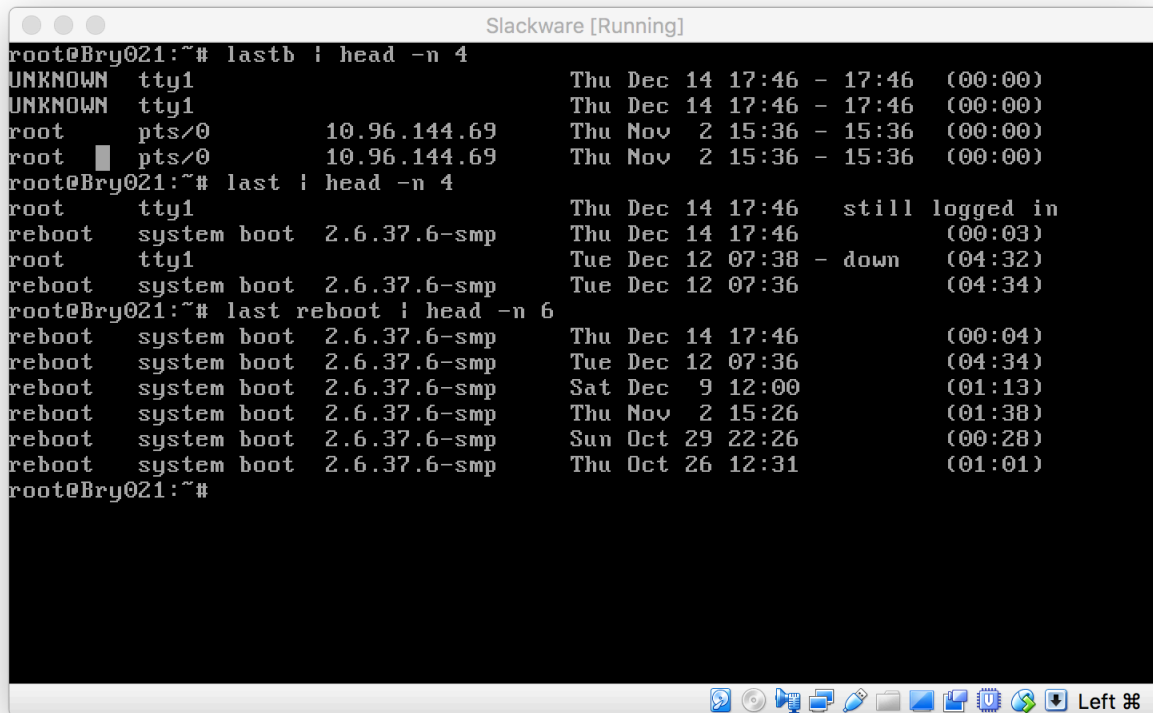


Lab 10: Further UNIX tools

Exercise 1. User and system information

1.1. How many login attempt occurred in the past 48 hours

1.2. How many system reboots occurred in the past 48 hours?



```
root@Bry021:~# lastb | head -n 4
UNKNOWN tty1 Thu Dec 14 17:46 - 17:46 (00:00)
UNKNOWN tty1 Thu Dec 14 17:46 - 17:46 (00:00)
root pts/0 10.96.144.69 Thu Nov 2 15:36 - 15:36 (00:00)
root pts/0 10.96.144.69 Thu Nov 2 15:36 - 15:36 (00:00)
root@Bry021:~# last | head -n 4
root tty1 Thu Dec 14 17:46 still logged in
reboot system boot 2.6.37.6-smp Thu Dec 14 17:46 (00:03)
root tty1 Tue Dec 12 07:38 - down (04:32)
reboot system boot 2.6.37.6-smp Tue Dec 12 07:36 (04:34)
root@Bry021:~# last reboot | head -n 6
reboot system boot 2.6.37.6-smp Thu Dec 14 17:46 (00:04)
reboot system boot 2.6.37.6-smp Tue Dec 12 07:36 (04:34)
reboot system boot 2.6.37.6-smp Sat Dec 9 12:00 (01:13)
reboot system boot 2.6.37.6-smp Thu Nov 2 15:26 (01:38)
reboot system boot 2.6.37.6-smp Sun Oct 29 22:26 (00:28)
reboot system boot 2.6.37.6-smp Thu Oct 26 12:31 (01:01)
root@Bry021:~#
```

Answer 1.1: The command “lastb” reports information about failed logins (2 unknown users on Thursday December 14th). The last reboots and successful logins are shown by the command “last”. See that there were 2 logins for the root user on the same day using the local terminal tty (notice that there was a reboot in between).

Answer 1.2: The command “last reboot” specifically reports the system reboots (init 0). In particular one occurred on Thursday December 14th at 17:46 (first line).

Exercise 2. Symbolic and hard links

2.1. Create a file `./extra_file` and a symlink `./links/extra_file_link` which links to `extra_file`. (use “`ls -l`” to check that the symlink has been created)

```
Slackware [Running]
root@Bry021:~/unixstuff# ls /usr/bin/mail -l
lrwxrwxrwx 1 root root 5 Mar  1 2012 /usr/bin/mail -> mailx*
root@Bry021:~/unixstuff#
root@Bry021:~/unixstuff# pwd
/root/unixstuff
root@Bry021:~/unixstuff# ln -s /root/unixstuff/extra_file /root/unixstuff/links/
extra_file_link
root@Bry021:~/unixstuff# ls -l links/
total 0
lrwxrwxrwx 1 root root 26 Dec 16 10:45 extra_file_link -> /root/unixstuff/extra_
file
root@Bry021:~/unixstuff# cat ./extra_file
extra_file links/
root@Bry021:~/unixstuff# cat ./links/extra_file_link
Exerciese 10.2

Symbolic and Hard Links
root@Bry021:~/unixstuff#
```

2.2 Edit `extra_file` and add some text to it. Now open `extra_file_link` by executing the `cat` command. Do you see the changes you made?

```
Slackware [Running]
root@Bry021:~/unixstuff# cat ./links/extra_file_link
Exerciese 10.2

Symbolic and Hard Links

Computer System and Servers

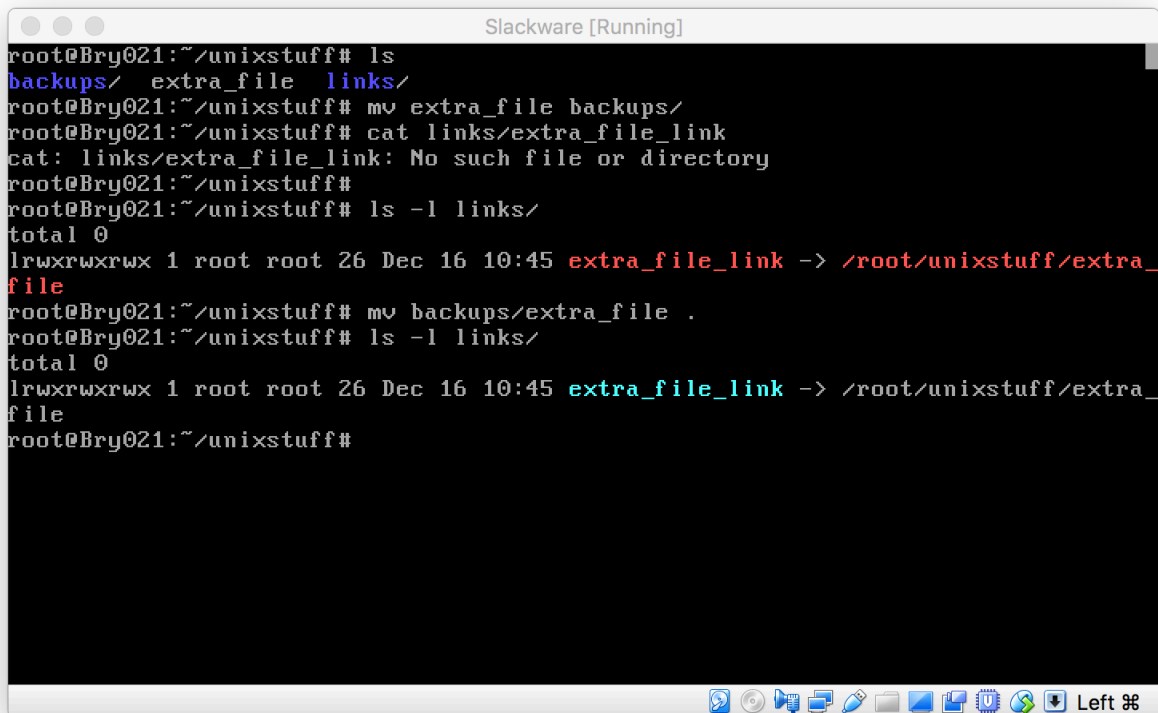
Cambridge, 2017

root@Bry021:~/unixstuff# _
```

Answer 2.1: At the beginning of the first screenshot the command “ls /usr/bin/mail -l” shows the example about the mail and mailx applications. In the same way the symlink /root/unixstuff/links/extra_file_link links to /root/unixstuff/extra_file, therefore the command cat ./links/extra_file_link will show the content of the file ./extra_file.

Answer 2.2: After editing the original file, the previous cat command (link) reports the changes I made.

2.3. Move extra_file to backups directory ./unixstuff/backups.



```
root@Bry021:~/unixstuff# ls
backups/  extra_file  links/
root@Bry021:~/unixstuff# mv extra_file backups/
root@Bry021:~/unixstuff# cat links/extra_file_link
cat: links/extra_file_link: No such file or directory
root@Bry021:~/unixstuff# ls -l links/
total 0
lrwxrwxrwx 1 root root 26 Dec 16 10:45 extra_file_link -> /root/unixstuff/extra_
file
root@Bry021:~/unixstuff# mv backups/extra_file .
root@Bry021:~/unixstuff# ls -l links/
total 0
lrwxrwxrwx 1 root root 26 Dec 16 10:45 extra_file_link -> /root/unixstuff/extra_
file
root@Bry021:~/unixstuff#
```

2.3.1. What happens to extra_file_link? Execute ls -l whilst in ./unixstuff/links, do you notice anything different?

There is an error in the symlink (message of no such a file or directory) and the command ls -l shows it in red colour.

2.3.2. Move extra_file back to unixstuff directory - predict what happens to extra_file_link.

When I move the file back to the unixstuff directory, the link comes back to the blue colour and the command cat ./links/extra_file_link works again.

2.4. Delete extra_file_link. What happens to extra_file?

It's evident that if I delete the symlink, nothing happens to the original extra_file, meaning that it's possible to open it using cat extra_file.

2.5. Recreate the extra_file_link and delete extra_file. What happens to extra_file_link?

If I recreate the symlink and delete now the extra_file, the symlink it's not going to be linked to anything. Moreover, the command ls -l is going to show it in red colour.

```
Slackware [Running]
root@Bry021:~/unixstuff# ls links/ -l
total 0
lrwxrwxrwx 1 root root 26 Dec 16 11:21 extra_file_link -> /root/unixstuff/extra_file
root@Bry021:~/unixstuff# rm links/extra_file_link
root@Bry021:~/unixstuff# cat extra_file
Exerciese 10.2

Symbolic and Hard Links

root@Bry021:~/unixstuff# ln -s /root/unixstuff/extra_file /root/unixstuff/links/extra_file_link
root@Bry021:~/unixstuff# rm extra_file
root@Bry021:~/unixstuff# cat links/extra_file_link
cat: links/extra_file_link: No such file or directory
root@Bry021:~/unixstuff# ls links/ -l
total 0
lrwxrwxrwx 1 root root 26 Dec 16 11:22 extra_file_link -> /root/unixstuff/extra_file
root@Bry021:~/unixstuff#
```

2.6. Delete extra_file_link and redo questions 1-5 above, this time use hard links instead. Hence, explain the differences between symbolic and hard links.

A file in the filesystem is a link to an inode. A hard link just creates another file with a link to the same underlying inode. When you delete a file, the OS just removes one link to the underlying inode. The inode is only deleted when all links have been removed. Once a hard link has been made, deleting, renaming or moving the original file will not affect the hard link as it links to the underlying node. Any changes to the data on the inode is reflected in all files that refer to that inode.

In the next screenshot a hard link has been create in the unixstuff/links directory.

```
Slackware [Running]
root@Bry021:~/unixstuff# ls
backups/ extra_file links/
root@Bry021:~/unixstuff# ln /root/unixstuff/extra_file /root/unixstuff/links/extra_file_link
root@Bry021:~/unixstuff# ls links/ -l
total 4
-rw-r--r-- 2 root root 60 Dec 16 11:46 extra_file_link
root@Bry021:~/unixstuff#
root@Bry021:~/unixstuff# mv extra_file backups/
root@Bry021:~/unixstuff# cat links/extra_file_link
Exercise 10.2

Symbolic and Hard Links

CSS, Cambridge 2017
root@Bry021:~/unixstuff# nano backups/extra_file _
```

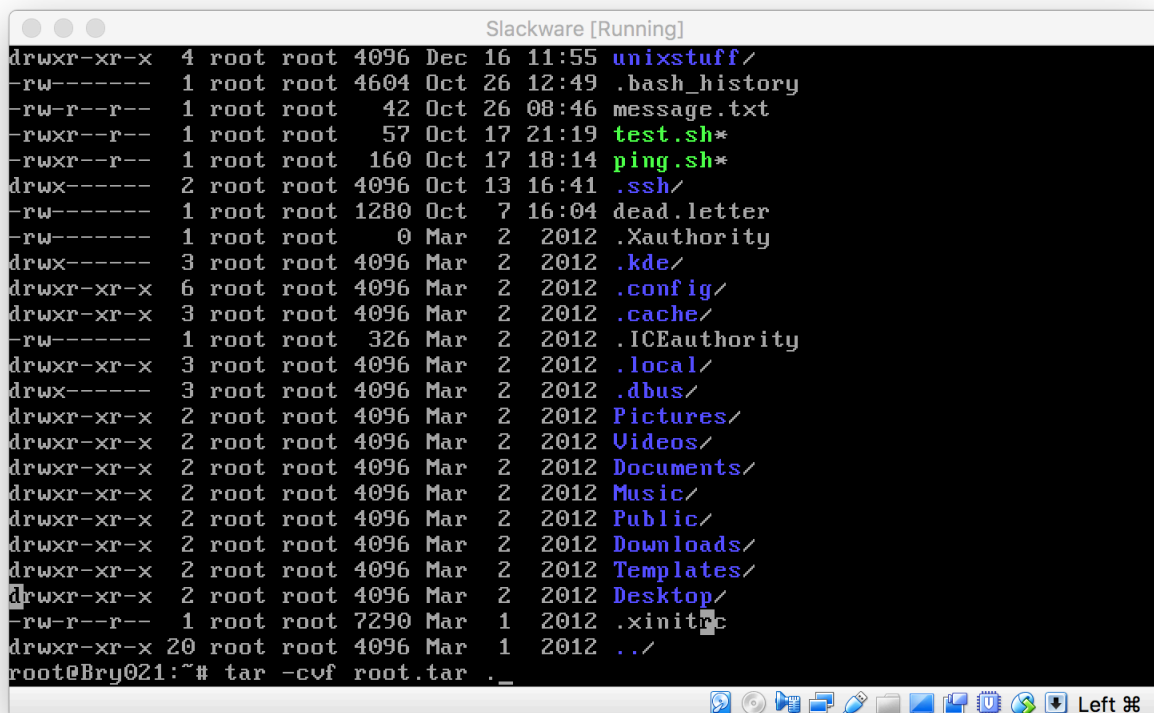
If we move the original `extra_file` to the `unixstuff/backups` directory, the hard link is still linking to the same inode. Moreover if `extra_file` is modified from there (backups folder), the hard link is still showing the latest modifications.

On the other hand, a symlink or symbolic link is a link to another name in the filesystem (similar to an alias). So once the location of the `extra_file` has changed, the link doesn't know if the file exists anymore.

Exercise 3. File compression and reboot

3.1. Archive the contents of your home directory using tar and cpio.

My home directory contains some files and hidden folders (`ls -alt`). To archive all of them in a file, I use the command:



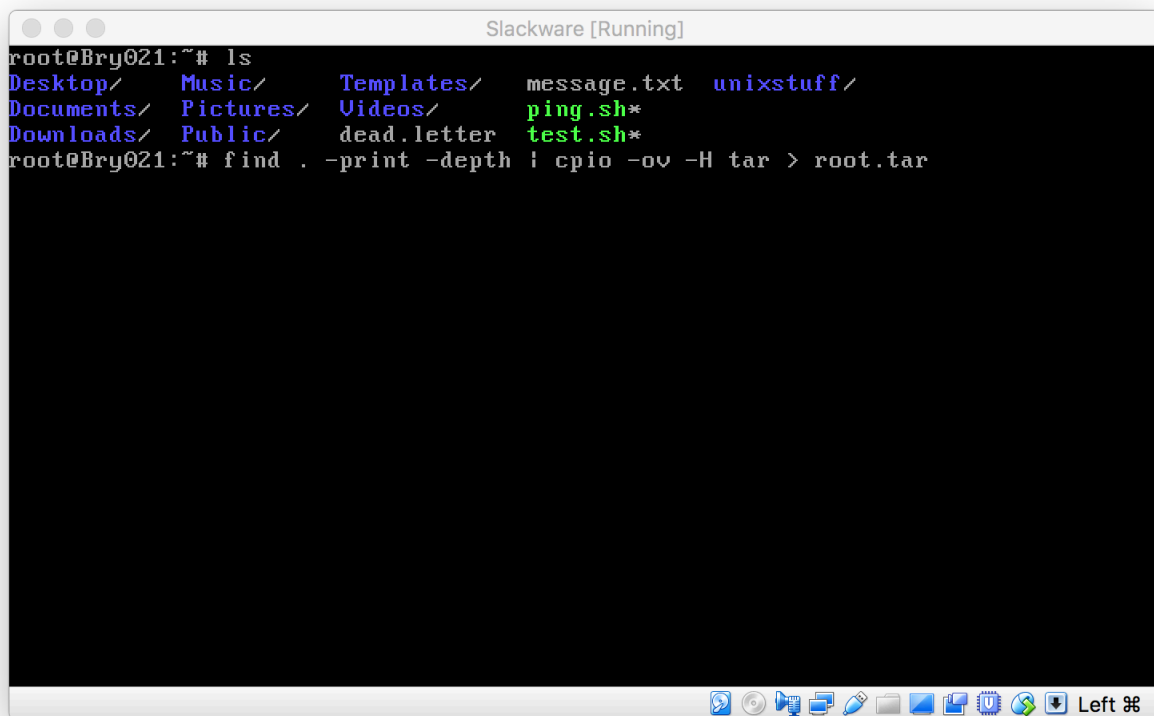
```
Slackware [Running]
drwxr-xr-x  4 root root 4096 Dec 16 11:55 unixstuff/
-rw-r----- 1 root root 4604 Oct 26 12:49 .bash_history
-rw-r--r--  1 root root  42 Oct 26 08:46 message.txt
-rwxr--r--  1 root root  57 Oct 17 21:19 test.sh*
-rwxr--r--  1 root root 160 Oct 17 18:14 ping.sh*
drwx----- 2 root root 4096 Oct 13 16:41 .ssh/
-rw-r----- 1 root root 1280 Oct  7 16:04 dead.letter
-rw-r----- 1 root root    0 Mar  2 2012 .Xauthority
drwx----- 3 root root 4096 Mar  2 2012 .kde/
drwxr-xr-x  6 root root 4096 Mar  2 2012 .config/
drwxr-xr-x  3 root root 4096 Mar  2 2012 .cache/
-rw-r----- 1 root root  326 Mar  2 2012 .ICEauthority
drwxr-xr-x  3 root root 4096 Mar  2 2012 .local/
drwx----- 3 root root 4096 Mar  2 2012 .dbus/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Pictures/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Videos/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Documents/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Music/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Public/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Downloads/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Templates/
drwxr-xr-x  2 root root 4096 Mar  2 2012 Desktop/
-rw-r--r--  1 root root 7290 Mar  1 2012 .xinitrc
drwxr-xr-x 20 root root 4096 Mar  1 2012 ../
root@Bry021:~# tar -cvf root.tar .
```

Here the `-c` option means create, `-v` means verbose and `-f` means file. The name of the output file is **root.tar** and the folder to compress is the current one (`/root`).

To list the content of the tar file, the options `-tvf` are used instead

```
# tar -tvf root.tar
```

Unlike tar, **cpio** doesn't automatically archive the contents of directories, so it's common to combine cpio with **find** when creating an archive:

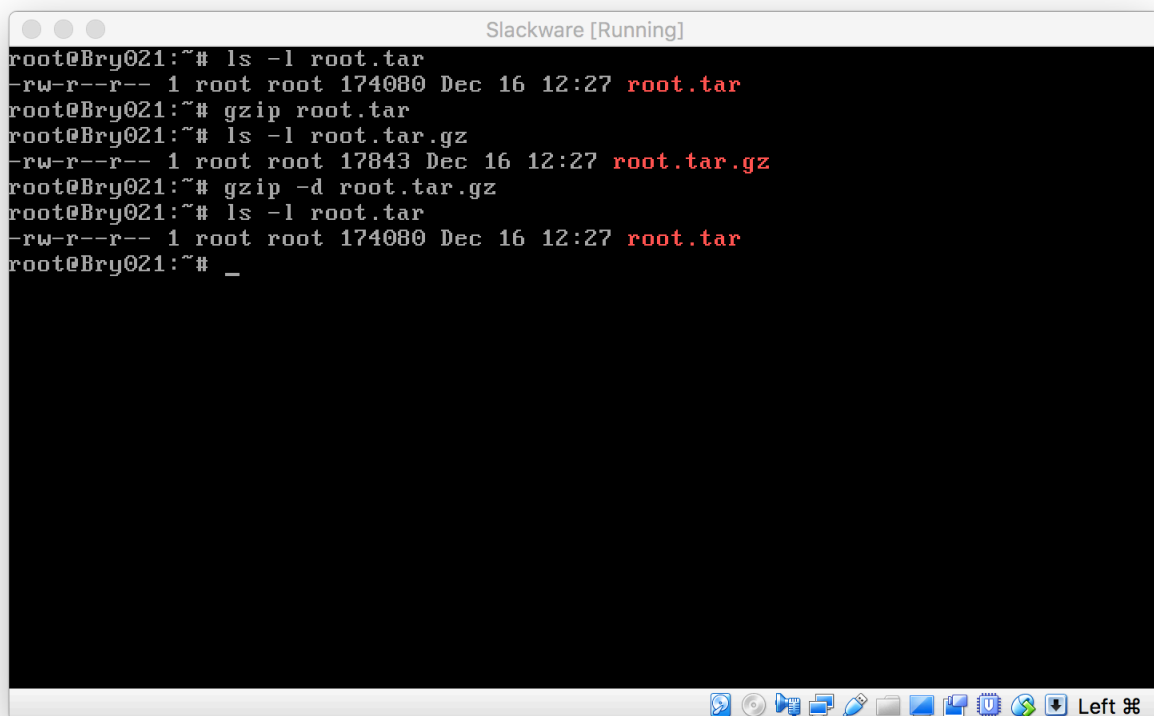
A terminal window titled "Slackware [Running]" with a black background and white text. The user is root@Bry021. The first command is 'ls', which lists the contents of the current directory: Desktop/, Music/, Templates/, message.txt, unixstuff/, Documents/, Pictures/, Videos/, ping.sh*, Downloads/, Public/, dead.letter, and test.sh*. The second command is 'find . -print -depth | cpio -ov -H tar > root.tar', which creates a tar archive named root.tar from the current directory and its subdirectories.

```
root@Bry021:~# ls
Desktop/    Music/      Templates/  message.txt  unixstuff/
Documents/  Pictures/   Videos/    ping.sh*
Downloads/  Public/     dead.letter test.sh*
root@Bry021:~# find . -print -depth | cpio -ov -H tar > root.tar
```

To list the contents of a cpis archive, use:

```
# cpio -tv < root.tar
```

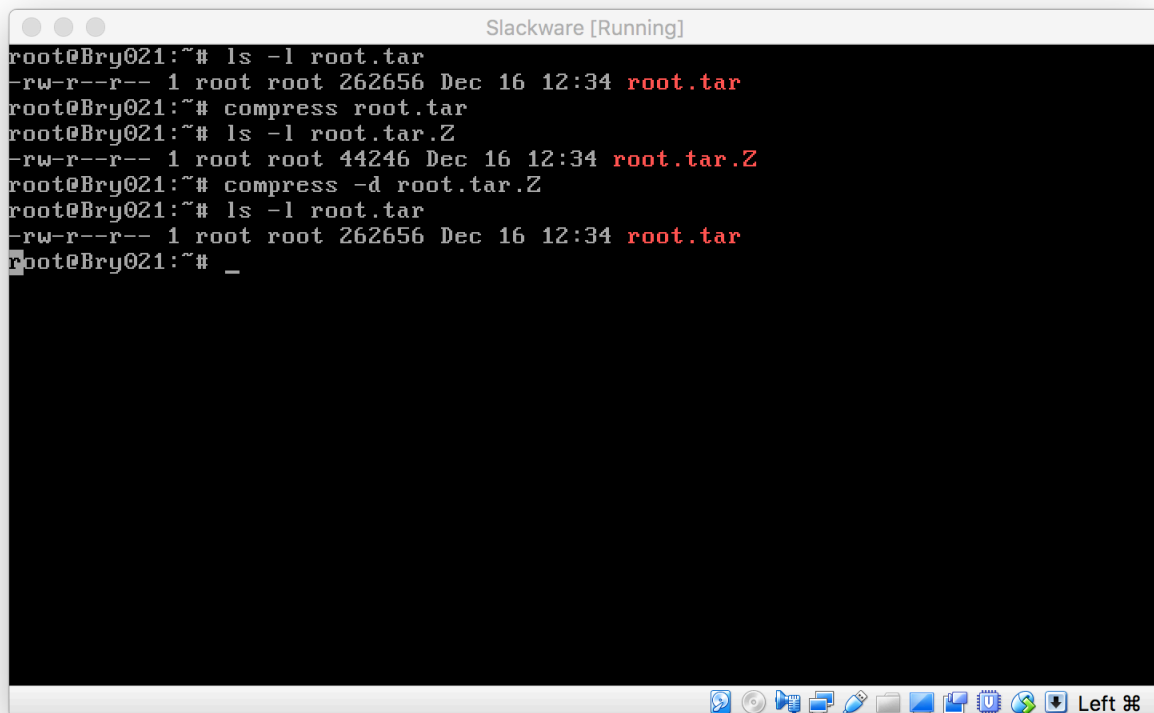
3.2. Compress the tar archive with gzip. Now extract their contents.

A terminal window titled "Slackware [Running]" with a black background and white text. The user is root@Bry021. The first command is 'ls -l root.tar', which shows the file root.tar with permissions -rw-r--r--, owner root, size 174080, and date Dec 16 12:27. The second command is 'gzip root.tar', which compresses the file. The third command is 'ls -l root.tar.gz', which shows the compressed file root.tar.gz with permissions -rw-r--r--, owner root, size 17843, and date Dec 16 12:27. The fourth command is 'gzip -d root.tar.gz', which decompresses the file. The fifth command is 'ls -l root.tar', which shows the decompressed file root.tar with permissions -rw-r--r--, owner root, size 174080, and date Dec 16 12:27. The sixth command is an underscore '_' which is not executed.

```
root@Bry021:~# ls -l root.tar
-rw-r--r-- 1 root root 174080 Dec 16 12:27 root.tar
root@Bry021:~# gzip root.tar
root@Bry021:~# ls -l root.tar.gz
-rw-r--r-- 1 root root 17843 Dec 16 12:27 root.tar.gz
root@Bry021:~# gzip -d root.tar.gz
root@Bry021:~# ls -l root.tar
-rw-r--r-- 1 root root 174080 Dec 16 12:27 root.tar
root@Bry021:~# _
```

Notice that the size of the tar.gz file is about 10 times smaller than the tar file is (174080/17843).

3.3. Compress the cpio archive with compress. Now extract their contents.



```
Slackware [Running]
root@Bry021:~# ls -l root.tar
-rw-r--r-- 1 root root 262656 Dec 16 12:34 root.tar
root@Bry021:~# compress root.tar
root@Bry021:~# ls -l root.tar.Z
-rw-r--r-- 1 root root 44246 Dec 16 12:34 root.tar.Z
root@Bry021:~# compress -d root.tar.Z
root@Bry021:~# ls -l root.tar
-rw-r--r-- 1 root root 262656 Dec 16 12:34 root.tar
root@Bry021:~# _
```

Notice that the size of the tar.Z file is about 6 times smaller than the tar file is (262656/44246).