Current working directory (cwd) also pwd

root: /

under root other directories

/home/kbeine

~kbeine = /home/kbeine

mkdir = make directory

cd = change directory, this is how to get into a file

“touch” creates files

list displays files in directories

rm deletes

rmdir deletes

ls -lu lists read/write permissions for files

ls – lists all under directory

rm -r removes all things under a directory even if there's stuff in it.

Ctrl-c terminates programs

rm -rf instantaneously deletes whole directory.

Cp – copies

less looks at contents of file

mv moves documents

cp -r copies directories

cat shows contents of files

nano lets you edit a file

.. indicates directory above, . Indicates current directory

ln creates links or something like that – I can't actually figure out how to link stuffffffffff

Ls -il shows info about file?

AHA.

So to link you can't use a file name/location that already exists.

i.e. ln new-file one-file will only work if one-file doesn't already exist.

When I do ls -a I can see the dots. But I don't know why...

nano (file name)/ you can change the file name

Annnd...how do you unlink stuff?

PERMISSIONS

3 agents: User Group Other (UGO)

3 permissions Read Write Execute (RWX)

read can list files/directories in a directory

write can create/delete

chmod – change permissions

chmod o-w takes away write permissions

chmod o+w

o = other u = user

w =write r=read

execute stops you from changing into a directory, i.e. cd won't work.

Base eight: one octal digit is three binary digits

first digit is r, second is w, third is x

e.g. 755

LAMP – Linux, Apache, My SQL, Php (Pearl)

find: to find files; “find (starting directory) -name”

e.g. “find . -name new-file”

For days : -mtime +7 -mtime -14

equals older than seven days, newer than 14 days.

Mtime= modification time

atime= access time

-mtime +365 -atime +365

breadth and depth: breadth goes across directories (looks at contents of directories first)

depth: goes to lowest directory and works up.

-o = or

grep – looks inside files for stuff

G – Global, look on every line

RE – Use regular expression

P – print

. matches one character

grep “.\*e.\*x” /usr/share/dict/words

^ zero length regular expression, matches to beginning of line

.. makes it put at least one character between the two letters.

$ matches to end of line

[ ] you can search for multiple letters

-i is case insensitive

[a-l] gives range

| wc = word count

\bword\b matches word exactly

{5} matches exactly 5 times

^C stops stuff

single quotes makes it read directly, if telling you weird stuff, try quotes.

Q stops less running

9/29/13

REDIRECTION:

Takes what is on the screen and puts it into a file

Process – Three connections: Standard out(stdout) Standard Error(stderr) Standard input (stdin)

1 2 0

Redirection: when you do a find: find . -name “\*.pdf” > (name of file)

e.g. find . -name “\*.docx” > worddocs

to overwrite, type the file name again, to append, do double arrow >>

|--> “Clobber”

doing a find in /etc (on mathserv2)

To redirect standard error: use 2> instead of >

/dev/null – file that immediately deletes stuff sent to it. If you want to find something and not see the standard error output, send 2> /dev/null

>& sends standard output and error to the same place

find . -name “\*.conf” 2>&1 > conf-files.txt

Redirecting standard input: use < with grep, etc.

e.g. grep spear “return” blank line

e.g. grep spear < /usr/share/dict/words

“hear” documents: grep spear <<EOF

>typing on line

> more typing

>spearmin

>EOF

The three lines before EOF become standard input

PIPELINING

Puts output of process one to be input of process two

e.g. find . -name “\*.pdf” | grep color

to pipeline stuff you put | after first process

tr changes 'aeio' to 'x', can take ranges, a-l

\n makes new line

e.g. tr 'aeiou' 'AEIOU' < /home/share/andrews/iliad-buckley.txt | less

uniq removes repetitions

-n sorts in numeric order -nr sorts in reverse numeric order

e.g. tr -dc 'a-zA-Z09 \n' < /home/share/andrews/iliad-buckley.txt | tr ' ' '\n' | sort | uniq -c | sort -nr | less

| tail (or head) -n -15 gives you the last 15 lines (or first 15 lines) tail -n +15 starts on fifteenth line

“gnome-open” will open a document in the right application e.g .docx files

c means complement as in opposite so -d 'a-z' deletes a-z but -cd 'a-z' deletes complement

PUBLIC KEY ENCRIPTION

ssh-keygen

HOW DOES THE INTERNET WORK

IPv4 – IPv6

Packet-switched Network – information sent in packets, sent by different routes

IP (Internet packet) Destination: Dotted quad e.g. 192.168.1.1 (sent in 4 bytes, a byte is 8 bits) numbers can be 0-255, but first quad doesn't start with 0

Source:

Ports: 16 bytes port 80 is official port for internet connections, 22 is for ssh connections

If DNS goes down, you can't access the internet with names, but you can do it with IP addresses.

Google: 173.194.46.16-20

udallas.edu: 50.56.227.17

wikipedia: 208.80.154.225

live.com: 65.55.206.154

facebook: 173.252.110.27

minastirith.com: 98.139.135.125

MAIL:

telnet mail.udallas.edu smtp

HELO (name)

MAIL FROM (name)

RCPT TO (name)

DATA

(type message)

(to end, type period enter)

http://dal.ent.sirsi.net/client/default

Alias mv

alias mv='mv -i'

touch one-file two-file

mv one-file two-file

history

edit .bashrc – be careful. Make a backup before editing

can copy the lines into the terminal and edit there

PS1=

changes dollar sign to whatever you want

033 – octal 33 is escape key

when writing long lines, use \+enter to go to next line of same command

PROMPT\_COMMAND runs every time it gives you a prompt

echo $USER – to find out user name

# Bash Variables

noclobber=on

cdable\_vars=on

shopt – shell options

shopt -s \_\_\_\_\_ – to turn on options. -u to turn off

.profile

umask – permissions when you create a file after the command

umask 222 – all read

two kinds of variables

* shell – variables contained only in the specific shell.

xterm – opens new shell, descendant, with shell variables

bash/exit

ed – line editor

ex – extended editor

vi – visual editor

move mode: hjkl <^v>

ctrlf, page down

ctrlb, page up

w forward by word

b backward by word

$ end of the line

^ beginning of the line

f “character” jumps to the first instance of that character in the line

G goes to end of file, 1G goes to first line

insert mode: type i to insert before the cursor, type a to insert after cursor

type o – opens new line to type

O opens new line above

u in move mode will undo typing from insert

. in move mode repeats typing action in spot where your cursor is.

Copy/paste:

yy yanks current line

p inserts yanked line below current line

10 yy yanks next ten lines (starting on current line)

y with any of the move commands yanks specific chunks

d deletes, i.e. cuts

also d with any of the move commands

: begins extended commands

:s substitutes – s/a/x/g – changes all a's into x's on current line

:1,$ includes whole file

:g/a/s/a/x/g – global, all lines with an a, substitute x for a to end of line

:g/[aeiou]/x/s//XX/g

// refers to what was searched for

:g/run/d deletes all lines with string “run”

:10Jj joins ten lines into one line

:map @ 10Jj defines @ as 10Jj, so when you hit @ it executes 10 Jj

ctrlg tells you where you are and gives line, word and percent

:qi quits without writing

:wq writes and quits

ZZ writes and quits

vim – extended vi