cat (takes files (or stdin) as input and copies them, sending result to stdout

cat >filename (creates a file)

cat filename (shows you what is in file)

cat file1 file2 >file3 (creates file3 and puts contents of file1 and 2 into it)

mkdir x (make directory named x) rmdir x (removes directory named x)

rm f (removes files named f) more (display files page by page) date (display current date) id (display information about current user) # (comment 🡪 comp will ignore anything after it

script (record a terminal session in a file named \_\_\_) 🡪 script yyy names the file yyy

mkdir –P x/y/z

path x/y/z/z (last z is a file created) – real application is to update a file (time)

touch c/b/{d,e}

rm –r r (erases everything (files + dir); -r is recursive cmd; r at end is name of top lvl dir

ls –FAR (recursive listing); how to go to top of tree structure and look down

-C (forces to list everything out in a row/line)

-1 (forces everything to be listed own)

who (who else is online)

wc (word count 🡪 count lines (wc -l), words (wc -w), and bytes (wc -c). default is –lwc

rmbs < typescript > typescript1

sftp (secure file transfer protocol) 🡪 sftp [user@]host connect to host as user

sftp cmds:

ls, pwd, mkdir, cd operate on remote system

use 1 prefix to each cmd to operate on the local system (eg. 1pwd)

get x (transfers file x from current directory on the remote system to the current dir on local system)

put x (transfers file x from current dir on local system to current dir on remote system)

If I am on Linux; want to copy script from current dir to hills; placing it in ~ (use put)

$ sftp hills.ccsf.edu

Pw

Sftp> ls (remote) vs lls (local)

Sftp> put typescript

Msg

Sftp> pwd (remote) vs lpwd (local)

If on hills; T/F file to linux

$sftp IPaddress

Pw

Sftp>pwd

Sftp> (from dir your in)

Sftp> put (filename)

scp (secure copy program. Execute a single copy cmd between two hosts

scp xxx [user@]hostx:[path] copy xxx from the current host to hostx, logging in as user, and placing the result at path. If user is missing, the current login is used. If path is missing, the user’s home dir is used. \*\*\*\*

$scp

Copy local 🡪 remote

$scp (path local) user@host:(path remote – relative to home dir)

Eg. scp typescript hills.ccsf.edu:typescript1

Copy from remote 🡪 local

$scp user@host:(path remote) (path local)

If copying a dir: (from local to remote)

$scp –r (path local) user@host:(path remote)

Eg. scp –r (dir) (remote):dir

If copying a dir: from remote to local

$scp –r user@host:(path remote) (path local)

A\* capital A followed by 0 or more char

? single char

[] single character that is one of the characters in the brackets.

[a-z][A-Z][0-9]

[[:alpha:]] upper, lower, digit, alnm, blank, space(whitespace), punct

[!a-z] a single character whose value is not greater than or equal to a and less than or equal to z

? and \* will not match a leading . in a name (indicating a hidden file)

NOTE: you can’t copy a file on top of a dir or or a dir on top of a file

cp file1 file2 🡪 copy file1 to file2. File2 is overwritten if it exists.

cp file1 file2 file3 fileN dir 🡪 copy one of more files into an existing dir

cp –r dir1 dir2 🡪 copy dir1 and all of its contents to dir2. If dir2 does not exit, the new copy of dir1 is named dir2. If dir2 exists, the copy of dir1 is placed in dir2.

cp –r dir1….dirN dir2 🡪 copy one or more directories and their contents into the existing dir2

mv file1 file2 🡪 file2 is deleted if it exists. Then file1 is renamed file2

mv file or dir1….fileordirN dir 🡪 move one or more existing files or dir into an existing dir. If a dir is moved, everything is moved, everything beneath it goes with it.

mv dir1 dir2 🡪 if dir2 does not exist, rename dir1 to dir2. If dir2 exists, dir1 is moved into dir2

Variables: (to create a variable: $variable= ) 🡪 to use type $ $variable?

$echo $PWD (absolute path to current dir) 🡪 ie. /students/vmalchik

$echo $SHELL (your shell) /usr/bin/bash

$echo $LOGNAME (your login) 🡪 vmalchik

$echo $HOME (absolute pathway to home) 🡪 /students/vmalchik

$echo $PATH (a list of dir searched to locate cmds)

$echo $TERM 🡪? vt100 (vt100 is language used by terminal)

$echo $HISTSIZE (size of history – set to 500)

Permissions:

Owners – each piece of data on unix has one owner (whomever created it)

Groups – users on the system are organized into groups

* Most users in may groups (each piece of data is in one group)
* One of my groups is my default group (the one that data goes into)

u - if you are the owner

g - I am not the owner but a member of the data’s group

o -I am not the owner nor do I belong to a data’s group

-(file or d for dir)|r –x (u)|rwx (g)|r - - (o) 1 (ownder) (group #) ….. file1

R – read (ie cat the file)

W – write (ie change file content)

X – execute (ie. attempt to execute it as a program)