

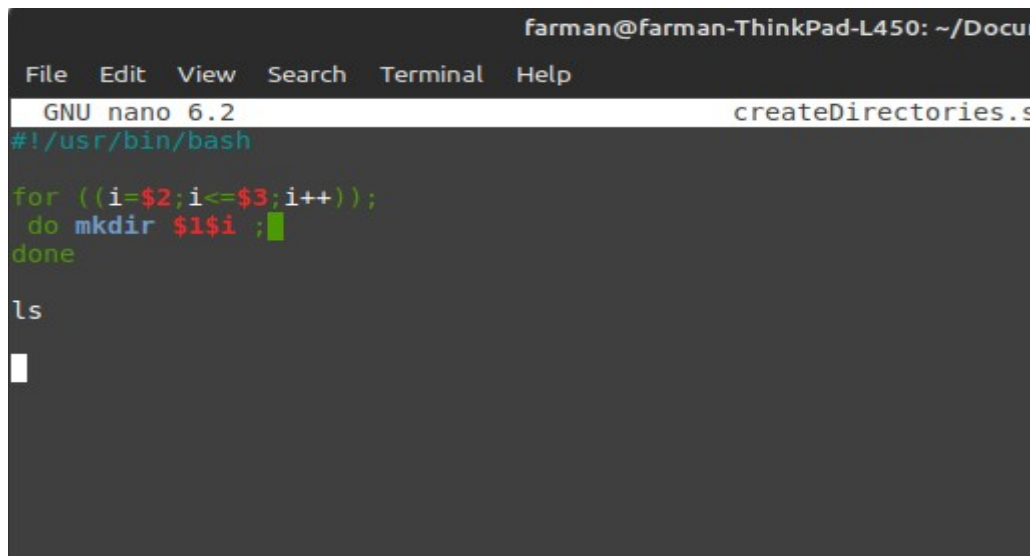
90 Days of Devops Challenge (Day 5)

Task # 1 : You have to do the same using Shell Script i.e using either Loops or command with start day and end day variables using arguments.

So Write a bash script createDirectories.sh that when the script is executed with three given arguments (one is directory name and second is start number of directories and third is the end number of directories) it creates specified number of directories with a dynamic directory name.

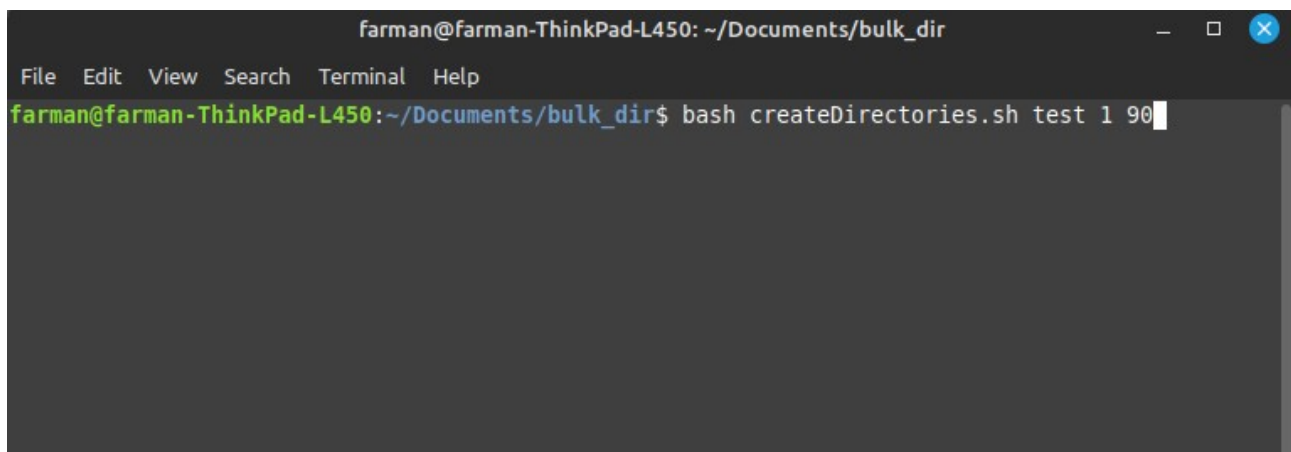
Solution: Below mention is the Script through which you can create multiple directories. You need to mention 3 arguments (1st Argument (\$1) = Directory Name, 2nd Argument (\$2) = Directory number starts with and 3rd Argument (\$3) = end number of directories)

Script:

A screenshot of a terminal window showing the nano text editor. The title bar reads 'farman@farman-ThinkPad-L450: ~/Documents'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The editor shows the file 'createDirectories.sh' with the following content:

```
#!/usr/bin/bash
for ((i=$2;i<=$3;i++));
do mkdir $1$i ;
done
ls
```

Script Command:

A screenshot of a terminal window showing the execution of the script. The title bar reads 'farman@farman-ThinkPad-L450: ~/Documents/bulk_dir'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The prompt shows the user is in the directory ~/Documents/bulk_dir. The command 'bash createDirectories.sh test 1 90' has been entered and is being executed.

```
farman@farman-ThinkPad-L450:~/Documents/bulk_dir$ bash createDirectories.sh test 1 90
```

Script Result:

```
farman@farman-ThinkPad-L450: ~/Documents/bulk_dir
File Edit View Search Terminal Help
farman@farman-ThinkPad-L450:~/Documents/bulk_dir$ bash createDirectories.sh test 1 90
createDirectories.sh test18 test27 test36 test45 test54 test63 test72 test81 test90
test1 test19 test28 test37 test46 test55 test64 test73 test82
test10 test2 test29 test38 test47 test56 test65 test74 test83
test11 test20 test3 test39 test48 test57 test66 test75 test84
test12 test21 test30 test4 test49 test58 test67 test76 test85
test13 test22 test31 test40 test5 test59 test68 test77 test86
test14 test23 test32 test41 test50 test6 test69 test78 test87
test15 test24 test33 test42 test51 test60 test7 test79 test88
test16 test25 test34 test43 test52 test61 test70 test8 test89
test17 test26 test35 test44 test53 test62 test71 test80 test9
farman@farman-ThinkPad-L450:~/Documents/bulk_dir$
```

Task # 2 : Create a Script to backup all your work done till now. Backups are an important part of DevOps Engineers day to Day activities.

Solution: Here is the Script to take backup of my 'my-daily-work' folder which is on my Desktop to the backup folder which is in my Documents.

Script:

```
farman@farman-ThinkPad-L450: ~/Documents
File Edit View Search Terminal Help
GNU nano 6.2 backups.sh
#!/usr/bin/bash

src_dir=/home/farman/Desktop/my_daily_work
tgt_dir=/home/farman/Documents/backups

curr_timestamp=$(date "+%Y-%m-%d-%H-%M-%S")
backup_file=$tgt_dir/$curr_timestamp.tgz

echo "Taking backups on $curr_timestamp"
tar czf $backup_file -P $src_dir
echo "Backup Complete"
```

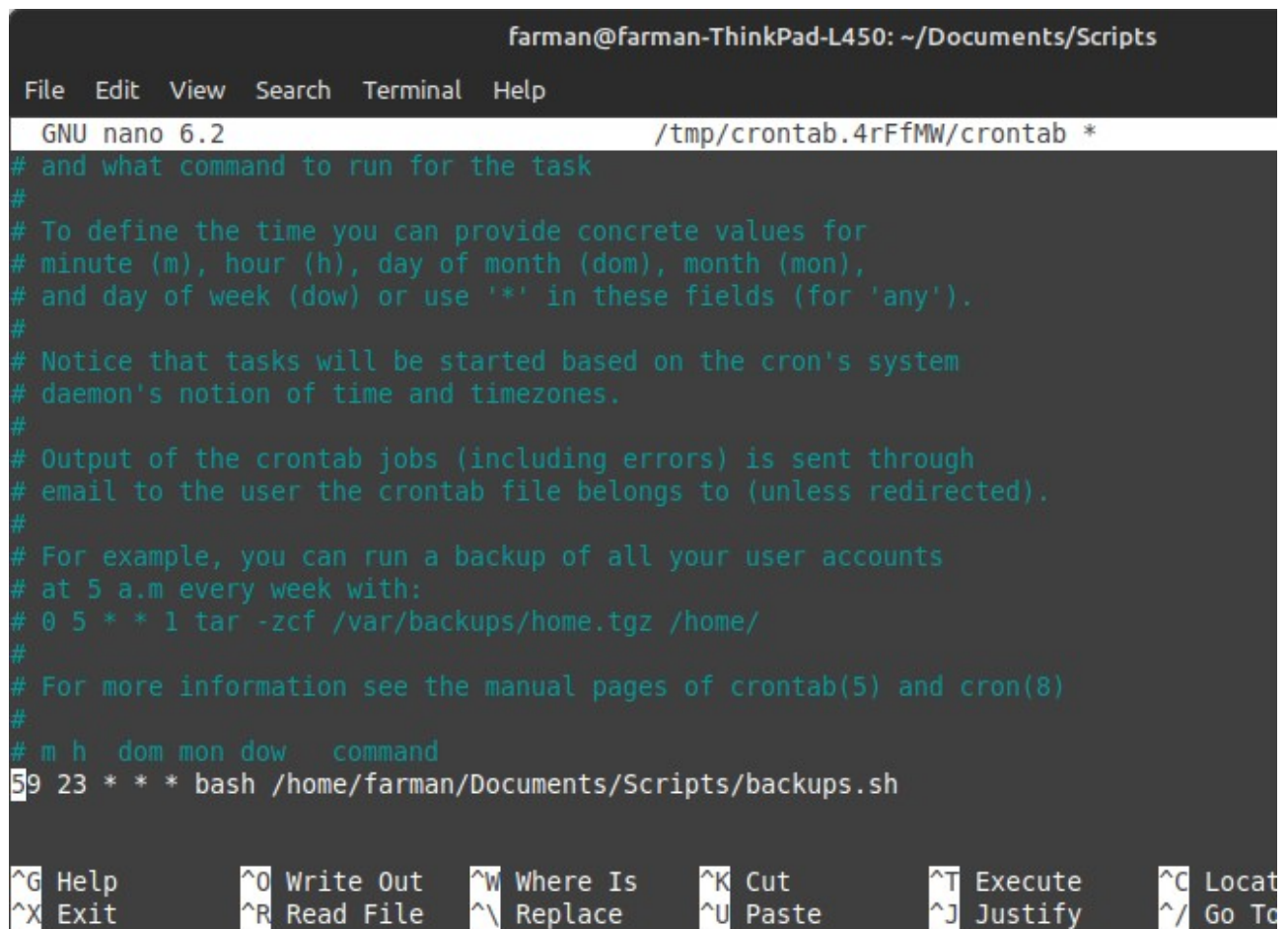
Task # 3 : Read About Cron and Crontab, to automate the backup Script.

Definition: Just like a 'Task Scheduler' in Windows, Cron is the system's main scheduler for running jobs or tasks unattended. A command called crontab allows the user to submit, edit or delete entries to cron. A crontab file is a user file that holds the scheduling information.

To View Cron Jobs : `crontab -l`

To Edit, Create or delete Crons : `crontab -e`

Scheduled a Cron to take Backups : As mention above the backup.sh script, I have created a Cron job to take backups every day at 11:59 pm.



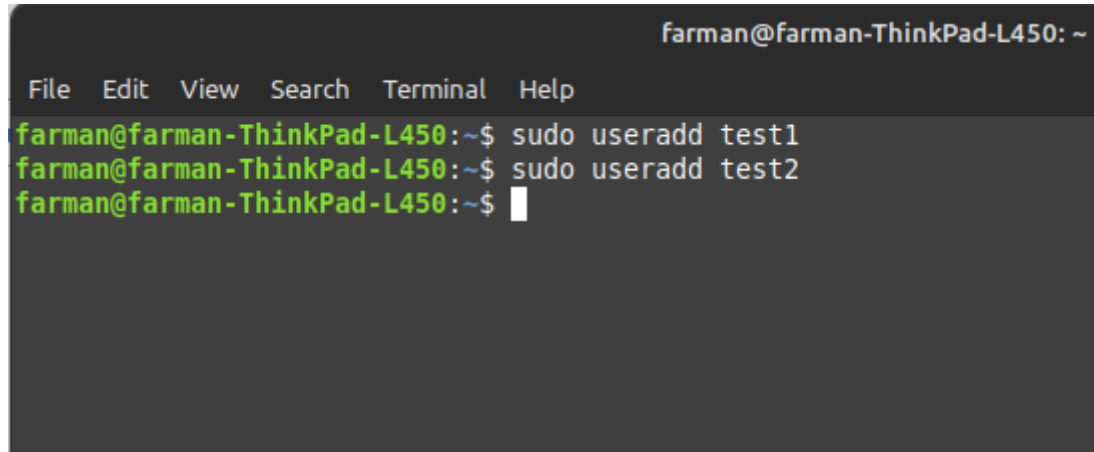
```
farman@farman-ThinkPad-L450: ~/Documents/Scripts
File Edit View Search Terminal Help
GNU nano 6.2 /tmp/crontab.4rFfMW/crontab *
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
59 23 * * * bash /home/farman/Documents/Scripts/backups.sh
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Locat
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^/ Go To
```

Task # 4 : Read about User Management and Let me know on LinkedIn if you're ready for Day 6.

Done

Task # 5 : Create 2 users and just display their Usernames:

Solution: To create a user you can use 'sudo useradd <username>' command.

A terminal window titled 'farman@farman-ThinkPad-L450: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows three lines of command execution: 'farman@farman-ThinkPad-L450:~\$ sudo useradd test1', 'farman@farman-ThinkPad-L450:~\$ sudo useradd test2', and 'farman@farman-ThinkPad-L450:~\$' followed by a cursor. The prompt and command text are green, while the output is white on a dark background.

```
farman@farman-ThinkPad-L450: ~  
File Edit View Search Terminal Help  
farman@farman-ThinkPad-L450:~$ sudo useradd test1  
farman@farman-ThinkPad-L450:~$ sudo useradd test2  
farman@farman-ThinkPad-L450:~$
```

List out all the users in Linux: You can use below mention command to display all user's usernames.

```
'awk -F':' '{ print $1}' /etc/passwd'
```