* ssh

**Commands**

* **ls**

(List files)

**Atributos mais usados:** (-h) exibe os dados para leitura humana, ou seja, utilizando KB, MB, GB.

(-l) exibe os arquivos em forma de lista, ou seja, verticalmente

(-a) exibe os arquivos ocultos

* **pwd**

(Together with ls, the pwd command is useful to show where you are in the filesystem)

* **cd**

(change directory)

* **mkdir**

(Creates a new directory in the current working directory)

* **touch** keyborad.txt

(The touch command creates a new file inside the working directory)

* **del** c:\test\test.txt

(Remove arquivo)

* **rm -r** testdirectory

(Remove diretório)

* **cat** /home/userna5/access-logs/example.comcat /home/userna5/access-logs/example.com

(Display the full contents of that file)

**cat** example.com example2.com

(You can also use the **cat** command on two files in a row)

**cat** example\*

(You can use the **\*** asterisk anywhere in the filename, so for instance **cat example\*** would display data from any filename in the current directory that begins with **example**)

$ cat /dir/file.txt | wc –l (Contar linhas em arquivo)

3272485

wc -l : Prints the number of lines in a file.

wc -w : prints the number of words in a file.

wc -c : Displays the count of bytes in a file.

wc -m : prints the count of characters from a file.

wc -L : prints only the length of the longest line in a file.

**cat >> newFile**

(Using **cat >> newFile** will start allowing you to insert data into a file called **newFile**, or if that file already exists append the new data onto the end. After you're done typing in the data to place inside the file, you'll want to hit **Ctrl-D** on your keyboard to write the data to the file)

**cat newFile**

(We can see the new data we just added)

**cat -n newFile**

(Using **cat -n** will allow us to see the line numbers of the file)

**cat -b newFile**

(Using **cat -b** will only count non-blank lines)

**cat -s newFile**

(Using **cat -s** will supress multiple blank lines to a maxium of one)

**tac newFile**

(In reverse)

**tac bigLogFile | grep Error**

(A good example of when the **tac** command shines, is if you had an extremely large log file, and you were looking for some specific text in the log using a command like **grep.** Once you've found enough data, you can hit **Ctrl-C** on your keyboard to stop the **tac** command)

Obs:

$ cat demo\_text

4. Vim Word Navigation

You may want to do several navigation in relation to the words, such as:

\* e - go to the end of the current word.

\* E - go to the end of the current WORD.

\* b - go to the previous (before) word.

\* B - go to the previous (before) WORD.

\* w - go to the next word.

\* W - go to the next WORD.

* **grep -** When you’re searching for a text in a particular file under a Linux system you’re usually using the grep command

**grep** "textToSearchFor" file.txt (basic search)

**grep** -i "THIS CASING DOESN'T MATTER" file.conf

(case-insensitive search)

**grep** "this" demo\_\*

(For this example, let us copy the demo\_file to demo\_file1)

**grep** "REGEX" filename

**grep** "lines.\*empty" demo\_file

(It searches for all the pattern that starts with “lines” and ends with “empty” with anything in-between)

A regular expression may be followed by one of several repetition operators

* ? The preceding item is optional and matched at most once.
* \* The preceding item will be matched zero or more times.
* + The preceding item will be matched one or more times.
* {n} The preceding item is matched exactly n times.
* {n,} The preceding item is matched n or more times.
* {,m} The preceding item is matched at most m times.
* {n,m} The preceding item is matched at least n times, but not more than m times.

**grep** -i "is" demo\_file

Retorno:

this line **is** the 1st lower case line in this file.

Th**is** Line Has All Its First Character Of The Word With Upper Case.

**grep** -iw "is" demo\_file

Retorno:

THIS LINE **IS** THE 1ST UPPER CASE LINE IN THIS FILE.

this line **is** the 1st lower case line in this file.

**grep** -A <N> "string" FILENAME

(Display N lines after match)

**grep** -B <N> "string" FILENAME

(Display N lines before match)

**grep** -r "ramesh" \*

(Search ‘String’ in all the files under the current directory and its sub directory)

**grep** -v "go" demo\_text

(When you want to display the lines which does not matches the given string/pattern)

**grep** -v -e "pattern" -e "pattern"

(display the lines which does not matches all the given pattern)

$ **cat** test-file.txt

a

b

c

d

$ **grep** -v -e "a" -e "b" -e "c" test-file.txt

D

**grep** -c "go" demo\_text

(Counting the number of matches using grep -c)

**grep** -c this demo\_file

(When you want do find out how many lines matches the pattern)

**grep** -v -c this demo\_file

(When you want do find out how many lines that does not match the pattern)

**grep** -l this demo\_\*

demo\_file

demo\_file1

(Display only the file names which matches the given pattern using grep -l)

* **mv** file-one.txt file-two.txt

(rename **file-one.txt** to **file-two.txt**)

**mv** includes/\* ./

(This will move all files (and folders) from the includes/ directory to the current working directory)

**mv** -u includes/\* admin/includes

(update and move only files that were changed)

* **cp** original\_file new\_file

(The command will copy the original\_file file to new\_file and will preserve the original one. The file will NOT be removed after it is copied)

**cp** -R includes/ includes\_backup/

(**-R** instructs cp to copy files recursively, for example, a whole directory)

**cp** -Rf includes/ admin/includes/

(To overwrite already existing files you should use the **-f** argument)

* **df** <atributos>

(Exibe as informações relacionadas ao disco do servidor)

Atributo mais usado: **-f** (Exibi dados em KB, MB, GB)

* **du** <atributos> <arquivo>

(Exibe as informações relacionadas à tamanho do arquivo/diretório)

Atributo mais usado: **-f** (Exibi dados em KB, MB, GB)

**-s** (exibe apenas o total de cada argumento, ou seja, caso utilizado em uma pasta, exibe o somatório de todos os arquivos presentes)

* **tar** <arquivo compactado> <diretório/arquivos para compactar>

(Compactação e Descompactação de arquivos)

Atributo mais usado: (-z) zip

(-x) extrair

(-v) visualizar execução, ou seja, serão exibidos na tela todos os arquivos que estiverem sendo compactados/descompactados

(-c) comprimir

(-f) força a execução da compactação/descompactação

* **wget** <atributos> <endereço>

(Utilizado para realizar o download com uma URL)

Atributo mais usado: (-4) utilização do ipv4 para realizar a conexão

* **zip** <arquivo compactado> <diretório/arquivos para compactar>

(Compacta arquivos)

* **unzip** <arquivo para descompactar>

(Extrai arquivos compactados)

* **ps auxf**

(Lista processos)

* **htop**

(Verifica processos rodando)

* **history**

(Exibe o histórico de comandos executados pelo usuário)

* **vi** <arquivo>

(Abre um arquivo em modo de edição de texto)

* **clear**

(Limpa a tela de comando)

* **find**

(Utilizado para localizar arquivos e trechos de arquivos)