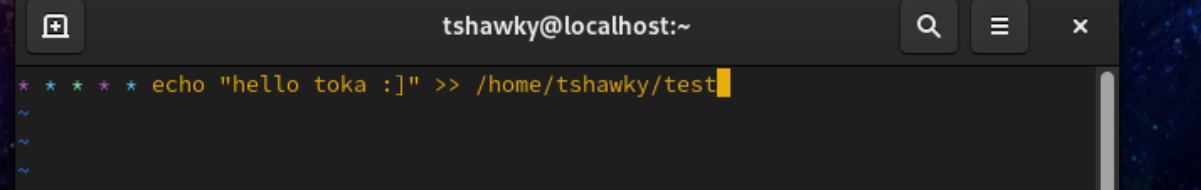


1. How do you edit the current user's crontab?

```
[tshawky@localhost ~]$ crontab -e
```



2. How do you list scheduled cron jobs for the current user?

```
[tshawky@localhost ~]$ crontab -l
* * * * * echo "hello toka :]" >> /home/tshawky/test
[tshawky@localhost ~]$
```

3. How do you delete all cron jobs for the current user?

```
[tshawky@localhost ~]$ crontab -r
[tshawky@localhost ~]$ crontab -l
no crontab for tshawky
[tshawky@localhost ~]$
```

4. Cron Syntax Questions

- a. Schedule a job to run every day at 3:30 AM
- b. Run a script every Monday at 5:15 PM
- c. Execute a command every 10 minutes
- d. Schedule a task to run every weekday (Mon-Fri) at 6:30 PM
- e. Run a job on the 1st and 15th of each month at 8:00 AM

```
30 3 * * * echo "task 3:30 " >> /home/tshawky/test
15 17 * * 1 echo "mon 5:15pm task" >> /home/tshawky/test
*/10 * * * * echo "every 10min" >> /home/tshawky/test
30 18 * * 1-5 echo "Every weekday (Mon-Fri) at 6:30 PM task" >> /home/tshawky/test
* 8 1,15 * * echo "1st and 15th of each month at 8:00 AM task" >> /home/tshawky/test
```

5. How do you restrict users from using cron?

Put them in deny ...OR make only allowed persons in .allow file

```
[root@localhost ~]# echo "tshawky" >> /etc/cron.deny
```

```
[tshawky@localhost ~]$ crontab -e
You (tshawky) are not allowed to use this program (crontab)
See crontab(1) for more information
[tshawky@localhost ~]$
```

6. Set UserID && SetGroupID

- a. Create File and Set Execution and setuid permission
 - b. Check if setuid is enabled on File
-
- a. Create directory testdir and set setgid
 - b. Check if setgid is enabled on directory
 - c. Logout and login as root
 - d. Create testfile in testdir and verify ownership

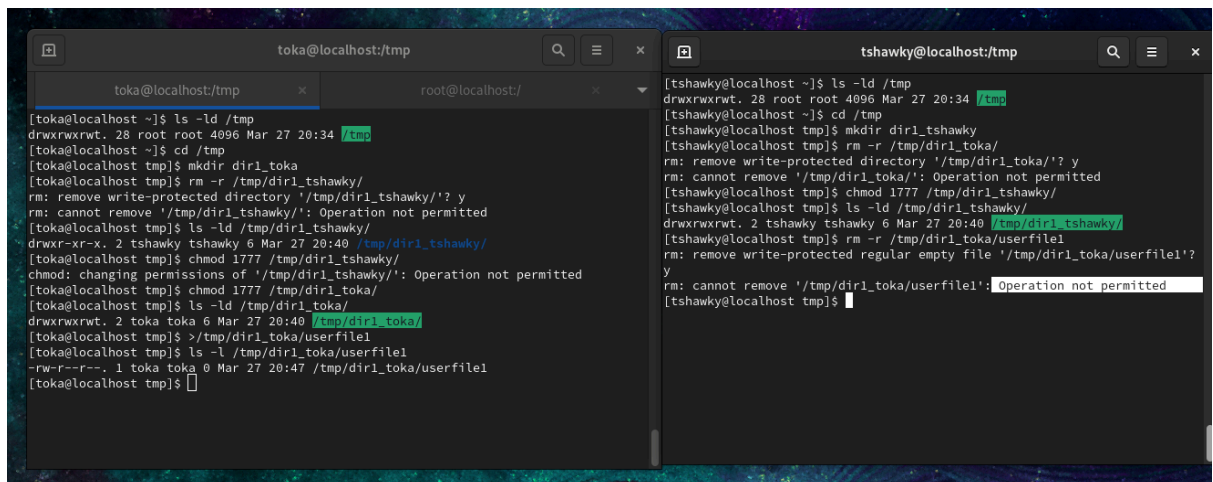
```

[toka@localhost tmp]$ touch /tmp/setuidfile
[toka@localhost tmp]$ chmod 4755 /tmp/setuidfile
[toka@localhost tmp]$ ls -ld /tmp/setuidfile
-rwsr-xr-x. 1 toka toka 0 Mar 27 23:23 /tmp/setuidfile
[toka@localhost tmp]$ mkdir /tmp/testdir
[toka@localhost tmp]$ chmod 2775 /tmp/testdir
[toka@localhost tmp]$ ls -ld /tmp/testdir
drwxrwsr-x. 2 toka toka 6 Mar 27 23:24 /tmp/testdir
[toka@localhost tmp]$ su -
Password:
[root@localhost ~]# touch /tmp/testdir/testfile
[root@localhost ~]# ls -l /tmp/testdir/testfile
-rw-r--r--. 1 root toka 0 Mar 27 23:26 /tmp/testdir/testfile
[root@localhost ~]#

```

8. Stick bit

- Create public directory dir1 with sticky bit
- Check if sticky bit is enabled on directory
- Logout and login as user1
- Create userfile1 in dir1
- Login as user2 and attempt to remove userfile1



```

toka@localhost/tmp
toka@localhost/tmp
root@localhost/

[toka@localhost ~]$ ls -ld /tmp
drwxrwxrwt. 28 root root 4096 Mar 27 20:34 /tmp
[toka@localhost ~]$ cd /tmp
[toka@localhost tmp]$ mkdir dir1_toka
[toka@localhost tmp]$ rm -r /tmp/dir1_tshawky/
rm: remove write-protected directory '/tmp/dir1_tshawky/': Operation not permitted
[toka@localhost tmp]$ ls -ld /tmp/dir1_tshawky/
drwxr-xr-x. 2 tshawky tshawky 6 Mar 27 20:40 /tmp/dir1_tshawky/
[toka@localhost tmp]$ chmod 1777 /tmp/dir1_tshawky/
chmod: changing permissions of '/tmp/dir1_tshawky/': Operation not permitted
[toka@localhost tmp]$ ls -ld /tmp/dir1_toka/
drwxrwxrwt. 2 toka toka 6 Mar 27 20:40 /tmp/dir1_toka/
[toka@localhost tmp]$ >/tmp/dir1_toka/userfile1
[toka@localhost tmp]$ ls -l /tmp/dir1_toka/userfile1
-rw-r--r--. 1 toka toka 0 Mar 27 20:47 /tmp/dir1_toka/userfile1
[toka@localhost tmp]$

tshawky@localhost/tmp
[tshawky@localhost ~]$ ls -ld /tmp
drwxrwxrwt. 28 root root 4096 Mar 27 20:34 /tmp
[tshawky@localhost ~]$ cd /tmp
[tshawky@localhost tmp]$ mkdir dir1_tshawky
[tshawky@localhost tmp]$ rm -r /tmp/dir1_toka/
rm: remove write-protected directory '/tmp/dir1_toka/': Operation not permitted
[tshawky@localhost tmp]$ chmod 1777 /tmp/dir1_tshawky/
[tshawky@localhost tmp]$ ls -ld /tmp/dir1_tshawky/
drwxrwxrwt. 2 tshawky tshawky 6 Mar 27 20:40 /tmp/dir1_tshawky/
[tshawky@localhost tmp]$ rm -r /tmp/dir1_toka/userfile1
rm: remove write-protected regular empty file '/tmp/dir1_toka/userfile1': Operation not permitted
[tshawky@localhost tmp]$

```

9. Give user john read and write permissions on myfile.txt

10. How to view the ACL of a file named myfile.txt?

```
[toka@localhost ~]$ ls -l myfile.txt
-rw-r--r--. 1 toka toka 0 Mar 28 00:07 myfile.txt
[toka@localhost ~]$ sudo useradd nada
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for toka:
[toka@localhost ~]$ setfacl -m u:nada:rw myfile.txt
[toka@localhost ~]$ getfacl myfile.txt
# file: myfile.txt
# owner: toka
# group: toka
user::rw-
user:nada:rw-
group::r--
mask::rw-
other::r--

[toka@localhost ~]$
```

11. How do you give the group developers execute permissions on the directory /home/user/documents

```
[toka@localhost ~]$ sudo groupadd developers
groupadd: group 'developers' already exists
[toka@localhost ~]$ mkdir -p /home/user/documents
mkdir: cannot create directory '/home/user': Permission denied
[toka@localhost ~]$ sudo mkdir -p /home/user/documents
[toka@localhost ~]$ sudo setfacl -m g:developers:x /home/user/documents
[toka@localhost ~]$ getfacl /home/user/documents
getfacl: Removing leading '/' from absolute path names
# file: home/user/documents
# owner: root
# group: root
user::rwx
group::r-x
group:developers:--x
mask::r-x
other::r-x

[toka@localhost ~]$
```

12. How to remove the ACL entry for the user john on the file file1.txt?

```
[toka@localhost ~]$ sudo useradd john
[toka@localhost ~]$ ls -l file1.txt
-rw-r--r--. 1 toka toka 0 Mar 28 00:20 file1.txt
[toka@localhost ~]$ setfacl -m u:john:rw file1.txt
[toka@localhost ~]$ getfacl file1.txt
# file: file1.txt
# owner: toka
# group: toka
user::rw-
user:john:rw-
group::r--
mask::rw-
other::r--

[toka@localhost ~]$ setfacl -x u:john file1.txt
[toka@localhost ~]$ getfacl file1.txt
# file: file1.txt
# owner: toka
# group: toka
user::rw-
group::r--
mask::r--
other::r--

[toka@localhost ~]$
```

13. How do you set a default ACL for a directory so that all new files and directories created inside it inherit the same permissions?

```
[toka@localhost ~]$ mkdir /tmp/dir
[toka@localhost ~]$ chmod 2755 /tmp/dir
[toka@localhost ~]$ ls -ld /tmp/dir
drwxr-sr-x. 2 toka toka 6 Mar 28 00:39 /tmp/dir
[toka@localhost ~]$ >/tmp/dir/testFile
[toka@localhost ~]$ ls -l /tmp/dir/testFile
-rw-r--r--. 1 toka toka 0 Mar 28 00:40 /tmp/dir/testFile
[toka@localhost ~]$ su -
Password:
[root@localhost ~]# >/tmp/dir/testFile2
[root@localhost ~]# ls -l /tmp/dir/testFile2
-rw-r--r--. 1 root toka 0 Mar 28 00:41 /tmp/dir/testFile2
[root@localhost ~]#
```

14. Allow group developers to read all files in ~/Music recursively

```
[toka@localhost ~]$ setfacl -R -m g:developers:r ~/Music
[toka@localhost ~]$ getfacl ~/Music
getfacl: Removing leading '/' from absolute path names
# file: home/toka/Music
# owner: toka
# group: toka
user::rwx
group::r-x
group:developers:r--
mask::r-x
other::r-x

[toka@localhost ~]$
```

15. How to remove all ACLs from the file file1.txt and revert to the standard Unix permissions?

```
[toka@localhost ~]$ getfacl file1.txt
# file: file1.txt
# owner: toka
# group: toka
user::rw-
group::r--
mask::r--
other::r--

[toka@localhost ~]$ ls -l file1.txt
-rw-r--r--+ 1 toka toka 0 Mar 28 00:20 file1.txt
[toka@localhost ~]$ setfacl -b file1.txt
[toka@localhost ~]$ ls -l file1.txt
-rw-r--r-- 1 toka toka 0 Mar 28 00:20 file1.txt
[toka@localhost ~]$
```

16. Set a mask on myfile.txt to limit all non-owners to r--.

```
toka@localhost:~  
[toka@localhost ~]$ >file2.txt  
[toka@localhost ~]$ ls -l file2.txt  
-rw-r--r--. 1 toka toka 0 Mar 28 00:47 file2.txt  
[toka@localhost ~]$ getfacl file2.txt  
# file: file2.txt  
# owner: toka  
# group: toka  
user::rw-  
group::r--  
other::r--  
  
[toka@localhost ~]$ setfacl -m m:r-- file2.txt  
[toka@localhost ~]$ getfacl file2.txt  
# file: file2.txt  
# owner: toka  
# group: toka  
user::rw-  
group::r--  
mask::r--  
other::r--  
  
[toka@localhost ~]$
```

17. How would you allow user 'alice' to run only the 'yum' command as root?

```
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)        ALL  
nada    ALL=(ALL)        /usr/bin/yum  
## Allows members of the 'sys' group to run networking, software,  
## service management apps and more.  
  
[nada@localhost ~]$ sudo clock  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
[sudo] password for nada:  
Sorry, user nada is not allowed to execute '/sbin/clock' as root on localhost.localdomain.  
[nada@localhost ~]$ sudo yum install ksh  
[sudo] password for nada:  
Last metadata expiration check: 0:12:51 ago on Fri 28 Mar 2025 12:41:23 AM EDT.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
ksh	x86_64	3:1.0.6-6.el9	appstream	879 k

```
=====
```

18. How would you allow a group 'developers' to run any command as root without a

password?

```
##
#=====
User_Alias DEV=nada,john
Cmnd_Alias COM=/sbin/clock
#=====
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
nada    ALL=(ALL)        /usr/bin/yum
#=====
DEV     ALL=(ALL)        NOPASSWD:COM
#=====
## Allows members of the 'sys' group to run networking, software,
```

```
[nada@localhost ~]$ sudo clock
2025-03-28 01:05:57.991746-04:00
[nada@localhost ~]$ su - john
Password:
[john@localhost ~]$ sudo clock
2025-03-28 01:06:09.991072-04:00
[john@localhost ~]$
```