Lab 06 - Package Management

1. *What is the meaning of `-o rw` in the command for mounting the floppy?*

Using `mount -o` will allow the user to select specific mounting options in a comma separated list, and `rw` will override the kernel default of mounting as read and will instead mount as read/write. `mount` will never attempt to read-only mount a filesystem when the `rw` argument is used.

2. *What is the meaning of the `-c` qualifier in the tar command?*

`-c` will create a new archive.

3. *What is the name of the tar file on the floppy?*

```
[root@UWS ~]# ls -lah /media/floppy/
total 39
drwxr-xr-x    3 root     root        1.0K Oct 28 19:40 .
drwxr-xr-x    3 root     root          60 Sep  9  2020 ..
-rw-r--r--    1 root     root       20.0K Oct 28 19:40 backuptest.tar
-rw-r--r--    1 root     root          61 Sep 13  2020 file.txt
drwx------    2 root     root       12.0K Sep 13  2020 lost+found
```

`backuptest.tar`

4. *Note down what was displayed by the tar command.*

```
[root@UWS ~]# tar -c /home/backuptest/ > /media/floppy/backuptest.tar
"tar: Removing leading `/' from member names"
```

There was no size displayed by the tar command, though the size of the tar is `20K`

5. *What is the meaning of the `-s` qualifier in `ls`? What does the `-k` qualifier achieve when used in conjunction with the `du` command?*

- The `-s` qualifier prints the allocated size of each - or in this case, a - file in blocks
- The `-k` qualifier in alongside `du` is equivalent to `--block-size=1K`

6. *What is the difference in size between the original directory and the backup on the floppy? Does `tar` perform any compression?*

```
[root@UWS ~]# ls -s /media/floppy/backuptest.tar
21 /media/floppy/backuptest.tar
[root@UWS ~]# du -k /home/backuptest/
20      /home/backuptest/
```

- The tar is a block larger in size.
- `tar` does not perform compression, it only archives.

7. *Examine `/tmp/home/backuptest` directory. Is this an exact copy of the original `/home/backuptest` directory?*

```
[root@UWS tmp]# diff /tmp/home/backuptest/ /home/backuptest/
[root@UWS tmp]#
```

`diff` presented no output, so they're the same

8. *Explain the meaning of the `xvf` qualifiers. Why is there no hyphen sign?*

The `xvf` qualifiers extract archive files verbosely.

9. *Why is there no output from the `cmp` command?*

`cmp` will only produce output if there's any difference between the passed files. Since they're identical, there's no output.

10. *Compare the date of creation of both files. Did the backup file inherit the same creation date?*

```
[root@UWS tmp]# ls -lisan /tmp/home/backuptest/world.dat
/home/backuptest/world.dat
   6747      4 -rw-r--r--    1 0        0               12 Nov  1 11:03
/home/backuptest/world.dat
   3884      4 -rw-r--r--    1 0        0               12 Nov  1 11:03
/tmp/home/backuptest/world.dat
```

The creation time of a file appears to be inaccessible, however both files share the same modified datetime which corresponds to the extraction datetime.

11. *Whilst examining `/tmp` do you see why tar removes the leading `/` of the path when creating an archive? Consider what would happen if the leading forward-slash was not removed by the `tar` command and you later attempted to un-tar the tarball (`tar` archive) in superuser mode*

- The path of the file we're archiving is relative.
- If the leading `/` wasn't omitted the application could potentially overwrite existing data.

12. *Check the size of the resulting file `/media/floppy/backuptest.tar.gz`. What is the compression rate now if you use the whole size of the directory `/home/backuptest` as a reference point?*

```
[root@UWS /]# tar -c /home/backuptest | gzip --verbose >
/media/floppy/backuptest.tar.gz
"tar: Removing leading `/' from member names"
"96.8%"
```

The compression ratio is 96.8%

```
[root@UWS /]# ls -lah /media/floppy/backuptest.tar.gz /home/backuptest
-rw-r--r--    1 root     root         674 Nov  1 11:29
/media/floppy/backuptest.tar.gz

/home/backuptest:
total 24
drwxr-xr-x    2 root     root          90 Nov  1 11:03 .
drwxr-xr-x    4 root     root          88 Nov  1 11:00 ..
-rw-r--r--    1 root     root        8.7K Nov  1 11:05 vmstat.dat
-rw-r--r--    1 root     root          12 Nov  1 11:03 world.dat
```

- The zipped tarball is 674 bytes in size

```
[root@UWS /]# du -k /home/backuptest /media/floppy/backuptest.tar.gz
20      /home/backuptest
1       /media/floppy/backuptest.tar.gz
```

- The zipped tarball is allocated a single block
- The original directory is assigned 20 blocks

13. *What means the `-p` option?*

`-p` extracts information about file permissions. It will preserve permissions on extraction.

14. *Why did the second command using `/home/backuptest/world.dat` as the destination not work?*

`tar` removed the leading `/` when the archive was initially created.

15. *Write a shell script that produces a zipped-tar version of any input directory called `dir_name`, whereby `dir_name` represents the relative directory name, not the absolute one. The output should be send to `/tmp` or a floppy device and should be of the form: `dir_name.tar.gz`.*

```
#!/bin/bash
file=`basename $1`
dir=`dirname $1`
tar -czf /media/floppy/`$file`.tar.gz -C $dir $file
```

16. *What does the `nsnake` package do?*

```
[root@UWS ~]# apt-cache search nsnake
nsnake - classic snake game on the terminal
```

17. *What does the number shown in the output of the command pidof represent?*

```
[root@UWS ~]# pidof nsnake
573
[1]+  Stopped(SIGTTOU)        /usr/games/nsnake
```

- `573` represents the process ID
- `[1]` represents the parent ID

18. *Verify that nsnake has been deleted by checking `/usr/games/nsnake` . What did you find this time?*

```
[root@UWS ~]# whereis nsnake
nsnake:
[root@UWS ~]# ls -lah /usr/games/nsnake
ls: /usr/games/nsnake: No such file or directory
```

19. *Briefly outline the functionality of the `md5sum` command.*

`md5sum` calculates MD5 cryptographic checksums.

20. *Did you get a match with the md5 checksum and if so what does this prove?*

```
[root@UWS ~]# md5sum hello.c
1813f4cdd3fcf986a25981f95c0dc0d2  hello.c
```

I'm not sure I understand the question. The above command simply presents the MD5 checksum for a file. The wording of this question '... get a match' implies some other file's MD5 sum would be tested against `hello.c`. If the aim of the exercise was to compare the sum of `hello.c` and its compiled `hello` counterpart, it stands to reason that the values would not match.

```
[root@UWS ~]# md5sum hello hello.c
8fcc5e999b835db0c79c570918539317  hello
6b9d1a1169e6ec3b160e5e5883765e7c  hello.c
```

If the aim was to compare the file to itself:

```
[root@UWS ~]# md5sum hello hello.c
8fcc5e999b835db0c79c570918539317  hello
6b9d1a1169e6ec3b160e5e5883765e7c  hello.c
```

It stands to reason that the sum would match. The point of checking an MD5 sum is to ascertain the integrity of a file after it's been through some change. A common practice is to check the sum of a file locally against some known value in order to ensure the file's integrity is valid.

21. *What is the difference in size between `hello.c` and `hello`?*

```
[root@UWS ~]# ls -lah | grep hello
-rwxr-xr-x    1 root     root        1.8K Nov  1 12:16 hello
-rw-------    1 student  root         108 Nov  1 12:16 hello.c
```

22. Can you explain the size differences?

The compilation process converts source code into object or machine code and comprises of pre-processing, compiling, assembling, and linking, though these steps are generally contingent on the compiler. An output binary is almost always significantly larger than the input source file as a result of this process.

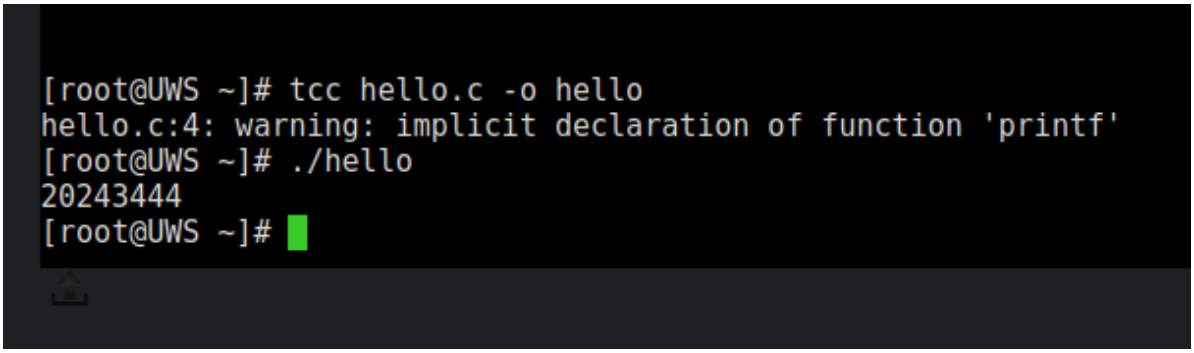
23. *Modify the `hello.c` to display your name and banner ID instead of "Hello, World!".*

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
    printf("20243444\n");
    return 0;
}
```

24. *Compile and run the modified program.*

```
[root@UWS ~]# tcc hello.c -o hello
"hello.c:4: warning: implicit declaration of function 'printf'"
[root@UWS ~]# ./hello
20243444
```

25. *Do a screenshot and add it to your notes and lab book.*


[root@UWS ~]# tcc hello.c -o hello
hello.c:4: warning: implicit declaration of function 'printf'
[root@UWS ~]# ./hello
20243444
[root@UWS ~]#