TUTORIAL NO. 9

Q1. Create a pipe between the "/bin/ls -al" command and the "/usr/bin/tr a-z A-Z" command. This is the equivalent of running the shell command:

```
/bin/ls -al | /usr/bin/tr a-z A-Z
```

The first command generates a long-format directory listing of the root (/) directory and the second command takes that listing and translates all lowercase characters to uppercase.

Steps:

1. Start by creating a pipe.
2. Then fork a child process.
3. The parent should use the pipe for command output. That means it needs to change its standard output file descriptor (1) to the writing end of the pipe (pfd[1]). It does this via the dup2 system call: dup2(pfd[1], 1). Then it executes the command in cmd1.
4. The child should use the pipe for command input. It needs to change its standard input file descriptor (0) to the reading end of the pipe (pfd[0]). It also does this via the dup2 system call: dup2(pfd[0], 0). Then it executes the command in cmd2.

Q2. Copy the contents of a file “example.txt” to “example_copy.txt” using cat command. Use exec() family to execute cat inside a child process. Redirect the output of the command “cat example.txt” to the file example_copy.txt. by using dup() or dup2().

Steps:

1. Using open(), create a stream which refers to example_copy.txt . (use appropriate arguments to create the file if the file is missing)
2. dup()stdout of the process to the above created stream.
3. Close the stdout of the main process.
4. Create a child which executes “cat example.txt” using any of the exec() functions.

a. Write the above code using only dup().
b. Write it now using only dup2().

Advanced: Check if the above question can be solved using pipes. If yes, then write a code to achieve the same. If no, support with valid reasons.