Hard Links and Symbolic Links

Exercise 1: Understanding Links
Objective: Learn the difference between hard links and symbolic links.

1. Create a new directory called "link_test" and navigate into it:
```
mkdir link_test
 cd link_test
```

2. Create a new file called "original_file.txt":
```
touch original_file.txt
```

3. Check the inode number of "original_file.txt" using:
```
 ls -i original_file.txt
```

Note down the inode number.

Exercise 2: Creating a Hard Link
Objective: Create and verify a hard link.

1. Create a hard link to "original_file.txt" called "hard_link_file.txt":
```
ln original_file.txt hard_link_file.txt
```

2. Check the inode numbers of both files to confirm they are the same:
```
ls -i original_file.txt hard_link_file.txt
```

- What do you observe about the inode numbers?

3. Modify "original_file.txt" (add text to it):
```
echo "This is the original file." > original_file.txt
```

4. Check the contents of both files to verify that they are the same:
```
cat original_file.txt
cat hard_link_file.txt
```

Exercise 3: Deleting a Hard Link
Objective: Understand the implications of deleting a hard link.

1. Delete "original_file.txt":
```
rm original_file.txt
```

2. Check if "hard_link_file.txt" still exists and verify its contents:
```
ls
cat hard_link_file.txt
```

- What happens to "hard_link_file.txt" when you delete "original_file.txt"?

Exercise 4: Creating a Symbolic Link
Objective: Create and verify a symbolic link.

1. Navigate back to the "link_test" directory:
```
cd link_test
```

2. Create a symbolic link to "hard_link_file.txt" called "symbolic_link_file.txt":
```
ln -s hard_link_file.txt symbolic_link_file.txt
```

3. List the contents of the directory and observe the symbolic link:
```
ls -l
```

- What does the output show regarding "symbolic_link_file.txt"?

4. Check the contents of the symbolic link:
```
cat symbolic_link_file.txt
```

Exercise 5: Modifying the Target of a Symbolic Link
Objective: Understand how symbolic links behave when the target file changes.

1. Modify "hard_link_file.txt":
```
echo "This is a hard link file." > hard_link_file.txt
```

2. Check the contents of "symbolic_link_file.txt" to see if it reflects the change:
```
cat symbolic_link_file.txt
```

Exercise 6: Deleting a Symbolic Link
Objective: Learn what happens when a symbolic link is deleted.

1. Delete the symbolic link:
```
rm symbolic_link_file.txt
```

2. Verify that "hard_link_file.txt" still exists and check its contents:
```
ls
cat hard_link_file.txt
```

- What happens to "hard_link_file.txt" when you delete "symbolic_link_file.txt"?

Exercise 7: Limitations of Hard Links
Objective: Understand the limitations of hard links.

1. Try to create a hard link to a directory (you should get an error):
```
  ln /path/to/some/directory
/path/to/some/directory/hard_link_to_directory
```

2. Check the error message. Hard links cannot be created for directories (except for the special entries . and ..).

Exercise 8: Summary of Links
Objective: Summarize your understanding of links.

1. Write a brief note (in a file called link_summary.txt) explaining the key differences between hard links and symbolic links, including when to use each type of link.

```
echo "Hard links point to the same inode and cannot link to
directories. Symbolic links point to the file name and can link to
directories, but if the target file is deleted, the symbolic link
becomes broken." > link_summary.txt
```