ISCTE IUL
Instituto Universitário de Lisboa
Escola de Tecnologias e Arquitectura
Dep. de Ciências e Tecnologias da Informação

Sistemas Operativos
Ano letivo 2013/2014
1º Semestre

Conteúdo

Editor de texto: vi

Trabalhar remotamente no servidor implica, na maior parte dos cados, poder editar remotamente ficheiros. O Linux dispõe de um conjunto de editores em modo texto muito evoluídos. A título de exemplo podem referir-se o `emacs`, o `vi`, o `joe`, o `nano`, o `pico`, etc. Nesta disciplina será usado preferencialmente o `vi` (lê-se "vê i").

1 Elementos básicos

1.1 Entrada

Para entrar no vi dá-se o comando:

```
vi ficheiro
```

Se o ficheiro indicado já existir aparece o respectivo conteúdo; senão o ecrã aparece vazio.

Observação. Pode acontecer que parte do ecrã apareça preenchida com o sinal ~. Este sinal denota uma linha vazia. O `vi` preenche com este símbolo as linhas que se situam entre o fim do conteúdo do ficheiro e o fundo do ecrã. Evidentemente que se o ficheiro estiver vazio todas as linhas serão preenchidas desta forma.

1.2 Edição de Texto

O `vi` funciona em dois modos: *comando* e *texto*. No modo *comando* os caracteres dados pelo utilizador são interpretados como comandos. No modo *texto* os caracteres dados pelo utilizador são tomados como texto a escrever no ficheiro.

Ao entrar o `vi` fica no modo *comando*, para passar ao modo *texto* pode-se dar o comando `i` (isto é escrever a letra i); outras hipóteses são `a`, `o`, `I`, etc. Para regressar ao modo comando usa-se a tecla `<Esc>`. No modo texto funcionam os mecanismos habituais de escrita: carrega em <enter> para mudar de linha, em <backspace> para apagar, etc.

Exercício. experimente a sequência tecla i (passar ao modo texto), escrever texto, tecla <ESC> (regressar ao modo comando) repetidas vezes. As setas funcionam para se movimentar no texto.

1.3 Conjunto de comandos

Este pode ser um conjunto mínimo de comandos para editar ficheiros com o vi:

i a	passar ao modo texto
j k h l	movimento no texto
x dd	apagar um caracter / apagar uma linha completa
J	juntar duas linhas
:wq ZZ	guardar o ficheiro e sair
q!	sair sem guardar

Os comandos **`i a`** e **`A`** todos passam ao modo texto, diferido apenas o local onde o cursor de inserção de texto fica posicionado: no caso do i é a posição anterior; no caso do a a posição seguinte; no caso do A a posição a seguir ao fim da linha. Outras hipóteses são A, I, o, O. Os comandos j k h l movimentam o cursor sobre o texto. Serão pouco usados porque em geral as setas funcionam para o mesmo efeito. As teclas ^b e ^f (quer dizer Control-f e Control-B) movimentam página a página (identicamente poucas usadas em desfavor de PgDown e PgUp).

O comando J apaga um fim de linha, ou seja, faz com que a linha seguinte se junte àquela onde está o cursor.

Exemplo. Ao escrever, por engano, dá um `<enter>` e corta uma linha a meio. Solução: vai ao modo comando, posiciona-se na primeira "metade" da linha e dá o comando J para lhe juntar a outra "metade".

O comando :wq guarda o ficheiro em edição e sai do vi. O comando `:q!` sai do vi sem guardar o ficheiro. Os comandos como este, iniciados com o caracter `:`, aparecem na última linha do ecrã do vi. Depois de : o cursor vai para a última linha e o resto do comando vai aparecendo nessa linha; para terminar o comando carrega em <enter>.

1.4 Sair e gravar

O comando **`:w`** guarda o ficheiro (apenas; não sai do vi). Para sair pode-se usar apenas **`:q`**. Mas, caso não tenha gravado as últimas alterações aparecerá uma mensagem de erro. Para forçar a saída sem guardar dá-se o comando **`:q!`**. Estes comandos terminam com <enter>.

2 Mecanismos comuns de edição

2.1 Repetição de comandos

Muitos comandos podem ser antecedidos de um número e, nesse caso, serão repetidos o número de vezes indicado. Por exemplo **5x** repete 5 vezes o comando **x**, ou seja, apaga 5 caracteres (seguidos); da mesma forma o comando **5dd** apaga 5 linhas.

Exemplo. experimente escrever **5**, depois **i** para passar ao modo texto, escrever alguns caracteres e regressar ao modo comando com **<esc>**. O efeito é repetir a inserção 5 vezes.

2.2 Alteração de texto

O comando **R** permite alterar ("escrever por cima") um texto já existente. O comando passa o modo texto: começa a substituir o texto no ponto inicial e termina quando se der o <esc>, voltando ao modo comando. O comando **r** substitui apenas um caracter. Neste caso não vai para modo texto: depois de dado o caracter de substituição permanece no modo comando.

Exemplo. Imagine, por exemplo, que quer substituir um caracter por ***x***. Escreve **r*x***, ou seja o comando **r** e *x* o caracter de substituição. Note que continua em modo *comando*. Se fizer **3r*x*** substitui três caracteres por ***x***.

O comando **cw** permite substituir uma palavra (os caracteres até ao próximo separador: espaço, enter, etc). Neste caso o vi assinala com o símbolo **$** o âmbito da substituição: até esse ponto altera; a partir daí insere novo texto. O comando passa ao modo texto, terminando com **<esc>**.

Exemplo. Imagine, por exemplo, que tem num texto a palavra hipotótamo. Tendo o cursor da letra **p** e fazendo o comando **cw** vai substituir os caracteres potótamo. Se escrever **t<esc>** fica com a palavra **hit**; se escrever **lário<esc>** fica com a palavra **hilário**.

O comando **s** é util na forma ns permitindo substituir o número de caracteres indicado. Por exemplo fazendo **3s** aparece o símbolo **$** indicando a substituição de 3 caracteres; a partir daí é como no comando anterior.

2.3 Procura. Posicionamento

Para localizar determinado texto no conteúdo do ficheiro usa-se o comando **/texto**. Por exemplo **/main** procura a palavra main no ficheiro, ficando o cursor posicionado na primeira ocorrência que for encontrada (caso exista, evidentemente). O comando **/** aparece também na última linha: depois de escrita a **/** indica-se o texto a procurar terminando com **<enter>**.

Na sequência de uma primeira procura pode-se continuar com os comandos **n** que procura o mesmo texto seguindo para a frente no ficheiro ou **?** ou **N** que procuram o mesmo texto seguindo para trás no ficheiro.

Outras formas de posicionamento comuns são :*n* que posiciona o cursor na linha *n* do ficheiro, **g** que posiciona no inicio do ficheiro e **G** que posiciona no fim do ficheiro.

2.4 Substituição de texto

O comando **:%s/org/novo/g** permite substituir texto, indicando-se o texto original org a procurar no ficheiro e o texto novo que o irá substituir. Por exemplo **:%s/ola/ugue/g** substitui todas as ocorrências do texto "ola" pelo texto "ugue".

2.5 Copiar e colar

Se fizer **dd** a linha apagada não desaparece definitivamente; é guardada num buffer de serviço podendo ser recuperada. Para repôr a linha dá-se o comando **p** ou **P** (p repõe abaixo da linha corrente, P acima da linha corrente).

Observação. Um mecanismo simples de duplicar uma linha é **ddPP**: o **dd** apaga a linha salvaguandado-a no buffer de serviço; cada **P** repõe a linha uma vez. Esta seria digamos uma sequência "cut/paste/paste".

O comando **Y** copia uma linha para o buffer de serviço, mas sem a apagar.Assim, por exemplo, para copiar uma linha pode fazer com **Y** e **P** a sequência típica de "copy/paste": com **Y** copia a linha original; posiciona-se no sítio onde a pretende replicar; com P insere-a nessa posição. Se fizer a sequência **YP** duplica a linha no mesmo sítio.

O comando **nY** salvaguarda *n* linhas no buffer e é portanto útil para copiar várias linhas. Por exemplo, para copiar 5 linhas faz-se **5Y**, posiciona-se o cursor no local de destino e inserem-se as linhas salvaguardadas com **P**.

Além do buffer de serviço usado, por defeito pelo Y e P, podem ser usados outros buffer's para efeito semelhante. Os outros buffer's são designados pelo nome "x sendo x uma letra - por exemplo "a ou "b.

Os comandos de "copiar e colar" podem ser antecedidos do nome de um buffer específico e nesse caso farão a operação com esse buffer. Por exemplo: **"a5Y** salvaguarda 5 linha de texto no buffer **"a**. Estas linhas poderão mais tarde ser repostas com "aP.

2.6 Ficheiros

Para gravar o ficheiro em curso de edição usa-se o comando **:w**. Para sair do vi usa-se o comando **:q**. Os dois podem-se juntar na sequência :wq, já vista, que grava e sai.

O comando **:q** sá uma mensagem de erro e não resulta se houver alterações não gravadas. Para sair sem gravar usa-se a variante **:q!** que força a saída sem gravar.

No comando **:w** pode-se explicitar o nome do ficheiro (confirmando ou alterando o nome dado na entrada para o vi). Por exemplo o comando **:w poema.txt** salva a edição em curso no ficheiro poema.txt.

2.7 Troca de texto entre ficheiros

O comando **:r** permite importar o conteúdo de um ficheiro juntando-o à edição em curso. Por exemplo, o comando **:r /etc/passwd** insere na posição do cursor o conteúdo do ficheiro designado.

É possível também, com pouco esforço, manter dois ficheiros em edição "simultânea". Para abrir um segundo ficheiro dá-se o comando **:e** ficheiro (o comando só resulta depois de salvar as alterações ao original). Por exemplo dando o comando **:e /etc/passwd** passará a editar em simultâneo o ficheiro original e o ficheiro `/etc/passwd` (este último sem poder gravar as alterações, como é evidente). A partir daqui pode trocar entre os dois ficheiros em edição com o comando **:e#**.

Tendo os dois ficheiros abertos para edição pode também copiar blocos de texto entre eles. A ideia geral é a mesma que para fazer a cópia dentro do memso ficheiro: primeiro dá-se o comando **Y** para copiar depois o comando **P** para colar. Neste caso o processo será então: salvaguardar um conjunto de linhas num ficheiro; mudar de ficheiro com **:e**, posicionar o cursor no local de destino e inserir o texto com **P**.

2.8 Outros Comandos

O comando **u** faz *undo* (anula o efeito da última alteração). O comando **U** anula todas as alterações na linha. o comando **:redo** refaz o último *undo*

2.9 Personalização do `vi`

O `vi` dispõe de várias opções de personalização que incluem opções como mostrar os números das linhas, formatar automaticamente o texto etc. É possível cada utilizador alterar as definições gerais do sistema. Para isso basta editar o ficheiro `.vimrc`. Para definir uma determinada opção usa-se o comando `set`. Algumas opções interessantes são:

```
set number
```

mostra os números das linhas ao lado do código

```
set nonumber
```

não mostra os números das linhas ao lado do código

```
set syntax on
```

faz syntax highlight