System Administration for Beginners

Week 3 Homework Solutions

1. (b) False — A symbolic link is just a *pointer* that points to the file. It does not make a copy of the file.

2. (c) 640 — Remember that read has a value of 4, write has a value of 2, and execute has a value of 1. Permissions are grouped together like so: owner, group, other. To assign read and write permissions to owner, we add up the values for both (4 + 2), and so on.

3. (c) command | grep word | less — The pipe is used to send output from one command as input into another command. The greater-than signs are used for outputting to a file (>) or appending to a file (>>).

4. (b) False — `tar` will put a simple container around the files/directories. It does not compress the files inside.

5. (c) `-r` — This was simply an exercise in looking at the `man` pages.

6. (b) False — On a system where you cannot trust your users, assigning 777 as permissions to any file can create security holes. Read up on permissions again if you are not sure why this is the case.

7. (a) True — As long as you have set up your groups correctly (we will cover this in a future lecture), then these default permissions should be fine.

8. (d) Execute — Since you are not the owner, you will fit under the "group" permission before "other", since that takes precedence rather than highest permissions first. That is, even though the "other" has read, you will still only get execute.

9. (b) Symbolic link permissions only refer to the link . . . — This question may have been worded poorly according to a few. The idea in question was if there was a file I wanted to access (for example, `/etc/shadow`), and I did not have the correct permissions, could I make a symbolic link to it in order to get access to that file, since the actual symbolic link permissions are set to 777? This will not work, as symbolic link permissions do not really mean anything. It is the permissions of what the symbolic link is pointing to which is important.