SED, AWK and basics of BASH scripting

Krzysztof Kluza

03.01.2022

1 SED

1. Using `sed` display the following lines of the file (e.g. `/etc/passwd`):
   (a) 4th, 7th, 10th and 13th lines,
   (b) the lines specified by the interval (e.g. from 3rd to 5th inclusive),
   (c) lines describing people with a login starting with `'z'`,
   (d) lines describing people with a login starting with `'w'` or `'z'`,
   (e) only non empty lines,
   (f) only lines containing digits.
2. Using `sed` how to:
   (a) replace the separator (colon) in the `/etc/passwd` file with a space,
   (b) display only the logins of users saved in the `/etc/passwd` file,
   (c) put a comment character in the `link-local` line in the `/etc/networks` [hint: try can use `&` in your command],
   (d) insert an X column after the first character of each line (an additional X character in each line),
   (e) insert an X column after the fifth character of each line,
   (f) repeat the first two letters of each line in a file 3 times,
   (g) change the order of words (i.e. in `/etc/aliases` change columns, e.g. `postmaster: root` to `root: postmaster`) [hint: define regions].
   (h) replace letters to encode a file with ROT13 (convert the letters into letters that are 13 letters further along in the alphabet, e.g. $a \rightarrow n$).
3. * How to imitate (emulate) other commands with `sed`:
   (a) `grep -v`, e.g. for the phrase `'lo'`? (`grep -v lo /etc/networks`)
   (b) `tr 'a-z' 'A-Z'`, e.g. `cat /etc/networks | tr 'a-z' 'A-Z'`
   (c) `cut -d: -f2`

2 AWK

Check the awk built-in variables and syntax of awk. Then, using `awk`:

1. print out a list of user ids and usernames from `/etc/passwd`, e.g. `0-root`,
2. print out the user id for the particular user (e.g `root`),
3. print only lines which contain `100`,
4. print only lines which contain digits,
5. print line numbers before the content of each line,
6. print the first 5 lines,
7. print the last line,
8. print only line number 80,
9. print only lines between 80 and 85,
10. count the number of users (number of lines in `/etc/passwd`),
11. count the number of blank lines in a file,
12. sum up the user ids of all users,

3 Basics of BASH scripting

The first line of the script should have a form: `#!/path/interpreter options`, which for bash is usually: `#!/bin/bash`.

1. Using `which` or `whereis` or `type` determine the path to bash shell in the system. What file mode (permission rights) should a script have set?
2. Write a bash script which displays the current date, as well as the following information: name of the system, name of the host, how long system has been running and how many users are logged in.
3. Write a bash script which displays the list of logged in users and session list of the person starting the script.
4. Write a bash script which displays the access rights to the file specified as a parameter for the script from the command line. If the file owner is logged in, the script should also display more information about them and their open sessions.
5. Write a script that archives (copies to a specific directory) all files in your home (user) directory that have been modified within the last 24 hours.
6. Write a script that displays all files from your home directory larger than 100kB. The script should write the name of the file and current time of size checking to the separate log file.