Exercise 37.1 Using tar for Backup

1. Create a directory called `backup` and in it place a compressed **tar** archive of all the files under `/usr/include`, with the highest level directory being `include`. You can use any compression method (**gzip**, **bzip2** or **xzip**).
2. List the files in the archive.
3. Create a directory called `restore` and unpack and decompress the archive.
4. Compare the contents with the original directory the archive was made from.

Solution 37.1

1. ```
   $ cd backup
   $ cd /usr ; tar zcvf include.tar.gz include
   ```

   or

   ```
   $ tar -C /usr -zcf include.tar.gz include
   $ tar -C /usr -jcf include.tar.bz2 include
   $ tar -C /usr -Jcf include.tar.xz include
   ```

   Notice the efficacy of the compression between the three methods:

   ```
   $ du -sh /usr/include
   55M        /usr/include
   ```

2. ```
   $ ls -lh include.tar.*
   -rw-rw-r-- 1 coop coop 5.3M Nov  3 14:44 include.tar.bz2
   -rw-rw-r-- 1 coop coop 6.8M Nov  3 14:44 include.tar.gz
   -rw-rw-r-- 1 coop coop 4.7M Nov  3 14:46 include.tar.xz
   ```

3. ```
   $ tar tvf include.tar.xz
   qdrwxr-xr-x root/root         0 2014-10-29 07:04 include/
   -rw-r--r-- root/root     42780 2014-08-26 12:24 include/unistd.h
   -rw-r--r-- root/root       957 2014-08-26 12:24 include/re_comp.h
   -rw-r--r-- root/root     22096 2014-08-26 12:24 include/regex.h
   -rw-r--r-- root/root      7154 2014-08-26 12:25 include/link.h
   .....
   ```

   Note it is not necessary to give the `j`, `J`, or `z` option when decompressing; **tar** is smart enough to figure out what is needed.

4. ```
   $ cd .. ; mkdir restore ; cd restore
   $ tar xvf ../backup/include.tar.bz2
   include/
   include/unistd.h
   include/re_comp.h
   include/regex.h
   include/link
   .....
   $ diff -qr include /usr/include
   ```