**TMA**

**ICT246 Operating systems.**

**Question No 1)**

I have chosen the Linux Ubuntu OS because it is to produce a modern, elegant and comfortable operating system which is both powerful and easy to use.

Linux OS has some **core functions** which are the primary reason for its existence. **Linux kernel is the core component of a GNU/Linux system.**

**It performs the following core functions:**

1. **Memory management:** The kernel allocates and deallocates memory and assigns physical memory locations based upon requests from application programs. The kernel also manages access to memory to ensure that programs only access those regions of memory which have been assigned to them.
2. **Task management:** The Linux kernel manages the execution of all tasks running on the system. The scheduler portion of the kernel allocates CPU time to each running process based on its priority and whether it is capable of running.
3. **Interprocess communication:**Interprocess communication (IPC) is vital to any multitasking operating system. Many tasks must be synchronized or communicate with each other to ensure that their work is properly coordinated.
4. **Device management:**The kernel manages access to the physical hardware through the use of device drivers. The Linux kernel manages this so that only one program actually has control of or access to a device at any given moment.
5. **I/O Management:**The kernel is also responsible for managing I/O devices. This includes USB, parallel and serial port I/O, and file system I/O. The kernel does not actually handle physical access to the disk, but rather manages the requests for disk I/O.

**The layers of the linux OS(Ubuntu) are as follow:**

1. The **kernel** is the core part of the operating system, which is  responsible for all the major activities of the LINUX operating system.
2. **System libraries** are special functions, that are used to implement the functionality of the operating system and do not require code access rights of kernel modules.
3. **System Utility** programs are liable to do individual, and specialized-level tasks.
4. **Hardware layer** of the LINUX operating system consists of peripheral devices such as RAM, HDD, CPU.
5. **The shell** is an interface between the user and the kernel, and it affords services of the kernel. It takes commands from the user and executes kernel’s functions.

**Difference Between the Ubuntu and Debian 9:**

Ubuntu and Debian are very similar, but they have some major differences too.

* Debian lacks a lot of firmware by default, you need to enable the repository yourself and install it manually **while** Ubuntu includes as much firmware as possible and tries to automatically install and configure drivers for you.
* Ubuntu has PPA(personal package archives) but Debian doesn’t.
* Both distributions use systemd and its associated subsystems.
* Both distributions use apt package management.

**References:**

* <https://www.elprocus.com/linux-operating-system/>
* <http://www.linux-databook.info/?page_id=3474>
* <https://www.quora.com/What-is-the-difference-between-Ubuntu-and-Debian>
* <https://linuxconfig.org/debian-vs-ubuntu>

**Question No 2**

I will utilize the the 9 hard disk using zfs file systems now we know that 3TB is neeeded for the marketing department. Also it will be only accessible to root and the marketing department. 7TB is needed for R&D department. Now each staff will be given 10GB so for 120 staff It will be 1.2TB . 1.5TB for file and non-sensitive data sharing. Now in my opinion i will utilize the drives according to needs and will give each of them the required space. I will create pools for each of them. I would recommend RAIDZ2 for R&D because of the additional protection it affords in case one drive somehow breaks or otherwise develops problems. RAID-Z2 would require at least 4 disks per vdev and can tolerate up to 2 disk failure and if the third disk fails before the 2 disks are replaced your valuable data is lost. A mirrored pool provides you the redundancy which enables us to store multiple copies of data on different disks. Here you can also detach a disk from the pool as the data will be available on the another disks.

To manage all these drives we can use the ZFS so that we can get physical backup and data redundancy also.It includes pooled storage (zpool), copy-on-write, snapshots, data integrity verification, automatic repair, protection against corruption, etc.

Now for the R&D department we will use 4 drives of 2TB as it requires 7TB so it will have an extra 1TB for future if needed. So a RAID-Z2 pool will be created. Create a pool of 4 drives and split the data onto the drives.

For marketing department 2 hard drives will be used in the pool so that total of 4TB will be used. You can RAID-Z1 as it requires less drives than RAID-Z2. Create a pool of these drives and split the data over them.

For staff create a pool and then create 10GB volume of virtual disk by using zfs command:

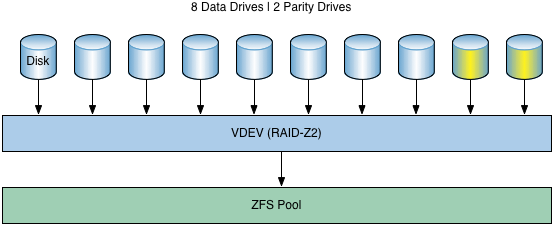
zfs create -V 10G name/virtualdisk

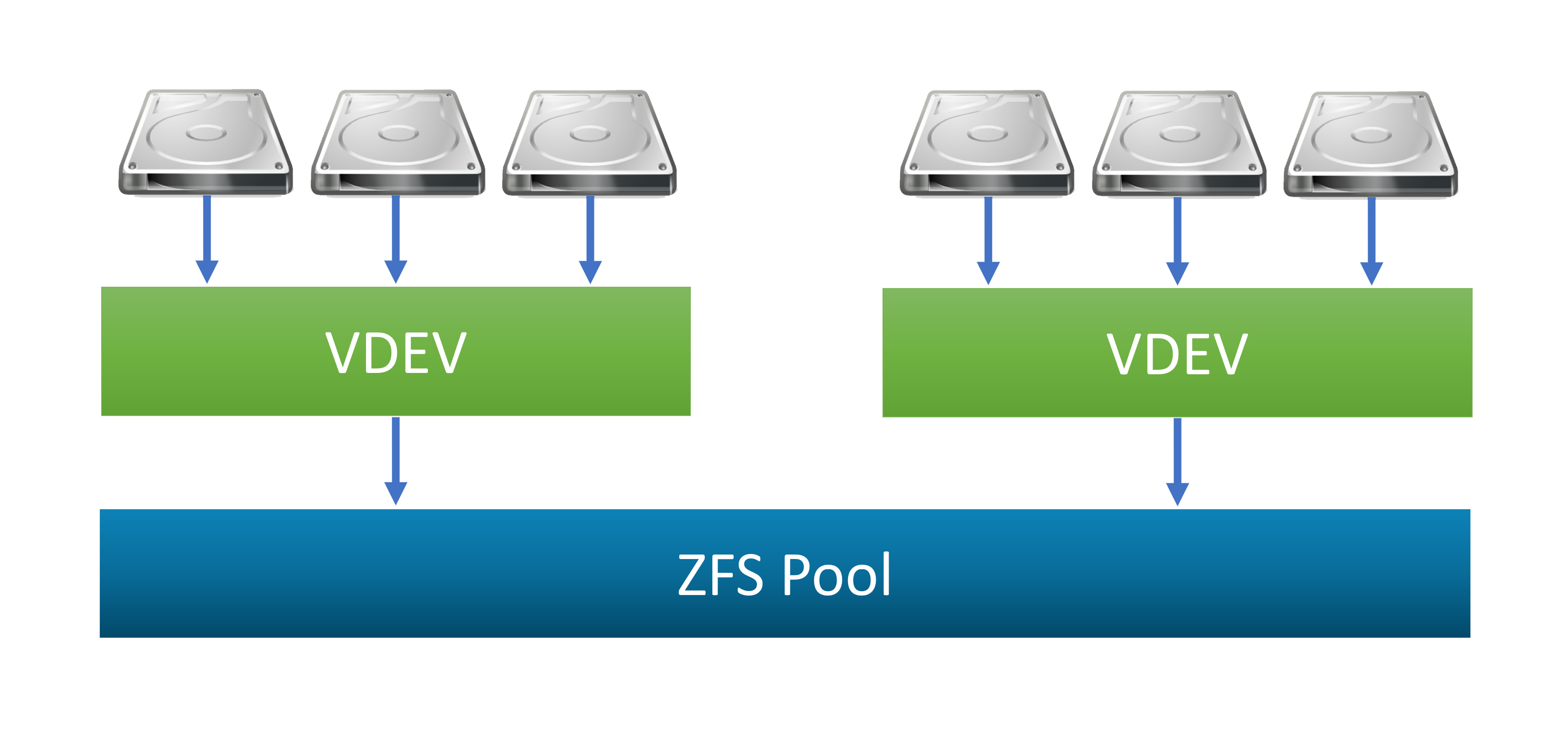
to see the list we can use command:

zfs list

For temporary data sharing a single drive will be used for first year after 1 year it is enough because 20% extra will only require 300 more gb that means 2TB is enough for 2 years . Now for 3rd year you can add drives to a pool to increase its capacity. Any new data will be dynamically striped across the pool, but existing data will not be moved in order to "balance" the pool.

A number of screenshots and disk distributions that will be used according to my explanation:





**Question No 3)**

**Part A)**

**#!bin/bash**

**if [ -d /etc/Folder ]**

**then**

**cd /etc**

**while true; do**

**read -p "Enter Yes or No if you want read write the whole folder or not : " inp**

**if [ "$inp" = "Yes" -o "$inp" = "yes" ]**

**then**

**rm -rf Folder**

**mkdir Folder**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**cat passwd >> $a**

**mv $a Folder/**

**break**

**### Can't re-making of floder**

**elif [ "$inp" = "No" -o "$inp" = "no" ]**

**then**

**cd Folder**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**if [ -e $a ]**

**then**

**while true; do**

**read -p "Enter Yes or No: if you want to rewrite the file passwrd-$currentdate.txt " inp**

**if [ "$inp" = "yes" -o "$inp" = "yes" ]**

**then**

**cd ../**

**rm -rf Folder**

**mkdir Folder**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**cat passwd >> $a**

**mv $a Folder/**

**echo "I want to re write the file"**

**break**

**elif [ "$inp" = "No" -o "$inp" = "no" ]**

**then**

**break**

**fi**

**done**

**else**

**cd /etc**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**cat passwd >> $a**

**mv $a Folder/**

**fi**

**break;**

**fi**

**done**

**else**

**cd /etc**

**mkdir Folder**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**cat passwd >> $a**

**mv $a Folder/**

**fi**

**cd Folder**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**chmod 700 $a**

**Part B and C**

**#!bin/bash**

**# Just taking the Enters**

**echo "Enter Usename" # Print the line**

**read username # Reading**

**echo "Enter userId"**

**read userid**

**echo "Enter GroupId"**

**read groupid**

**echo "Enter User Info"**

**read UserInfo**

**#chech if the user alread exist or not**

**cat /etc/passwd | grep ${username} >/dev/null 2>&1**

**if [ $? -eq 0 ] ; then**

**echo "User Exists"**

**else**

**echo "User Not Found"**

**echo "Adding user"**

**# Changing the DIrectory**

**cd /etc**

**#copying into the file passwd**

**echo "$username:x:$userid:$groupid:$username,,,:/home/$username:/bin/bash" >> passwd**

**fi**

**#moving back into the root**

**cd ../**

**#moving into dirctory name home**

**cd home**

**#making the Directory of the new user**

**mkdir $username**

**#giving the permissions**

**chmod -R 731 $username**

**#change the Password**

**passwd $username**

**Documentation:**

The piece of code is a simple adding user utility manually coded in part A when the code is executed

**It first checks that If the directory already exists** in the condition of if statement now if statement becomes true that a file exist then an input is required from the user that if he wants to overwrite the etc/passwd-DDMMYYYY the file or not. Now input is saved in to a variable then this variable is check in **if statements** if the user says Yes it makes the directory named folder and then add the current date into it using the piece of code:

**rm -rf Folder**

**mkdir Folder**

**currentdate=$(date +%d-%m-%Y)**

**a=passwrd-$currentdate.txt**

**cat passwd >> $a**

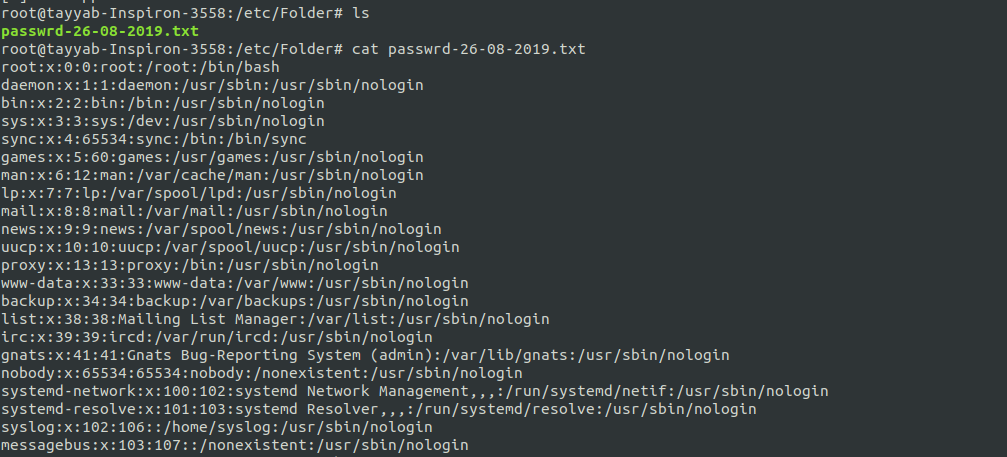
**mv $a Folder/**

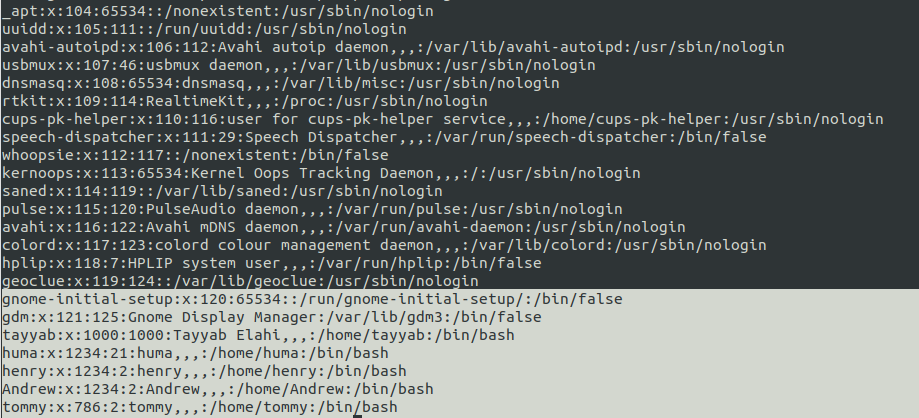
**break**

And then breaks the loop and if the user inputs the No then it shows the message. And if the user input something else other then [yes,Yes,No,no] then It will keep asking the user to input the right string so that if statement should pass.

**Now if the file doesn’t exist** then it will make the file with current date passwd-DDMMYYYY and then copy all the content of passwd into it and then it will chage the permission of this file with **CHMOD 700** so that only root have access to it.

**Part A screenshot:**





**Now in part B** it asks the user to input everything asked in the prompt then it will add this information to a new line in passwd using this **command:**

**echo "$username:x:$userid:$groupid:$username,,,:/home/$username:/bin/bash" >> passwd**

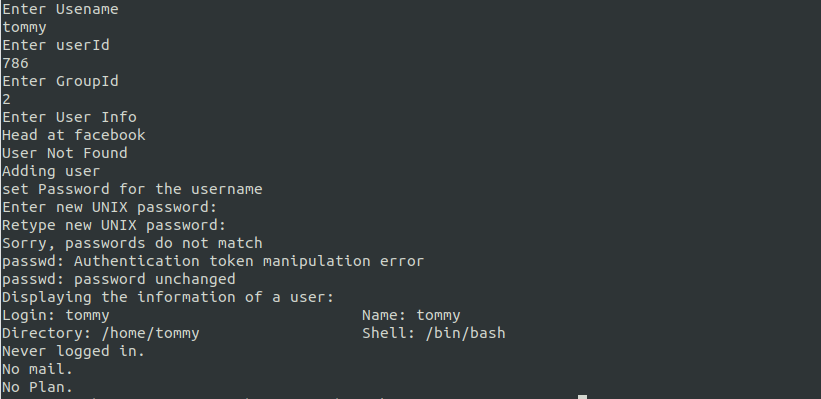
Now after this it go back to home and then make the directory of new user and will give the user the permissions of this directory **using:**

**chmod -R 731 $username**

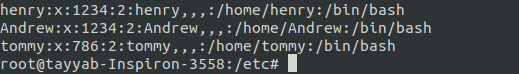
and then user will be asked to change the password.

**In part C** it will go into the passwd to check that if the user added already exists or not. If it exists then it will say the user exists and then it will display the info of user from both passwd and shadow file.

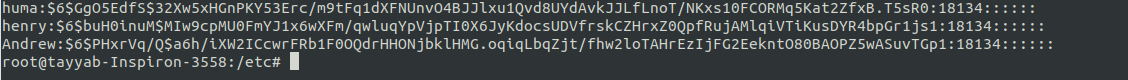
**The part B and C** are merged to test some use cases here are the **screenshot:**

**Adding new user:**

**Displaying the etc/passwd file:**

****

**Displaying the etc/shadow file:**

****