L I N U X C O M M A N D S

Get user info from /etc/passwd and change ownership of user's home directory (select userid higher than 1000)

a) View /etc/passwd file
b) Print the 1st field from /etc/passwd file
c) Print all userids > 1000
d) Print the 2nd field to get home directory
e) Use command substitution to get user list and home directory
f) Change ownership of above home directory with user which is retrieve above
g) Iterate above steps for all userid > 1000

E.g /etc/passwd =>
ashishv:x:1002:1002::/home/ashishv:/bin/bash
extract values mark in bold - user and home directory
Expected output -
/home/ashishv:
drwx------ 6 ashishv ashishv 4096 Aug 6 12:48 ashishv

Move files from one folder to the respective folders.

E.g current folder have files abc.txt, def.txt, ghi.txt, jkl.txt
You have to move these files to the folder like abc.txt => abc/ , def.txt => def/ ...
Expected outcome -
abc/abc.txt
def/def.txt
ghi/ghi.txt
jkl/jkl.txt

a) Create files in current directory or any temporary directory - abc.txt, def.txt, ghi.txt, jkl.txt
b) Print list of files to move.
c) Segregate basename and extension of a file.
d) Create folder using basename.
e) Move file to newly created folder.
f) Iterate above steps for all files.

Append current date to all log files name which has extension .log.1 from a folder

E.g original file - access.log.1
New updated file name - access-20102019.log

a) Create files with name abc.log.1, def.log.1 , ghi.log.1, jkl.log.1, mno.log.1
b) Print list of files to rename.
c) Segregate basename and extension of a file
d) Print Date Command to show in ddmmyy
e) Append Date to the log file name
f) Iterate above steps for all files which has extension .log.1

Archive the files from /var/log folder which have modified 7 days ago and move it to your backup folder

a) Identify files which have modified time greater than 7 days
b) Move these files to the backup folder

Print last 4 frequently access urls count in sorted order from /var/log/httpd/access.log

a) View /var/log/httpd/access.log
b) Print field which has urls data.
c) Sort extracted urls and count it
d) Print 4 unique urls
Expect sample output -
3458 /index.html
300 /api/swagger-ui.html
100 /favi.ico
20 /robots.txt

Print list of last 4 frequently access unique urls at particular hours from /var/log/httpd/access.log

a) View access.log without opening it using editor.
b) Print urls which has given timestamp.
c) Sort extracted urls and count it
d) Print 4 unique urls

Expect sample output -
3458 /index.html
300 /api/swagger-ui.html
100 /favi.ico
20 /robots.txt

Print list of web response code count in the unique sorted order at specific hours

a) View access.log without opening it using editor.
b) Print web response code field which has given timestamp
c) Sort extracted response code and count it
d) Print 4 unique response code count

Expected sample output -
1000 200
100 304

Print list of last 10 unique sorted client IP from /var/log/httpd/access.log

a) View access.log without opening it using editor.
b) Print client ip field from access log
c) Sort extracted client IP and count it
d) Print 4 unique client IPs
Expect sample output -
3635 107.181.177.135
423 27.62.203.44
45 157.44.195.138
4 157.39.158.225

L I N U X C O M M A N D S

Check if a folder exists or not. If it's not present, create it

a) Test if particular folder exists in current directory or not
b) If its doesn't exists then create it else print "folder already exists.."

Execute command "hello" and "ls" and check its execution status and print whether command executed successful or not.

a) Execute "hello" command at command prompt
b) Check execution status of "hello" command
c) Execute "ls" command at command prompt
d) Check execution status of "ls" command

Set environment usersecret="dH34xJaa23" if its already not set

a) Check whether environment variable usersecret assigned any value or not
b) Print error if usersecret already set
c) Set environment variable usersecret to given value.

Find a word "systemd" from all log files in the folder /var/log and print number of occurrence more than 0 against each file.

a) Use linux command to search word and print occurrence

Create process list table displays process id, parent process id, command name, % of memory consumption, % of cpu utilization

PID	PPID	CMD	%MEM	%CPU
760	1	/usr/bin/dockerd -H unix://	3.5	0.0
776	1	/usr/bin/containerd	0.7	0.1
7266	757	sshd: root@pts/0	0.6	0.0
759	1	/usr/sbin/rsyslogd -n	0.5	0.0
347	1	/usr/lib/systemd/systemd-jo	0.3	0.0
484	1	/usr/sbin/NetworkManager	0.3	0.0
1	0	/usr/lib/systemd/systemd	0.2	0.0
7268	7266	-bash	0.2	0.0
758	1	/usr/bin/python -Es /usr/sb	0.1	0.0

Data analysis / manipulation (Awk)

Id	Employee Name	Job Title	Base Pay	Overtime Pay	Other Pay	Total Pay	TotalPayBenefits
1	NATHANIEL	GM	167411	0	400184	567595	567595
2	GARY	CAPTAIN	155966	245131	137811	538909	538909
3	ALBERT	CAPTAIN	212739	106088	16452	335279	335279
4	CHRISTOPHER	MECHANIC	77916	56120	198306	332343	32343
5	PATRICK	DEPUTY CHIEF	134401	9737	182234	326373	326373
6	DAVID	ASST DEPUTY	118602	8601	189082	316285	316285
7	ALSON	BATTALION CHIEF	92492	89062	134426	315981	315981
8	DAVID	DEPUTY DIRECTOR	256576	0	51322	307899	307899
10	JOANNE	CHIEF	285262	0	17115	302377	302377
12	PATRICIA	CAPTAIN	99722	87082	110804	297608	297608
13	EDWARD	EXECUTIVE	294580	0	0	294580	294580

i) Print EmployeeName and TotalPay who has BasePay greater than 10000

a) Read data file 'data.csv' from command line and extract rows which have BasePay > 10000
b) Print only EmployeeName and TotalPay

ii) What is the aggregate TotalPay of employees whose jobtitle is 'CAPTAIN'

a) Read data file 'data.csv' from command line and extract rows which have 'CAPTAIN' in the column 'jobtitle'
b) Extract TotalPay and calculate sum. Print the result on terminal.

iii) Print JobTitle and Overtimepay who has Overtimepay is between 7000 and 10000

a) Read data file 'data.csv' from command line and extract jobtitle and overtimepay for column value range between 7000-10000
b) Print the result on terminal.

iv) Print average BasePay

a)Read data file 'data.csv' from command line and extract BasePay values and calculate its average
b)Print the result on terminal.

Find the difference between original file and the updated file. Apply changes to the original file.

a) Create two directories as "original' and "updated"
b) Copy given file 'original-file.sh' to the folder "original" and "updated-file.sh" to the folder "updated"
c) Find the difference between these directories using linux command
d) Make copy of folder "original" to some other directory as "original-backup" and apply changes to 'original-file.sh' file
e) Verify that both folders "updated" and "original-backup" have no difference.