# Day 5 : Advanced Linux Shell Scripting with User management

**This is #90DaysofDevops challenge under the guidance of Shubham Londhe sir.**

Day 5 TASK

check this for task:

**https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day05/tasks.md**

**1) Write a bash script createDirectoriess1.sh that when the script is executed with three given arguments (one is the directory name and second is the start number of directories and the third is the end number of directories ) it creates a specified number of directories with a dynamic directory name.**

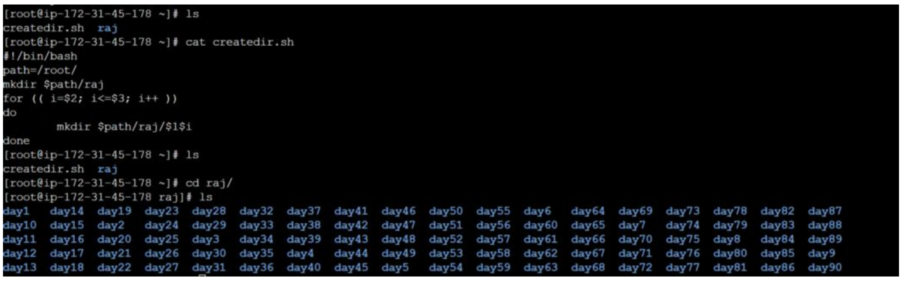
You have to do the same using Shell Script i.e using either Loops or command with start day and end day variables using arguments -

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.

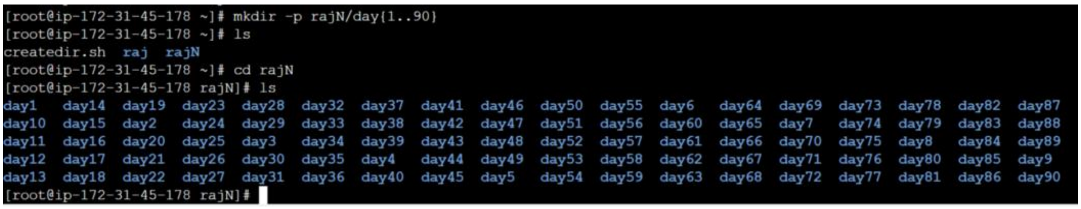
Example 1: When the script is executed as

```
./createDirectories.sh day 1 90
```

then it creates 90 directories as `day1 day2 day3 .... day90`

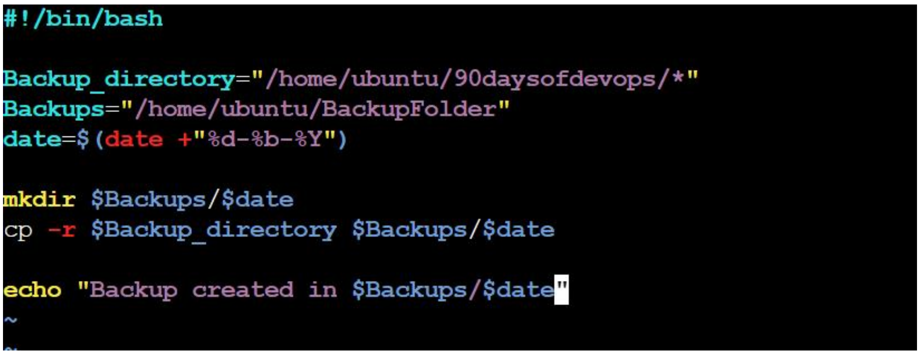


```
[root@ip-172-31-45-178 ~]# ls
createdir.sh  raj
[root@ip-172-31-45-178 ~]# cat createdir.sh
#!/bin/bash
path=/root/
mkdir $path/raj
for (( i=$2; i<=$3; i++ ))
do
        mkdir $path/raj/$1$i
done
[root@ip-172-31-45-178 ~]# ls
createdir.sh  raj
[root@ip-172-31-45-178 ~]# cd raj/
[root@ip-172-31-45-178 raj]# ls
day1   day14  day19  day23  day28  day32  day37  day41  day46  day50  day55  day6   day64  day69  day73  day78  day82  day87
day10  day15  day2   day24  day29  day33  day38  day42  day47  day51  day56  day60  day65  day7   day74  day79  day83  day88
day11  day16  day20  day25  day3   day34  day39  day43  day48  day52  day57  day61  day66  day70  day75  day8   day84  day89
day12  day17  day21  day26  day30  day35  day4   day44  day49  day53  day58  day62  day67  day71  day76  day80  day85  day9
day13  day18  day22  day27  day31  day36  day40  day45  day5   day54  day59  day63  day68  day72  day77  day81  day86  day90
```



```
[root@ip-172-31-45-178 ~]# mkdir -p rajN/day{1..90}
[root@ip-172-31-45-178 ~]# ls
createdir.sh  raj  rajN
[root@ip-172-31-45-178 ~]# cd rajN
[root@ip-172-31-45-178 rajN]# ls
day1   day14  day19  day23  day28  day32  day37  day41  day46  day50  day55  day6   day64  day69  day73  day78  day82  day87
day10  day15  day2   day24  day29  day33  day38  day42  day47  day51  day56  day60  day65  day7   day74  day79  day83  day88
day11  day16  day20  day25  day3   day34  day39  day43  day48  day52  day57  day61  day66  day70  day75  day8   day84  day89
day12  day17  day21  day26  day30  day35  day4   day44  day49  day53  day58  day62  day67  day71  day76  day80  day85  day9
day13  day18  day22  day27  day31  day36  day40  day45  day5   day54  day59  day63  day68  day72  day77  day81  day86  day90
[root@ip-172-31-45-178 rajN]# 
```

## 2) Create a Script to back up all your work done till now.



```
#!/bin/bash

Backup_directory="/home/ubuntu/90daysofdevops/*"
Backups="/home/ubuntu/BackupFolder"
date=$(date +"%d-%b-%Y")

mkdir $Backups/$date
cp -r $Backup_directory $Backups/$date

echo "Backup created in $Backups/$date"
~
~
```

## 3) Read About Cron and Crontab, to automate the Script

Cron is a Linux utility that schedules tasks to run automatically at specified intervals. The tasks are defined in a crontab file, which is a simple text file containing a list of commands meant to be run at specified times. Each line of the file represents a single cron job and follows a particular syntax.

A crontab file is composed of six fields, separated by spaces, that specify the schedule for a task to run. The fields, in order, are:

1. Minute (0–59)
2. Hour (0–23)
3. Day of the month (1–31)
4. Month (1–12)
5. Day of the week (0–7, where both 0 and 7 represent Sunday)
6. Command to be executed

You can use the command `crontab -e` to edit your crontab file and `crontab -l` to list the current crontab file.

You can also use `crontab -r` to remove your current crontab file, `crontab -d` to delete a specific user crontab and `cron -l -u <username>` to list a specific user crontab.

### 4) Read about User Management

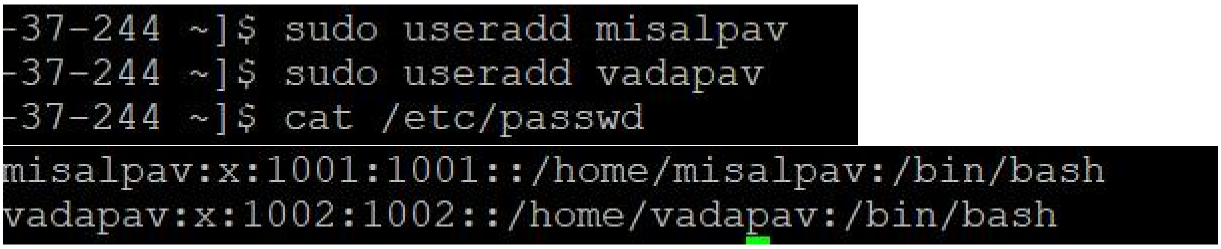
User management refers to the process of creating, modifying, and deleting user accounts on a computer system or network. It also includes managing user permissions and access to resources.

Some common tasks related to user management are:

- Creating new user accounts with a unique username and password
- Assigning different levels of privileges or permissions to users, such as access to certain files or directories
- Changing or resetting user passwords
- Disabling or deleting user accounts
- Managing groups of users and assigning group permissions
- Auditing user activity and tracking changes to user accounts

On Linux systems, user management can be done using the command line tools such as `useradd`, `usermod`, `userdel`, `passwd`, and `groupadd`, `groupmod`, `groupdel`.

### 5) Create 2 users and just display their Usernames



```
-37-244 ~]$ sudo useradd misalpav
-37-244 ~]$ sudo useradd vadapav
-37-244 ~]$ cat /etc/passwd
misalpav:x:1001:1001::/home/misalpav:/bin/bash
vadapav:x:1002:1002::/home/vadapav:/bin/bash
```

**Please, feel free to drop any questions in the comments below. I would be happy to answer them.**

**If this post was helpful, please do follow and click the clap**

**_Thank you for reading**

**_Rajani**