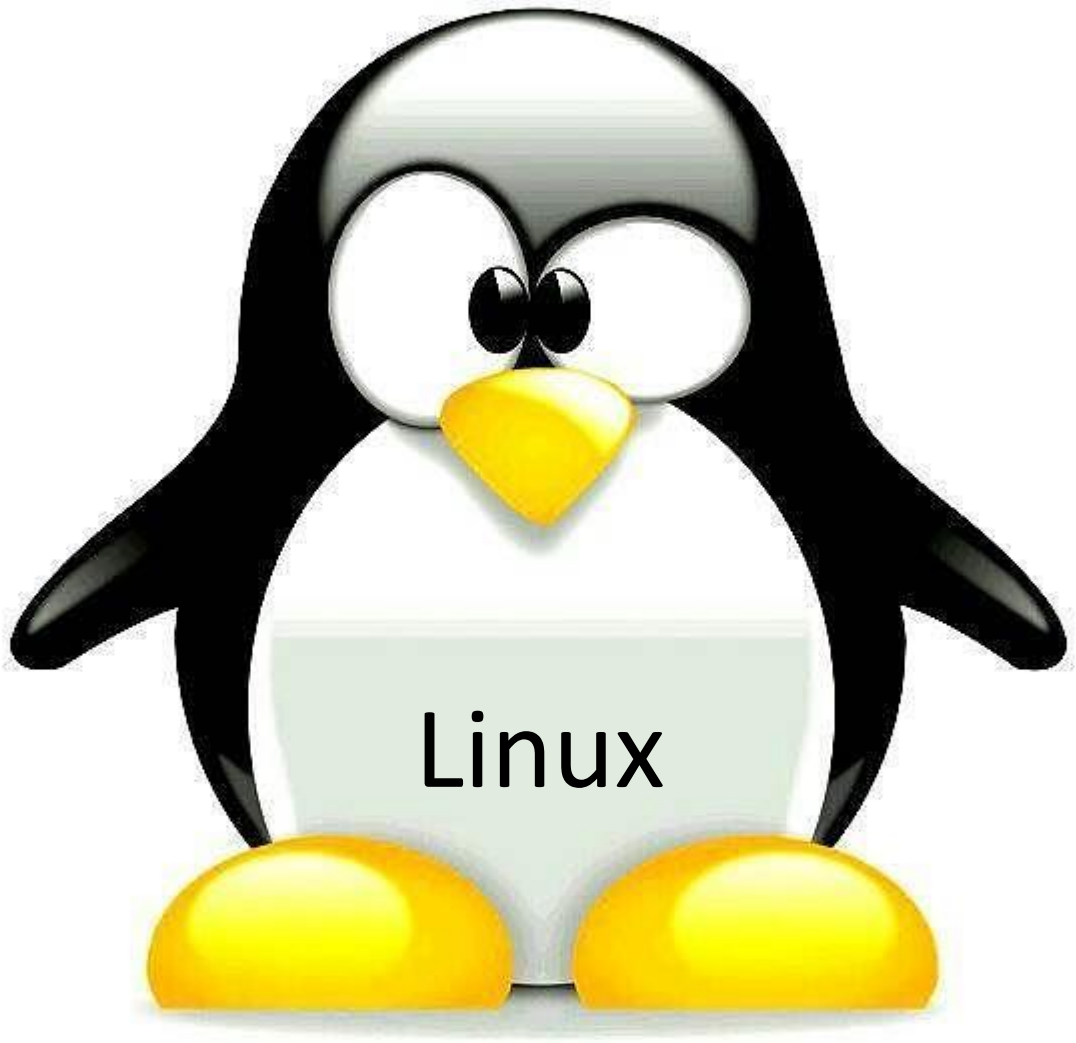
digital escola profissional

Nano

Vi

Shell Scripting



Editores de texto

Existem vários editores de texto, mas os mais usados são o nano e o vi/vim. Sendo o nano um editor mais fácil para principiantes visto apresentar as funções principais descritas no final do ecrân.

Embora sejam editores de texto são mais orientados à edição de programas, como se pode ver na imagem, usando cores diferentes dependendo o programa, facilitando bastante a programação, por no exemplo em linguagem C.

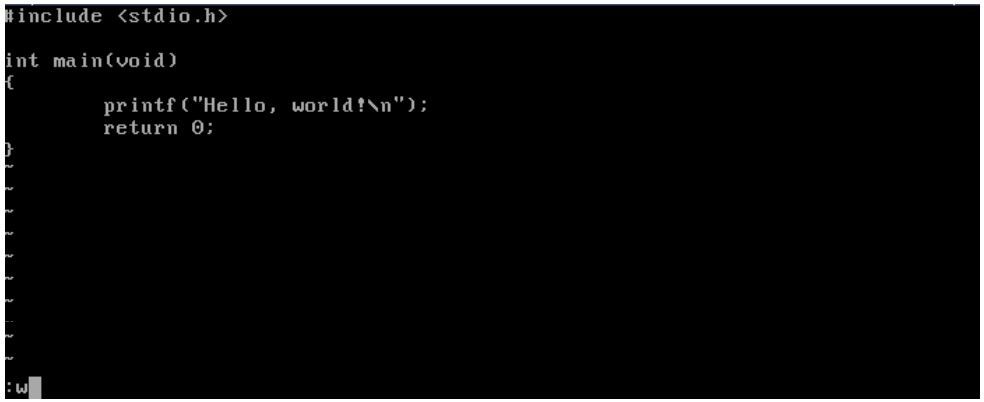
Editores de texto

O editor de programas (vou chamar de programas porque para editar texto ... imaginemos uma carta seria um completo suplício) muito usado é o vi, também conhecido por vim.

O Vim, tal como o nano já identifica o código e ajuda na escrita de programas em várias linguagens, não só o C mas também C++, Phyton, entre outras.

O vi foi a junção de um editor de linha (ed) com um editor de comandos (ex) e o nome vi vem de visual, já que ficou mais apelativo visualmente.

Embora comparado com as interfaces gráficas actuais, o visual é um pouco fora de moda ☺



Editores de texto

Como já falado o vi tem dois modos, modo de comandos, na gíria conhecido como modo escape já que para entrar neste modo se usa a tecla ESC onde podem ser dados comandos, alguns estão listados abaixo.

i – Entrar em modo de edição (insert) no local onde o cursor se encontra
a - Entrar em modo de edição (insert) no carácter a seguir ao cursor
A - Entrar em modo de edição (insert) no fim da linha onde o cursor se encontra
ESC – Sair do modo de edição (insert) para o modo de comandos (escape)
u – Desfazer a última acção
U – Desfazer todas as acções nesta linha
o – Abrir uma linha nova para edição
dd – Apagar a linha
3dd – Apagar o número de linhas indicado, neste caso serão 3

D – Apagar o resto da linha a partir da posição do cursor
C – Apagar o conteúdo da linha a partir da posição do cursor e substitui por novo texto. Termina com a tecla ESC
dw – Apaga uma palavra
4dw – Apaga o número de palavras indicado, neste caso 4.
cw – Altera esta palavra
x – Apaga o caracter marcado pelo cursor
r – Altera o caracter marcado pelo cursor
s – substitui um caracter e entra em modo edição
S – Substitui toda a linha e entra em modo de edição
~ - Altera as letras entre maiúsculas e minúsculas

Editores de texto

Para entrar no modo de inserção de texto podem usar-se vários comandos através de uma letra das listadas atrás, mas o mais usual é usar a letra i .

Para se deslocar ao longo do texto pode-se usar as setas do teclado, embora em teclados muito antigos que não tenhas setas se possam usar:

k - Move cursor para cima
j - Move cursor para baixo
h - Move cursor para a esquerda
l - Move cursor para a direita

Para sair e gravar também há várias opções:

Shift+zz ou :x – Grava o ficheiro e sai
:w – Grava mas continua a edição
:q – Sair sem gravar
:wq – Gravar e sair

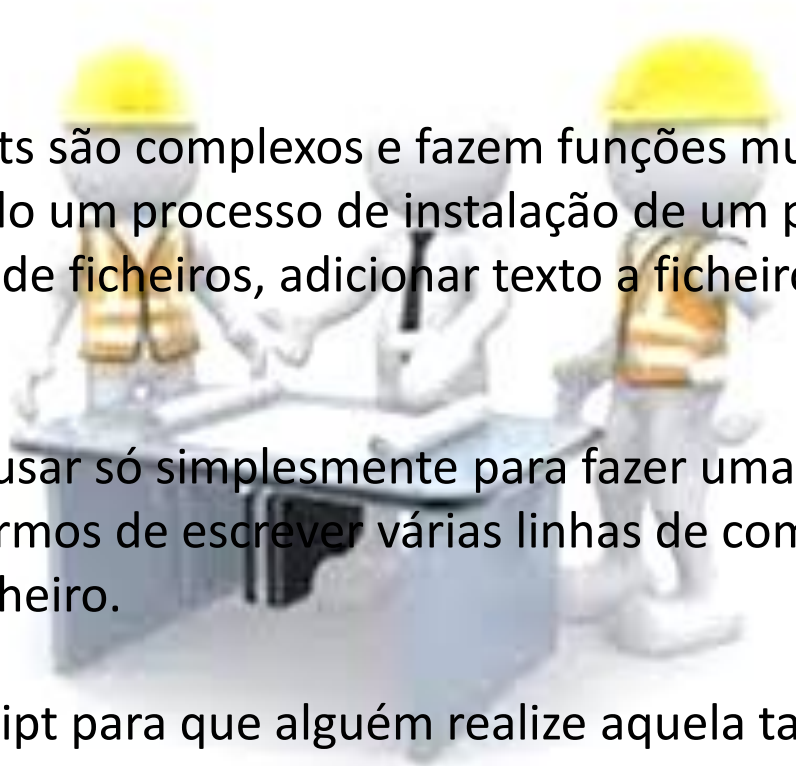
Shell Scripting

- Shell scripting é a forma de encadear comandos para obter um determinado objectivo.

- A maior parte dos shell scripts são complexos e fazem funções muito completas, como por exemplo um processo de instalação de um pacote de software ao fazer a cópia de ficheiros, adicionar texto a ficheiros e outras acções necessárias.

- Mas muitas vezes podemos usar só simplesmente para fazer uma actividade rotineira e não termos de escrever várias linhas de comando, juntando tudo num único ficheiro.

- Podemos ainda realizar o script para que alguém realize aquela tarefa por nós sem precisarmos de lhe estar a explicar comando por comando.



Shell Scripting

Para exemplificar aqui ficam dois exemplos muito simples, no primeiro caso iremos fazer o muito célebre "programa" que não faz rigorosamente nada a não ser uma mensagem que faz falta nestes dias ... Olá mundo ... Hello World!

my-script.sh

```
#!/bin/sh
# This is a comment!
echo Hello World        # This is a comment, too!
```

Para que se possa executar o programa teremos que o tornar executável com
$ chmod +x <ficheiro>
e depois correr com
$ ./<ficheiro>

Neste caso teremos:

$ chmod +x my-script.sh

$ ./my-script.sh

Shell Scripting

No segundo caso um também muito básico, na práctica é só correr o comando que nos mostra a data do sistema, mas com o formato como utilizamos normalmente em Portugal.

```
[root@localhost ~]# cat > data
date "+%d/%m/%y %H:%M:%S"
[root@localhost ~]# . ./data
25/03/20 00:46:41
```

Shell Scripting

Para que não seja só estar a “bater texto” já que tanto o nano como o vi/vim são maus editores de cartas, vamos tentar fazer uns Shell scripts básicos mas funcionais para treinar:

```
#!/bin/bash
echo -n “Diz um numero: ”
read num
if [[ $num -gt 10 ]]; then
echo “O número é menor que 10.”
else
echo  “O número é maior que 10.”
fi
```

Shell Scripting

O próximo identifica se o número é par ou impar ☺

```
#!/bin/bash

echo -n "numero? "
read num

if [ $(($num % 2)) -eq 0 ]; then
echo "par"
else
echo "impar"
fi
```

Shell Scripting

O próximo envia um email ao root de quais os utilizadores que entraram em sessão ☺

```
#!/bin/bash
recipient=”root”
subject=”Entradas”
`mail -s $subject $recipient < /var/log/lastlog`
```

Este script seria interessante para receber todos os dias às 09:00h da manhã para ver se não houve entradas estranhas na última noite:

```
# crontab -l
0 9 * * * /root/email_lastlog.sh
```

Shell Scripting

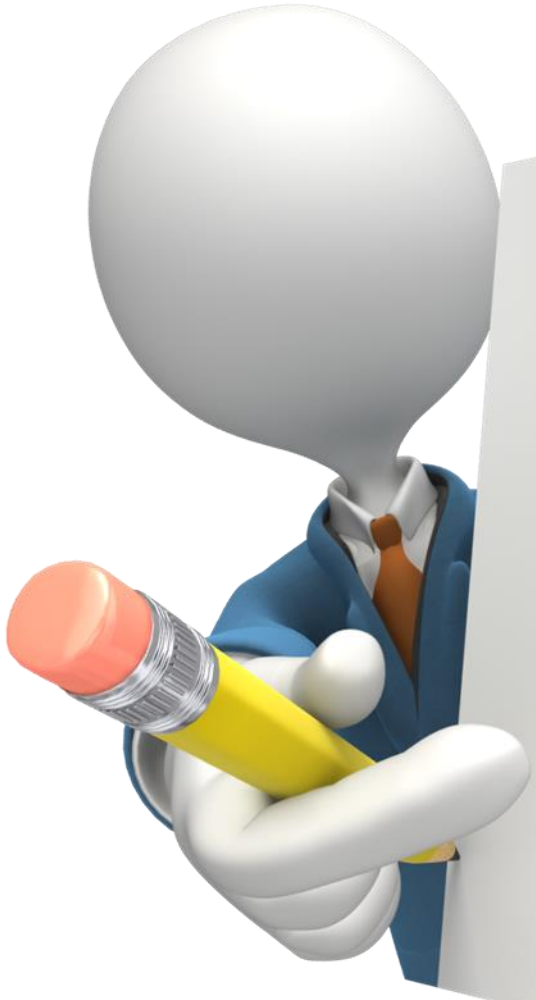
Este limpa os logs, para não ocupar espaço ou ... para limpar o meu rasto ☺ ☺ ☺

```
#!/bin/bash
LOG_DIR=/var/log
cd $LOG_DIR

cat /dev/null > messages
cat /dev/null > lastlog
cat /dev/null > wtmp
echo "Logs cleaned up."
```

Shell Scripting

Um desafio ... Olhar para o script e perceber para que serve ☺ terminando o mesmo indicando o que devemos meter nas frases para substituir ??? E !!!

```
#!/bin/bash

echo -n "Primeiro numero? "
read num1
echo -n "Segundo numero? "
read num2
if [ $num1 -le $num2 ]; then
echo "----  ??? ----"
else
echo "----  !!! ----"
fi
```



Perguntas

1. Qual a shell mais utilizada actualmente no Linux ?
2. Porque dizemos que o vi é um editor de programas se ele é um editor de texto?
3. Qual a razão para dizermos que o nano é mais fácil de utilizar por principiantes?
4. O que é a prompt do CLI ?
5. Diz uma utilização práctica para se utilizar um shell script?