To redirect both standard output (stdout) and standard error (stderr) to one file, first redirect stdout to a file and then redirect stderr to that same file by using the notation 2>&1

**sysadmin@localhost:~$** find /etc -name hosts > find.out 2>&1

**sysadmin@localhost:~$** cat find.out

/etc/hosts

find: '/etc/ssl/private': Permission denied

**sysadmin@localhost:~$**

The tr command accepts keyboard input (stdin), translates the characters and then redirects the output to stdout. To create a file of all lower-case characters, execute the following:

**sysadmin@localhost:~$** tr A-Z a-z > myfile

Wow, I SEE NOW

This WORKS!

Execute the following commands to use the tr command by redirecting stdin from a file:

**sysadmin@localhost:~$** cat myfile

wow, i see now

this works!

**sysadmin@localhost:~$** tr a-z A-Z < myfile

WOW, I SEE NOW

THIS WORKS!

**sysadmin@localhost:~$**

. Execute the following command to take the output of the ls command and send it into the more command, which displays one page of data at a time:

ls -l /etc | more

total 372

-rw-r--r-- 1 root root 2981 Jan 28 2015 adduser.conf

-rw-r--r-- 1 root root 10 Jan 28 2015 adjtime

drwxr-xr-x 1 root root 900 Jan 29 2015 alternatives

drwxr-xr-x 1 root root 114 Jan 29 2015 apparmor.d

drwxr-xr-x 1 root root 168 Oct 1 2014 apt

-rw-r--r-- 1 root root 2076 Apr 3 2012 bash.bashrc

drwxr-xr-x 1 root root 72 Jan 28 2015 bash\_completion.d

drwxr-sr-x 1 root bind 342 Jan 29 2015 bind

-rw-r--r-- 1 root root 356 Apr 19 2012 bindresvport.blacklist

-rw-r--r-- 1 root root 321 Mar 30 2012 blkid.conf

command called cut to extract all of the usernames from a database called /etc/passwd (a file that contains user account information). First, try running the cut command by itself:

**sysadmin@localhost:~$** cut -d: -f1 /etc/passwd

root

daemon

bin

sys

sync

games

sort command to provide some order to the output:

**sysadmin@localhost:~$** cut -d: -f1 /etc/passwd | sort

backup

bin

bind

daemon

games

gnats

irc

libuuid

list

lp

Now the output is sorted, but it still scrolls off the screen. Send the output of the sort command to the more command to solve this problem:

**sysadmin@localhost:~$** cut -d: -f1 /etc/passwd | sort | more

backup

bin

bind

daemon

games

The /etc/passwd is likely too large to be displayed on the screen without scrolling the screen. To see a demonstration of this, use the cat command to display the entire contents of the /etc/pass

wdfile:

**sysadmin@localhost:~$** cat /etc/passwd

Use the more command to display the entire contents of the /etc/passwd file:

**sysadmin@localhost:~$** more /etc/passwd

Use the less command to display the entire contents of the /etc/passwd file. Then search for the word bin, use **n** to move forward, and **N** to move backwards. Finally, quit the less pager by typing the letter **q**:

less /etc/passwd

/bin

nnnNNNq

You can use the head command to display the top part of a file. By default, the head command will display the first ten lines of the file:

head /etc/passwd

Use the tail command to display the last ten lines of the /etc/passwd file:

tail /etc/passwd

Execute the following command line to pipe the output of the ls command to the tail command, displaying the last five file names in the /etc directory:

ls /etc | tail -5

Another way to specify how many lines to output with the head command is to use the option -n -#, where # is the number of lines counted from the bottom of the output to exclude. Notice the minus symbol - in front of the #. For example, if the /etc/passwd contains 27 lines, the following command will display lines 1-7, excluding the last twenty lines:

head -n -20 /etc/passwd

The use of grep in its simplest form is to search for a given string of characters, such as sshd in the /etc/passwd file. The grep command will print the entire line containing the match:

**sysadmin@localhost:~$** cd /etc

**sysadmin@localhost:/etc$** grep sshd passwd

**sshd**:x:106:65534::/run/**sshd**:/usr/sbin/nologin

**sysadmin@localhost:/etc$**

To limit the output, you can use regular expressions to specify a more precise pattern. For example, the caret ^ character can be used to match a pattern at the beginning of a line; so, when you execute the following command line, only lines that begin with root should be matched and displayed:

**sysadmin@localhost:/etc$** grep '^root' passwd

**root**:x:0:0:root:/root:/bin/bash

**sysadmin@localhost:/etc$**

Match the pattern sync anywhere on a line:

grep 'sync' passwd

Use the $ symbol to match the pattern sync at the end of a line:

grep 'sync$' passwd

Use the period character . to match any single character. For example, execute the following command to match any character followed by a 'y':

grep '.y' passwd

The pipe character, |, or "alternation operator", acts as an "or" operator. For example, execute the following to attempt to match either sshd, root or operator:

grep 'sshd|root|operator' passwd

Use the -E switch to allow grep to operate in extended mode in order to recognize the alternation operator:

grep -E 'sshd|root|operator' passwd

Use another extended regular expression, this time with egrep with alternation in a group to match a pattern. The strings nob and non will match:

egrep 'no(b|n)' passwd

The [ ] characters can also be used to match a single character. However, unlike the period character ., the [ ] characters are used to specify exactly what character you want to match. For example, if you want to match a numeric character, you can specify [0-9]. Execute the following command for a demonstration:

head passwd | grep '[0-9]'

Suppose you want to search for a pattern containing a sequence of three digits. You can use { } characters with a number to express that you want to repeat a pattern a specific number of times; for example: {3}. The use of the numeric qualifier requires the extended mode of grep:

grep -E '[0-9]{3}' passwd