$nl (adds line # but not to non-blanks) –ba (adds # to blanks)

$cat –n (all lines numbered)

$head –n  # (output the first # numbers)

$tail –n # (output the last # numbers of the file) eg. $tail –n 10 file

$tail –n +# (start from line #) eg. $tail –n +90 (starts at line 90 and goes to end of file)

--$fold  -w # (wrap at # columns) –s (wrap at nearest space) eg. $fold –s –w 60 filename

$wc (has –l, -w, -c)  by default has –lwc  (lines, words, bytes)

Note: $wc –l < filename (gives u number of lines in file without file name)  or $wc –c \*

--$cut –d’ ‘ –f#, f-f (eg $cut –d’#’ –f1,6-4 >  tmp/file) \*\* default is tab \*\* note: -f5,9- the – gets filenames with blanks but has a problem with symlinks

--$cut –c#,#-# filename (eg $cut –c4,6-10 file1 > temp/file) note; if you don’t close the field it goes to end-c #-

$paste –d’ ‘ filefrom fileto (white space is default delimiter)

(Note: for $tr you need to use < > ) \*\* substitute / delete / squeeze

$tr ‘string’ ‘string2’ *(substitute*) filename (eg. ‘[a-z]’ ‘[A-Z]’ change to uppe

$tr –d ‘string’ < file (each char is *deleted*) > file1

Str –s ‘sting’ file (each successive duplicate is removed/squeezed) eg –s’ ‘ (removes spaces)

--$grep “pattern”  filename \*\*\*find pattern\*\*\*

-i (ignore case); -v (don’t contain pattern); -c (count number of lines that contain pattern);

-l (output the name of the files that contain at least one match)

- \*\*or \*\* -e (grep –e “pattern” –e “pattern” … (file name)

\*\*and\*\* grep “pattern” file | grep “pattern2”

^start end$

$ln –s path nameOflink (eg. $ln –s ~/tmp/file file1;  $ln –s ./resume resume1)  can point to dir or file

$ln abspath name (hard link)  can only point to file

$ls –l (see where the symlink points to) $ls –lL see the permissions of the symlink (which is the file/ dir it points to)

$ls –i (inode number)

Abs link can be anywhere for it will lead you to the place where as relative link has to be in the same dir\*

$sort (-n for numeric; -r for reverse; ) \*\* white space is default delimiter (organize)

$sort -f (ignore case)

-o outfile into outfile (ie. sort –o fileN –t’#’ –k2,2 –k1 fileN !!

-t ‘delimiter’ -k’field’ (ie. k1,1 or 1.2nr)

-u (sorts and removes duplicate fields)

$sort file | uniq = sort –u note u can also $uniq < file

$wc (has –l, -w, -c)  by default has –lwc  (lines, words, bytes)