precedidas por aspas ("), que indica que uma operação com um buffer com nome segue-se
- › Quando fazemos operações para o buffer com nome, o buffer tem que se referenciar ou com uma letra minúscula ou maiúscula (é parte do comando e não é enviado para o buffer)
  - › Letra minúscula – re-escreve o buffer
  - › Letra maiúscula – adiciona ao buffer

# Vi

## Vi: buffers

- › A seguinte string, em modo de comandos, copia 3 linhas para o buffer (a), re-escrevendo qualquer conteúdo lá existente
  - › “a3yy
- › A sintaxe para comandos de edição mantém-se o mesmo. Recordar apenas para preceder a operação com aspas (“), uma letra maiúscula ou minúscula, e depois qualquer operação que se deseje realizar.
  - › “A3yy – adiciona ao buffer (a) três linhas
  - › “ap – cola o conteúdo do buffer a

# Vi

## Vi: Pesquisas no vi

- Procurar texto nas ferramentas de Linux segue uma convenção comum.
- less, more e vi, uma contra-barras seguida da expressão executa uma procura (para a frente) no ficheiro a partir da posição do cursor, ou início do ficheiro, sobressaindo as palavras que correspondem
- Iniciar uma procura para cima no ficheiro é efectuada com um ponto de interrogação seguido da string a procurar
  - ?string
- Pesquisas só são efectuadas em modo de comandos
- Encontrar a próxima ocorrência (para a frente no ficheiro) é feito pressionando a tecla n.
- Encontrar a ocorrência anterior é efectuado pressionando N (Shift+n) – pesquisa para trás no ficheiro

# Vi

## Vi: Procurar e substituir

- Procurar por uma string ou caracter particular para substituir é comum ser efectuada no vi ou no sed para grandes conjuntos de dados.
- O vi e o sed partilham uma sintaxe comum, com pequenas diferenças.
- Saber procurar e substituir num deles irá ensinar também como fazer no outro
- Sintaxe:
  - acção/aprocurar/substituircom/modificador
- 

```
:s/Terra/Lua/  
:s/Terra/Lua/g  
:%s/Terra/Lua/g
```

# Vi

## Vi: Procuras parciais - ER

- › Procuras parciais no vi é algo que se deve saber
- › Numa procura parcial, procuramos texto que apenas sabemos uma parte

```
/^Um  
/kernel$
```

- › Mesmas regras das ER

```
/O caracter \*  
/vendo\.
```

- › Classe de caracteres

```
/6[1-9]
```

# Vi

## Vi: Opções no vi

- › Existem três formas de especificar opções no vi
  - › Directamente no vi (modo de comandos)
    - › :set number
  - › Num ficheiro de configuração
    - › .exrc (home do utilizador)
    - › Para global - /etc/exrc
  - › Na linha de comandos
    - › vi +"set number" ficheiro.txt (aspas devido ao espaço)

```
set number  
set tabstop=5  
set nohlsearch
```

```
:set number – opção activada  
:set nonumber – opção desactivada  
:se nu
```

- › Mais de 60 opções estão disponíveis
- › Para visualizar as definidas
  - › :set (modo de comandos)
- › Para visualizar todas as disponíveis
  - › :set all
- › Para desligar as opções (colocar no antes da opção):
  - › :set nonumber

# Vi

## Vi: vi avançado

- › Correr comandos externos desde o vi
  - › `:! ls -l`
  - › Se a saída for maior que o ecrã, é efectuado um *piping* para o comando `more`
- › Torna-se irritante no vi estar na frente de uma linha e querer usar a tecla de *backspace* para mover essa linha para o final da linha anterior
- › A tecla *backspace* apenas funciona na linha corrente
- › Para juntar-mos uma linha à linha anterior, colocamos o cursor na linha onde queremos unir a seguinte
  - › `Shift + j`



# Vi

## Vi: vi avançado – dividir janelas

- › Útil dividir a janela quando estamos a editar um ficheiro e queremos visualizar outro
  - › `:split` – divide o ecrã horizontalmente, com o mesmo ficheiro nas duas janelas
  - › `:vsplit` – divide o ecrã verticalmente, com o mesmo ficheiro em ambas as janelas
- › Mover-se entre as janelas é algo confuso
  - › `Ctrl + W` duas vezes
- › Para editar um ficheiro completamente diferente
  - › Editamos o primeiro ficheiro no vi
  - › `:split outroficheiro.txt`
- › Definir a altura da nova janela
  - › `:20split outroficheiro.txt` – dá-nos 20 linhas no ecrã
- › Para fechar o painel onde estamos
  - › `:close`
- › Fechar todos os painéis excepto onde estamos
  - › `:only`

# 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