M.U.P.P.I.X. purveyors of fine Data Analysis Tools

Home Training Get Started Linux Cheatsheet I SQL & Excel OneLiners Download

www.muppix.co linux basic navigation commands in the terminal window

TIP: the Muppix command is everything before the '##' in this Toolkit document. This is what you cut/paste & fill in ####! ##

```
pwd                          ## where am I ? full name of mydir, the directory you're in. use file manager/Explorer to find directories
cd c:                        ## + goto C: (mydrive) on windows. TIP: for linux/Apple ,use mount to show all harddrive names
cd mydir                     ## + goto directory called mydir ie: cd mydir TIP: can type beginning few characters & TAB and it will find the fill d
cd ..                        ## + change directory, goto, up 1 directory level
ls -al                       ## + all filesnames (including socalled hidden files) and time stamps in mydir- this directory only filenames & details, t
ls -altr                     ## filenames , sorted by date - what is the most recent file in this mydir directory only
find . -name "*" -ls         ## + select all filenames (& all its size/date information) in mydir & all sub directories, also select each subdirector
cal                          ## calendar , to see when was the directory or file was last saved, amount of days to another date
date                         ## today's date & system time
date -d "-2 months -2 days"  ## what calendar date was (last) 2 months and 2 days ago
cat myfile.txt               ## + select all lines of .txt ie: cat muppix.txt
cat myfile.txt | head        ## begin 10 lines of file TIP: try all your commands on just these 20 lines ie: cat muppix.txt | head -20
cat myfile.txt | tail        ## end 10 lines of file TIP: use this subset of the file to try all your commands ie: cat muppix.txt | tail
find . -exec cat {} \;       ## select all lines of all files in every subdirectory
more                         ## view the results - scroll up/down ie: cat .txt | more
less -I                      ## view & edit the results (see below for less Ctrol keys) ie: cat .txt | less
cut -c-2-88                  ## + delete character(s) before 2, select between character 2 (second) and 88. delete after 88th (fixed)
cut -c-88                    ## + only begin / less than 88 (fixed) characters of each line ie: cat .txt | cut -c-88
wc -lc                       ## + how many lines in the list /  ie: how many mytext found in myfile: cat myfile.txt | fgrep mytext | wc -lc
du .                         ## + directory (mydir) sizes for this directory & all its subdirectories (ie: what's filling up diskspace ??) ie: du . | sort -n
mount                        ## + names of all hard-drives (mydrive) on this version of linux & size. TIP: goto using these harddrive names
df .                         ## + current hard-drive name, size & available
fgrep -i 'mytext'            ## + select lines with 'mytext' anywhere on the line, ignore case. ie: cat .txt | fgrep -i 'mytext'
sed '/./,$!d;s/[ \t]*$//'    ## delete leading/beginning blanklines aswell as ending/trailing blanklines ie: cat .txt | sed '/./,$!d;s/[ \t]*$//'
awk -v OFS=" " '$1=$1'       ## + delete/replace all multiple/duplicate/consecutive spaces with single space/blank, also deletes begin spaces
```

TIP: String together commands using the character : '|' ####! ##

```
sort                         ## sort lines
sort -u                      ## + sort lines and then delete duplicate lines
sort -k2                     ## + sort on the second column
sort -t":" -k2               ## + sort text by second column, ":" is mydelimiter
history 100                  ## history of 100 mycommands I recently used in this terminal
>myfile.txt                  ## + save results to .txt in this directory (TIP: pls note there is no "|" ) ie: ls -al >m yfile
>myspreadsheet.csv           ## + save results to spreadsheet. (result needs column delimiters, such as ';', but best is "|" as delimiter)
u2d                          ## TIP: may need to run unix2dos or u2d , before looking at the file in Windows say notepad
diff -w myfile mysecondfile  ## + select differences in 2 files, but ignore differences of extra spaces or tabs (white space) TIP: "<" in the o
```

tools to help view the output

back

TIP: use these commands to temporarily view the output ####!   ##

```
cat myfile.txt                   ## select all lines in .txt
grep --color=auto -inHR 'mytext' *   ## + select/show in fancy colours/colors and linenumbers, 'mytext'
grep --color mytext *            ## select/show in fancy colours/colors and linenumbers, 'mytext'
head                             ## select beginning 10 lines of file ie: cat | head | someMuppixcommand
head -2                          ## select beginning (fixed) and second lines
cut -c-2                         ## + only select beginning and second characters of each line. ie: cut -c-77 to quickly view text, beginning 77 character
sort -u                          ## delete duplicates, just the unique list
more                             ##
less -I                          ## view file of ANY Size & quickly move up and down & search (ignore case) ie: grep 'mytext' | less -I
-- spacebar                      ## next page
-- PageUp, PageDown              ##
-- <                             ## top of file
-- >                             ## bottom of file
-- /mytext                       ## search mytext forwards (ignore case)
-- /^mytext                      ## search lines beginning with 'mytext'
-- /mytext$                      ## search lines ending with 'mytext'
-- /myt[ex]                      ## search 'myte' or 'mytx'
-- ?mytext                       ## search mytext backwards
-- n                             ## next match Forwards
-- N                             ## next match backwards
-- 999                           ## goto line 999 line
-- q                             ## Quit
-- CTRL z                        ## Quit
tail -2                          ## select (fixed) end line and second from end line ie: tail -100 , end 100 lines
tail -f myfile                   ## tails , selects end lines of the file & continues to select new updates if more lines are added . ie: a live log file TIP:
tail -300f myfile                ## tails, initially select the end 300 lines , then select new updates
tail -f file | grep 'mytext'     ## tail the file, but only select lines with 'mytext' in it. very usefull for tailing changing log files
history 100                      ## history of 100 mycommands I recently used in this terminal
history 100 | grep 'mytext'      ## out of the 100 recent mycommands, select those with 'mytext'
time mycommand                   ## See how long a command takes to execute ie: time fgrep 'mytext'
paste -s -d ' ' | xargs -n 2     ## convert list of words to 2 columns TIP:if its a Windows file, shld run dos2unix
nl -ba                           ## insert linenumbers at the beginning of each line
cat -n                           ## insert linenumbers at beginning of each line ie: find out linenumbers with 'mytext' : cat .txt| cat -n |fgrep 'mytext'
fmt myfile                       ## format/left align all the text into 80 chars wide
```

terminal keystrokes in linux window

```
UParrow        ## previous Linux command
DOWNarrow      ## next Linux command
HOME           ## cursor to beginning of line
END            ## cursor to end of line
CTRL r         ## reverse command , enter what to search on , CTRL r again , goes to previous command#
CTRL k         ## delete rest of line
CTRL u         ## delete everything to the beginning
CTRL d         ## delete current character
CTRL c         ## STOP / Exit ie: cat mymassivefile & then CTRL c or CTRL z
CTRL z         ## STOP / Exit
CTRL l         ## clean line/statement. use when the command & cursor get confused
history        ## mycommands
cut n paste    ## right click on blue banner of the Unix window, Edit , Paste or Mark
```

The Muppix Team provides innovative solutions and Training to make sense of large scale data Backed by years of industry experience, the Muppix Team have developed a Free Unix Data Science Toolkit to extract and analyse multi-structured information from diverse data sources

Company

Training

Professional Services

Get Started

Blog