RHCSA EX200 - Input/Output Redirection LAB

Use Input/Output Redirection

Create a Server Health Log File

1. Create a server health log file that contains a sequential number of outputs with the hostname, date and time, and a simple header:

```
cloud_user@server1: ~ $ { echo " " ; echo "==== `date` on `hostname` ====" ; df -hT ;
}

==== Wed May  1 20:53:07 UTC 2024 on server1 ====
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.8G     0  1.8G   0% /dev
tmpfs          tmpfs     1.9G     0  1.9G   0% /dev/shm
tmpfs          tmpfs     1.9G   17M  1.9G   1% /run
tmpfs          tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/xvda2     xfs        20G   14G  6.7G  67% /
tmpfs          tmpfs     373M  4.0K  373M   1% /run/user/1001
```

2. Rerun the previous command, and output this to a text file:

```
{ echo " " ; echo "==== `date` on `hostname` ====" ; df -hT ; } > `hostname`-health.txt
```

3. Review the output:

```
cloud_user@server1: ~ $ cat server1-health.txt

==== Wed May  1 20:53:51 UTC 2024 on server1 ====
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.8G     0  1.8G   0% /dev
tmpfs          tmpfs     1.9G     0  1.9G   0% /dev/shm
tmpfs          tmpfs     1.9G   17M  1.9G   1% /run
```

```
tmpfs        tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/xvda2   xfs        20G   14G  6.7G  67% /
tmpfs        tmpfs    373M  4.0K  373M   1% /run/user/1001
```

Observe what was added to the file.

4. Rerun the first `echo` command to create the server health log file twice more:

```
{ echo " " ; echo "==== `date` on `hostname` ====" ; df -hT ; } > `hostname`-health.txt
```

5. Review the output using `cat server1-health.txt` again to see how many times the command was run.
6. Change the server health log file to contain a double redirect to ensure that the initial and subsequent outputs are not overwritten:

```
{ echo " " ; echo "==== `date` on `hostname` ====" ; df -hT ; } >> `hostname`-health.txt
```

7. Run the above command 2 more times.
8. Inspect the output again using `cat server1-health.txt`.
9.

cat server1-health.txt

```
cloud_user@server1: ~ $ cat server1-health.txt

==== Wed May  1 20:58:43 UTC 2024 on server1 ====
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.8G     0  1.8G   0% /dev
tmpfs          tmpfs     1.9G     0  1.9G   0% /dev/shm
tmpfs          tmpfs     1.9G   17M  1.9G   1% /run
tmpfs          tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/xvda2     xfs        20G   14G  6.7G  67% /
tmpfs          tmpfs     373M  4.0K  373M   1% /run/user/1001

==== Wed May  1 21:00:24 UTC 2024 on server1 ====
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.8G     0  1.8G   0% /dev
tmpfs          tmpfs     1.9G     0  1.9G   0% /dev/shm
tmpfs          tmpfs     1.9G   17M  1.9G   1% /run
tmpfs          tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/xvda2     xfs        20G   14G  6.7G  67% /
tmpfs          tmpfs     373M  4.0K  373M   1% /run/user/1001

==== Wed May  1 21:00:25 UTC 2024 on server1 ====
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  1.8G     0  1.8G   0% /dev
tmpfs          tmpfs     1.9G     0  1.9G   0% /dev/shm
tmpfs          tmpfs     1.9G   17M  1.9G   1% /run
tmpfs          tmpfs     1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/xvda2     xfs        20G   14G  6.7G  67% /
tmpfs          tmpfs     373M  4.0K  373M   1% /run/user/1001
```

Search for Files in the `/home` Directory

1. Find all the files in the `/home` directory owned by the `cloud_user`:

   ```
   find /home -user cloud_user
   ```

   Observe a long list of filenames, along with 2 `Permission denied` errors at the end.
2. Generate clean lines of output without the 2 errors:

   ```
   find /home -user cloud_user 2> /dev/null
   ```

3. Find out how many clean lines of output there are:

   ```
   find /home -user cloud_user 2> /dev/null | wc -l
   ```

4. Save the output to a text file:

   ```
   find /home -user cloud_user 2> /dev/null > cloud_user-files.txt
   ```

5. Ensure the filenames are in the text file:

   ```
   cat cloud_user-files.txt
   ```

6. Number the list of filenames, and save them in another text file:

   ```
   nl cloud_user-files.txt > numberedfiles.txt
   ```

7. Ensure the numbered filenames are in the text file:

   ```
   cat numberedfiles.txt
   ```

8. Sort the numbered lines:

   ```
   sort numberedfiles.txt
   ```

   Observe that because there were no leading zeroes in front of the lower numbers, it doesn't sort properly.
9. To fix this, run a numeric sort:

   ```
   sort -n numberedfiles.txt
   ```

10. Generate a list showing the first-level directories inside `/etc`:

    ```
    find /etc -maxdepth 1
    ```

    Observe the output includes files that go 1 level further than then `/etc` directory.
11. Add on to the previous command to generate a list showing the space used for the first-level directories inside `/etc`:

    ```
    find /etc -maxdepth 1 -iname "*.*" -exec du -sh {} \;
    ```

    Observe the previous list now includes total space usage for each item.

12. Rerun the following command, and sort it by space usage from least to most used:

```
find /etc -maxdepth 1 -iname "*.*" -exec du -sh {} \; | sort -h
```

13. Rerun the previous command, and this time, output it to `etc-space-usage.txt`:

```
find /etc -maxdepth 1 -iname "*.*" -exec du -sh {} \; | sort -h > etc-space-usage.txt
```

    You shouldn't see any output aside from 2 directory errors.
14. Review the file:

```
less etc-space-usage.txt
```

    Observe the files listed and sorted by space usage.

Use `grep` and Regular Expressions to Analyze Text

1. Find all the files owned by the `cloud_user` in the `/home` directory:

```
find /home -user cloud_user
```

2. Find all the files owned by the `cloud_user` in the `/home` directory that contain the word "file":

```
cloud_user@server1: ~ $ find /home -user cloud_user | grep -i file
/home/cloud_user/.bash_profile
find: ‘/home/ssm-user’/home/cloud_user/numberedfiles.txt
/home/cloud_user/cloud_user-files.txt
: Permission denied
find: ‘/home/devops’: Permission denied
```

    Notice in the output that it only searches for the word "file" in the actual filenames rather than within the contents of the file.
3. Use the `-exec` feature of `find` to find the word "file" in the files themselves:

```
find /home -user cloud_user -exec grep -i file {} \;
```

4. Find out how many lines of output were found:

```
find /home -user cloud_user -exec grep -i file {} \; | wc -l
```

    You should see `140` lines at the bottom of the output.
5. Count how many lines were generated without errors:

```
find /home -user cloud_user -exec grep -i file {} \; 2> /dev/null | wc -l
```

You should again see `140` lines. Note that this counts only the standard out rather than the standard error.

6. Utilize the `grep` command directly to find the word "file" inside all files owned by `cloud_user` in the `/home` directory:

```
grep -ir file /usr/share/doc/zip
```

7. Add a total count of the output to the bottom of the list:

```
grep -ir file /usr/share/doc/zip ; !! | wc -l
```

You should see `965` at the bottom.

8. Run a case-sensitive search for only the word "file" as lowercase:

```
grep -ir file /usr/share/doc/zip ; grep -r file /usr/share/doc/zip | wc -l
```

You should see `925` at the bottom.

9. Create a text file containing all the search results for the word "file":

```
grep -ir file /usr/share/doc/zip ; grep -r file /usr/share/doc/zip > grepoutput.txt
```

10. Check the text file output:

```
cat grepoutput.txt
```

Revision #2
Created 1 May 2024 20:52:27 by Cesar Gzz
Updated 1 May 2024 21:48:20 by Cesar Gzz