# Linux File Permissions

Abraham J. Reines

February 15, 2024

## Understanding Linux file permissions

1. **Can Alice read Bob's file `data.txt`? Why?**
   Yes, because the permissions for `data.txt` are `-rw-r-r-`. The file is readable by others, which includes Alice.

2. **Can Alice remove Bob's file `data.txt`? Why?**
   No, because Alice does not have write permission on the parent directory `/cs/home/stu/bob`, which is necessary to remove a file.

3. **Can Alice read Bob's file `secret.txt`? Why?**
   No, because the permissions for `secret.txt` are `-rw-r---`. The file is only readable and writable by the owner, who is Bob.

4. **Can Alice remove Bob's file `secret.txt`? Why?**
   No, for the same reason as `data.txt`. Without write permissions on the containing directory, she cannot remove files within it.

5. **When Bob creates a new file with command `echo "My Super Secret is b7d5d78shes" > mysecret.txt`, what are the full permissions of this file?**
   The permissions would be `-rw-r-r-` because the umask defaults to `0022`, which subtracts write permissions for group and others from the default permissions of `666` for files.

## Setting Linux file permissions

1. **Can Bob change the permissions so that all other students in `csmajor` can read `data.txt`, but any other users who are not in `csmajor` cannot?**
   Yes, Bob can execute `chmod 640 /cs/home/stu/bob/data.txt`. This sets read and write permissions for the owner, read for the group, and no permissions for others.

2. **If Bob wants to set the default permission of his new files to be readable/writable by himself and the group, and readable by others, what commands should he use? Hint: use umask.**
   Bob should use `umask 002`. This umask value will create files with permissions `664` (`rw-rw-r-`) and directories with permissions `775` (`rwxrwxr-x`).

## A more complex case

Alice needs to perform the following steps to set the correct file permissions for `treasure.txt`:

1. Create a group named `treasure_group` and add Bob to it.
2. Change the group ownership of `treasure.txt` to `treasure_group`.
3. Set the file permissions so the owner and group have read and write permissions, but not execute permissions.
4. Utilize Access Control Lists (ACLs) to grant Charlie read permissions without granting him write permissions.
5. Ensure all other users have no permissions to `treasure.txt`.

The following LaTeX listing provides the exact Linux commands Alice should execute:

```
# Create a group for the treasure file and add Alice and Bob
sudo groupadd treasure_group
sudo usermod -a -G treasure_group alice
sudo usermod -a -G treasure_group bob

# Change group ownership of the file to treasure_group
chgrp treasure_group treasure.txt

# Set read and write permissions for the owner and group, and no permissions for
    others
chmod 660 treasure.txt

# If ACLs are enabled on the filesystem, set an ACL for Charlie to read
setfacl -m u:charlie:r-- treasure.txt

# Verify the ACL settings
getfacl treasure.txt

# Ensure the file is not executable by anyone
chmod a-x treasure.txt
```

Alice should test these commands in her Linux environment to ensure they work as intended. If she encounters any issues, such as not having the required permissions or ACLs not being enabled, she will need to contact the system administrator for assistance.