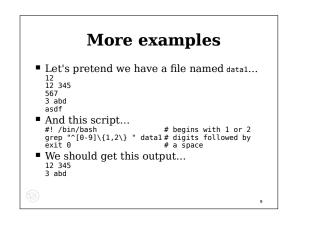
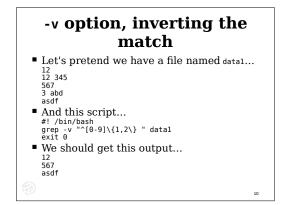
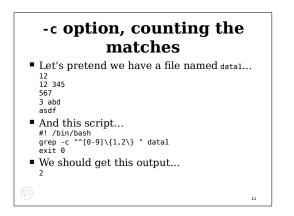


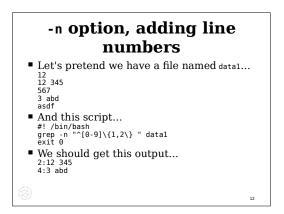
Examples

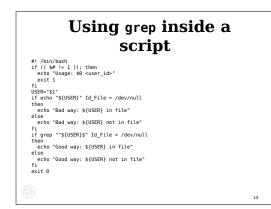
- grep "^string\$" file_name collects all lines which contain only string
 grep " ... " file_name
- collects all lines which have any three characters surrounded by spaces
- grep " [0-9]\{1,3\} " file_name collects all lines containing a sequence of one to three digits surrounded by spaces
 grep "^x*[abc]" file_name
- collects all lines which start with zero or more x's followed by a single a, b, or c

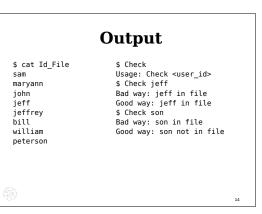










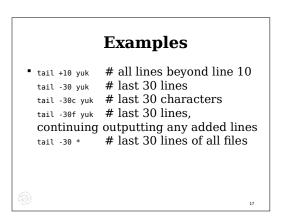


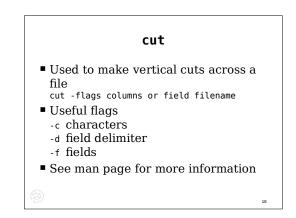
head

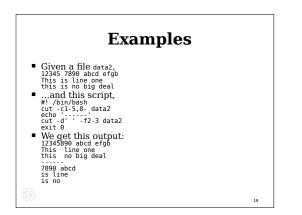
- Collects the first n lines of a file with n defaulting to 10 if unspecified head [-n] file
- Examples... head -30 yuk # top 30 lines head yuk # top 10 lines head * # top 10 lines of every file

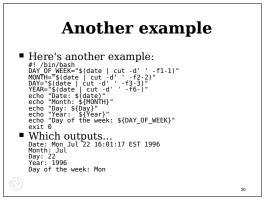
15

tail
 tail [+/-[n] [b[c]l] [-f]] file delivers n units from the file +n counting from the top -n counting from the end n defaults to -10 if unspecified counting by b blocks c characters l lines (default) -f means follow - infinite trailing output (use ctrl-c to stop) Has buffer limitations - see the man page
16











- Used to combine lines from two files together paste [-dlist] file1 file2 ...
- By default concatenates corresponding lines of the files together using a tab as the separator

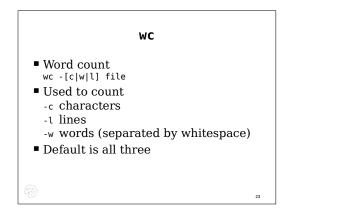
Example: paste -d" " x y z concatenates the corresponding lines of the files x, y, and z together using the list of separators circularly. In this case the list only contains a single space.

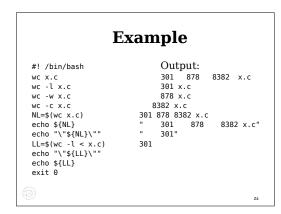


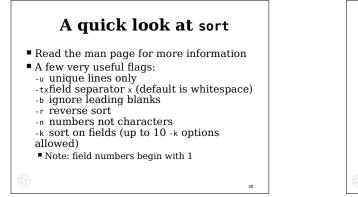
- paste -s [-d list] file1 file2 ... merges lines together serially (one file at a time)
- paste -s -d" \n" yuk
 pastes each pair of lines in the file yuk
 together
 - the list specified with -d is a space followed by a newline

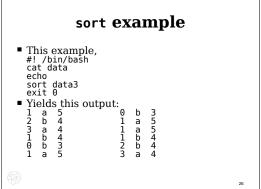
22

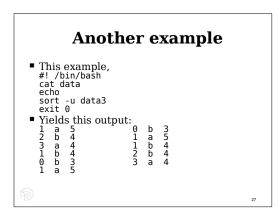
See man page for more options and information

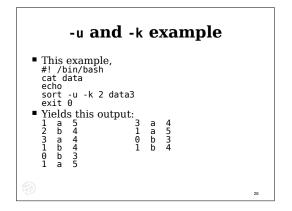


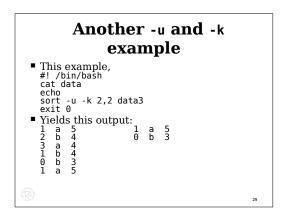


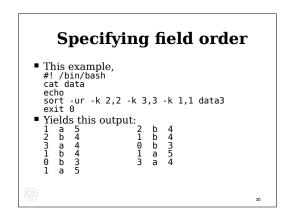












Beating the dead horse			
 This example, #! /bin/bash cat data echo sort -k 3,3 -k 1,1 -k 2,2 data3 echo sort -k 3bn,3 -k 1bn,1 -k 2b,2 data3 exit 0 Yields this output: 1 a 5 3 a 14 11 b 4 11 b 4 11 b 4 1 a 5 12 c 40 12 c 40 3 a 14 2 a 40 2 a 40 2 a 40 21 c 51 1 a 5 12 c 40 			
21 c 51 1 a 5 12 c 40 3 a 14 21 c 51 21 c 51			
	31		

