

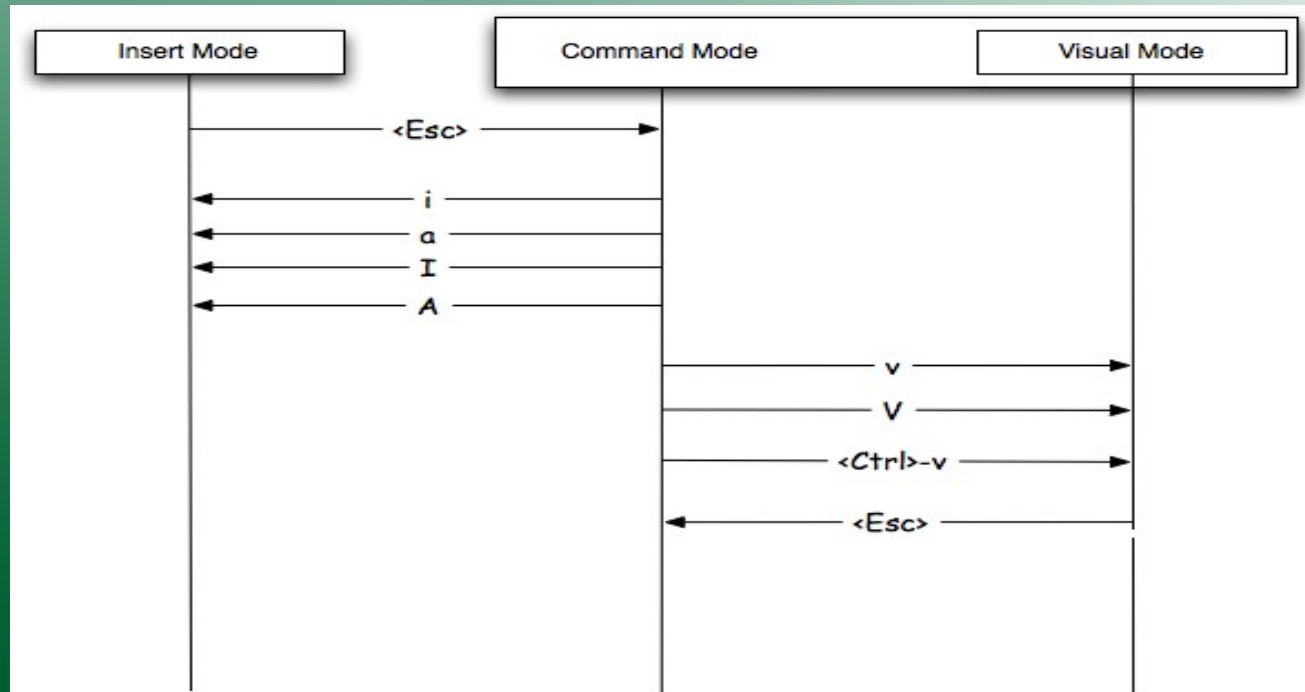
Aula 03

- Programa vim
- Execução de Programas
 - Foreground e Background
 - Execução sequencial
- Programas úteis
 - Gerenciamento de processos
 - Manipulação de diretórios
 - Manipulação de arquivos
 - Comandos Diversos

Editor de texto vim

- Vim vi Improvised (vim – 1991 / vi – 1976)
- Editor de rico em recursos
- Simples de utilizar (comandos padronizados)
- Disponível em todos os sistemas operacionais
- Indicado para usuários, programadores e administradores de sistemas
- Pode ser programado com funções ou expansões

Editor de Texto vim



Editor de Texto vim

- Comando para inserção
 - i Inserção antes do caractere atual
 - a Inserção depois do caractere atual
 - I Inserção no inicio da linha
 - A Inserção no fim da linha
- Comando para visual
 - v visual por caractere
 - V visual por linha
 - Ctrl+v Bloco

Comandos Vim

- “:” Acessa comandos
 - Ctrl+C Cancela comando
 - Ctrl+D completa comando
 - Cima e Baixo histórico de comandos
 - :help <expressão>
 - :help quit
 - :help copy
- :[range]co[py] {address}
- :help :s
- :help /s
- :help
- [tópico] Ctrl+]
- Voltar com Ctrl+T
- vimtutor
- man vim

Referencia <http://vimdoc.sourceforge.net/html/doc/>

Comandos Vim

- “d” apaga e copia – “dd” apaga linha
- “c” modifica – “cc” modifica linha
- “y” copia – “yy” copia linha
- “p” e “P” cola
- Repedindo n+ação
- ESC comando ESC
- “.” repete ação
- Movimentos
 - w próxima palavra
 - b palavra anterior
 - e final da palavra
 - 0 inicio da linha
 - ^ inicio da frase
 - \$ final da linha

vi / vim graphical cheat sheet

Esc
normal
mode

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence) end sentence	"soft" bol down	+ next line
\ goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto-format
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	} end parag.	
q record macro	w next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	[misc] misc	
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	. ex cmd line	" reg. spec	bol/ goto col	
a append	s subst char	d delete	f find char	g extra cmds	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	
Z quit	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un-indent	> indent	? find (rev.)			
Z extra cmds	X delete char	c change	v visual mode	b prev word	n next (find)	m set mark	reverse t/T/f/F	repeat cmd	/ find			

motion moves the cursor, or defines the range for an operator

command direct action command, if **red**, it enters insert mode

operator requires a motion afterwards, operates between cursor & destination

extra special functions, requires extra input

Q commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: quux(foo, bar, baz);

WORDS: quux(foo, bar, baz);

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:s/s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

Execução de Programas

- No GNU/Linux podemos executar programas:
 - Foreground execução interativa dependentes da sessão.
 - Background execução de tarefas independentes da sessão.
- Execução background
 - comando &
 - comando Ctrl+Z → bg
 - jobs comando para listar as execuções em background
 - fg voltar para execução foreground

Execução sequencial

- Podemos executar comandos um depois do outro com <enter>
 - ls
 - pwd
 - cd curso_linux
- Podemos colocar todos em uma linha separando com “;”
 - ls ; pwd ; cd curso_linux (sem relação entre os comandos)
- Podemos relacionar a execução dos comandos com o operador “AND” - “&&”
 - ls && pwd && cd curso_linux (relação entre os comandos)

Execução em paralelo

- Para executar comandos em paralelo, podemos colocar eles em background
 - `ls &`
 - `pwd &`
 - `find &`
- Execução de programas mesmo sem sessão
 - `nohup programa &> saida.txt &`

Gerenciamento de Processo

- Informações de processos
 - ps
 - pidof
 - pstree
- Matando um processo
 - Ctrl+C
 - kill
 - kill -l
 - kill --signal
- Signals
- 2 Ctrl+C termina o programa
- 3 Ctrl+\ termina o processo + despejo de memória
- 9 termina o processo imediatamente
- stty -a

Gerenciamento de diretórios

- ls - lista
- cd – troca diretório atual
- pwd – exibe diretório atual
- mkdir – cria diretório
- rmdir – remove diretório vazio
- rm – apaga arquivos e diretórios

Manipulação de arquivos

- cat – concatena arquivo e imprime (tac – inverso)
- rm – apaga arquivos e diretórios
- touch – altera data de modificação, cria arquivo
- cp – copia arquivo
- mv – move arquivo, renomeia
- head – exibir inicio de arquivo
- tail exibir final de arquivo
- grep busca em arquivo
- more and less
- sort – ordenar
- wc – contar letras, palavras e linhas
- cut – exibir partes
- diff – diferença entre arquivos

Comandos diversos

- date – data do Linux
- find – procura arquivos/diretórios
- time – tempo de execução
- seq – gerador de sequencia
- whereis – localizador de comandos e manuais
- which – localizador de comandos
- cal - calendario
- file – detecta tipo de arquivo
- Gerenciamento
 - free – memoria ram
 - df - disco
 - top - processos
 - ldd - bibliotecas
 - Passwd - senhas

Exercícios

Preparar o ambiente:

Utilizaremos o arquivo ~/GNULINUX/345.txt

- Crie um arquivo resultado.txt na pasta ~/GNULINUX/aula03/exercicios para colocar com suas palavras o resultado dos exercícios da aula 03.

Exercícios

1) Utilizando o vim entre na ajuda do copy.

copie o primeiro paragrafo do 1o capítulo para depois do último paragrafo do mesmo capitulo.

Escreva no resultado.txt o comando.

Exercícios

2) Utilizando o modo visual, faça a cópia do primeiro paragrafo do segundo capítulo para depois do ultimo paragrafo do mesmo capítulo.

Escreva no resultado.txt os comandos.

Exercícios

3) Escolha um outro comando do vim.

Veja no manual como utilizar, elabore um comando completo.

Escreva no resultado.txt o comando e o resultado com suas palavras.

Pode utilizar o arquivo 345.txt

(Indicação para case, sort e delete)

Exercícios

4) Tente utilizar os comandos de edição no arquivo 345.txt

d - apaga

c - modifica

com

w - palavra

0 - inicio da linha

\$ - final da linha

Descreva os resultados no arquivo resultado.txt

Exercícios

5) Escolha 3 comandos no slide comandos diversos.

Execute o man <comando> e escolha uma opção de comando.

Descreva com suas palavras a parte DESCRIPTION

Coloque no arquivo resultado.txt o comando com opção e o texto referente a DESCRIPTION

Exercícios

6) Descreva no arquivo resultado.txt comandos para

- a) Execução dos comandos sleep 1, sleep 2, sleep 3 – sequencial (sem relação entre os comandos)
- b) Execução dos comandos sleep 1, sleep 2, sleep 3 – paralelo
- c) Execução dos comandos sleep 1, sleep 2, sleep 3 – sequencial (Com relação entre os comandos)