**Musab Mehadi**

[mmehadi@jacobs-university.de](mailto:mmehadi@jacobs-university.de)

Sheet#9

**Problem 9.1**

A) The programme

\* checks if a file is attached i.e the number of command line arguments should be 2 (if not gives out error and exits) and opens the file in a read only mode.

\* If successful, it forks the created process and endlessly reads and writes the content in the file to the console.

$ rm -f foo -> removes the file foo

$ touch foo -> creates the file foo (but its going be empty meaning whatever was saved before it got deleted can not be recovered)

$ ./catloop foo &

$ echo -n "hello " > foo -> writes “hello” to the file foo (-n removes the new line at the end )

B) $ echo "world" >> foo appends world to the file we had with a new line at the end.

Helloworld ……going in an infinite loopA screenshot of a cell phone

Description automatically generated

$ truncate -s 0 foo this command emptys the file foo .However, the programme does not terminate yet as it just waits for another command.

A screenshot of a cell phone

Description automatically generated

C) This is useful as it doesn’t require us to restart the programme again whenever we want to make changes but it can be disadvantageous as it is risky when some accidental changes can be made without restarting the programm giving us output we don’t desire.

D) $ chmod 0 foo normally locks the attached file( no permissions to write read or execute the file for the user,group and others as well).

Executing the command while the program is running will have no effect and we can still be able to read from the file foo.

- Running the programme again after terminating it however, will give output Permission denied.

* $ ls -l foo gives the read write and execute information of the file.

Text

Description automatically generated

A screenshot of a cell phone screen with text

Description automatically generated

E) executing $ rm -f foo command while running the programme will not affect the output(although the file is removed) as it is already saved when programme started.Changes make effect when the programme is restarted again.

\* One implication of this that we could accidently delete a file and not realise it as the programme continues to print the previously stored content.

\*We should also restart the programme if we want to save the changes we made.

**Problem 9.2)**

1. - file owner -> read , write and execute

-file’s group members -> read and write

-everybody else -> only read

1. - file owner -> execute,read and write

-file’s group members -> execute and write

-everybody else -> they don’t have any permission

1. No, because as we can see from part B group are not allowed to read .

Yes,because they are given permission to write and execute.

D)-The permissions on of file world on linux by default is rw- rw- rw-

– when umask of 0022 executes, the writing privledge is removed from the file’s group members and everybody else.

-The sudo command calls for administrator permissions( asks for password) making it the owner

E)it shows the file permissions for the root user.The user is found in /user/bin/sudo.So any commands starting with sudo are executed with the root user that has the attribute -0s set which stands for setud or setgid.This tells the system to run the executable as the owner with the owner’s permission so that user can run the file without root access.