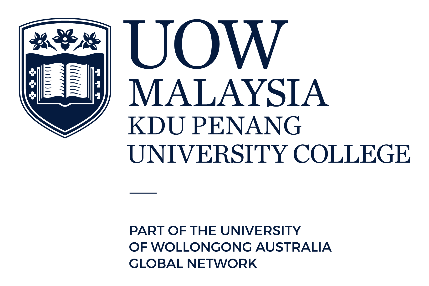
****

|  |  |  |  |
| --- | --- | --- | --- |
| **Programme** | | **Course Code and Title** | |
| Diploma in Information Technology | | DSM2294 Systems Administration & Management | |
| **Students Name & ID** | | **Lecturer Name:** Danny Chen | |
| 1. 0204677 Lim Zhe Yuan 2. 0205666 Roy Lim Chen Rui 3. 0205430 Tan Peng Heng 4. 0205096 Thor Wen Zheng | |  | |
|  | | | |
| **Date of Assignment Release** | **Submission Deadline** | | **Indicative Weighting** |
| 20/09/2022 | 11.59pm, Friday, 02/12/2022 | | 100 Marks, Weighted @ 30% |

|  |  |
| --- | --- |
| **Assignment title** | Assignment – Linux Setup & Permissions |

|  |
| --- |
| **Student’s declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Students signatures: Date: 2nd December 2022   1. *Zhe Yuan* 2. *Roy Lim* 3. *Peng Heng* 4. Wen Zheng |

Plagiarism

The assignment is based on an individual response. The report must be **completely your own work** and you must not copy from others. Any plagiarized work will be zero-rated. Any reference material you use (books, journals, Internet, magazines etc.) must be clearly identified in your report using procedures in the Harvard System of Referencing.

**Table of Contents**

**Part 1: Linux Setup1**

**Part 2: File Permission7**

**Part 3: Backup Directory Creation and Management25**

**Part 4: Process Management34**

**Part 5: Networking Concepts38**

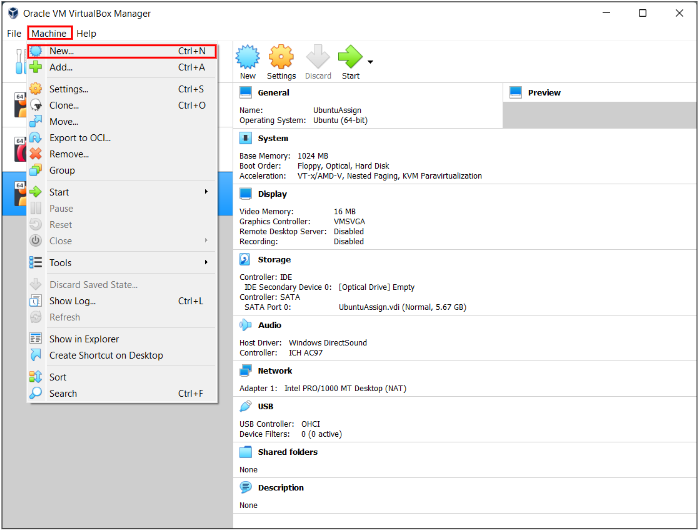
**References54**

**Part 1: Linux Setup**



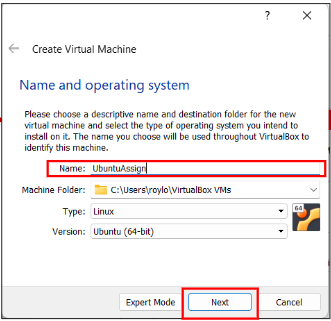
***Figure 1.1:*** *Oracle VM Virtual Machine installation website*

**Step 1:** The first step is to install Oracle VM VirtualBox from <https://www.virtualbox.org/>. VirtualBox is a cross-platform virtualization software which allows users to extend their computer virtually to run multiple Operating Systems such as Microsoft Windows and Linux.



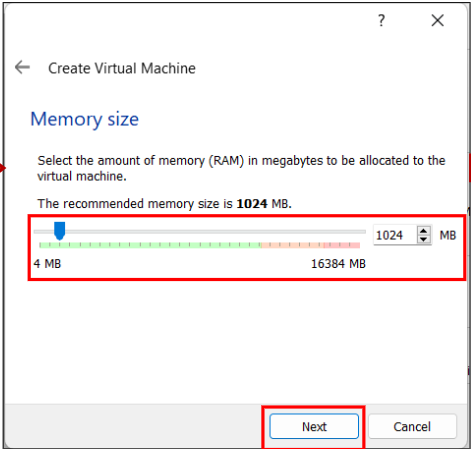
***Figure 1.2:*** *Creating a new virtual machine*

**Step 2:** After finishing installation of the Oracle VM VirtualBox, click on the “Machine” menu and select “New” to create a new virtual machine.



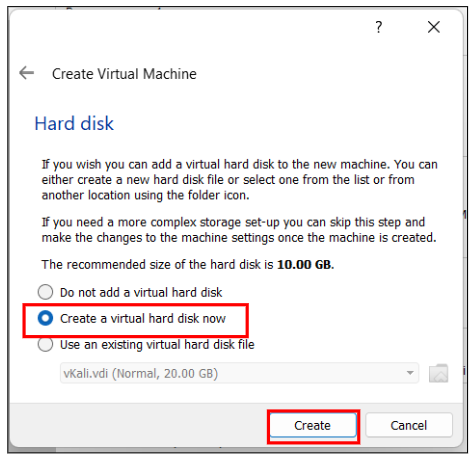
***Figure 1.3:*** *Choosing a suitable name for Virtual Machine*

**Step 3:** After selecting the “New” menu button from “Machine” menu, insert a suitable name for the Virtual Machine. Next, select “Linux” in the “Type” dropdown, select “Ubuntu (64-bit)” in the “Version” dropdown, and click on the “Next” button.



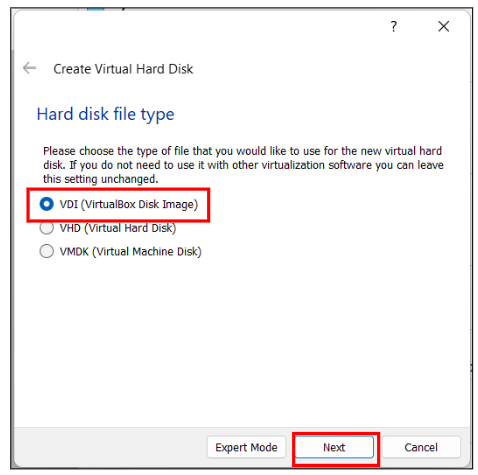
***Figure 1.4:*** *Selecting suitable memory size for Virtual Machine*

**Step 4:** After giving a name for the Virtual Machine, choose a suitable memory size for the Virtual Machine and click on “Next” button. The memory size given should depend on how much RAM the host machine has and should not be too little nor too much.



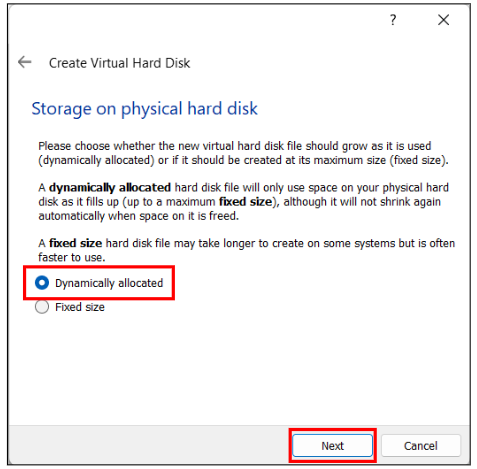
***Figure 1.5:*** *Choosing virtual hard disk*

**Step 5:** After choosing a suitable memory size, choose the “Create a virtual hard disk now” option and click on the “Create” button to proceed.



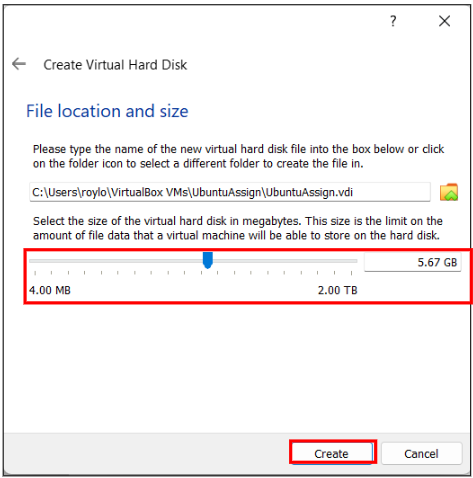
***Figure 1.6:*** *Choosing virtual hard disk type*

**Step 6:** After creating the virtual hard disk, choose the “VDI (VirtualBox Disk Image)” option for the hard disk file type and click on the “Next” button.



***Figure 1.7:*** *Type of storage on physical hard disk*

**Step 7:** After finishing choosing the hard disk file type, choose the “Dynamically allocated” option as for the storage on physical hard disk and click on the “Next” button.



***Figure 1.8:*** *File Location and Size*

**Step 8:** Lastly, for the file location and size, choose a suitable location to store the Virtual Machine’s file. Then, select the most suitable size for the virtual storage to store the data in the virtual hard disk. Finally, click on the “Create” button to finish the virtual machine setup process.

**Graphical user interface, application

Description automatically generated**

***Figure 1.9:*** *Starting the virtual machine*

**Step 9:** After finishing the setup of the virtual machine, select the virtual machine and click on “Start” to start the virtual machine.

**Part 2: File Permissions**

1. **Create 9 users and 3 new groups**

**Commands:**

sudo adduser user1

sudo adduser user2

sudo adduser user3

sudo adduser user4

sudo adduser user5

sudo adduser user6

sudo adduser user7

sudo adduser user8

sudo adduser user9

Firstly, “sudo adduser” is used to add a new user by specifying the name of the new user after the command. For example, “sudo adduser userx” commands the system to add a new user with the name “userx”. Next, the system will require the administrator to input their password to continue the process. Once the admin is verified, the system requires the administrator to input user credentials for the new user, such as phone numbers, confirming their user group, password, phone numbers, and other information. Lastly, the system accepts the given inputs and finishes up the creation of the new user. **Figure 2.1.1** to **Figure 2.1.9** show the processes of creating user accounts for 9 new users.

**Text

Description automatically generated**

***Figure 2.1.1:*** *Process of creating user1*

***Text

Description automatically generated***

***Figure 2.1.2:*** *Process of creating user2*

***Text

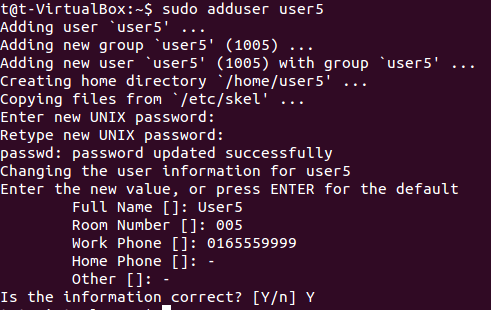
Description automatically generated***

***Figure 2.1.3:*** *Process of creating user3*

***Text

Description automatically generated***

***Figure 2.1.4:*** *Process of creating user4*

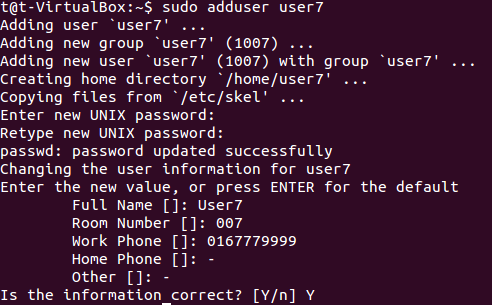
******

***Figure 2.1.5:*** *Process of creating user5*

***Text

Description automatically generated***

***Figure 2.1.6:*** *Process of creating user6*

******

***Figure 2.1.7:*** *Process of creating user7*

***Text

Description automatically generated***

***Figure 2.1.8:*** *Process of creating user8*

***Text

Description automatically generated***

***Figure 2.1.9:*** *Process of creating user9*

**Commands:**

sudo addgroup group1

sudo addgroup group2

sudo addgroup group3

Next, the “sudo addgroup” command is used to create and add new user groups by specifying the group name after the command. For example, “sudo addgroup groupx” is used to create a new user group named “groupx”. A group ID will be assigned to a new user group when it is created. **Figure 2.1.10** to **Figure 2.1.12** show the processes of creating and adding 3 new user groups.

***Text

Description automatically generated***

***Figure 2.1.10:*** *Process of creating group1*

***A screenshot of a computer

Description automatically generated with low confidence***

***Figure 2.1.11:*** *Process of creating group2*

***Text

Description automatically generated***

***Figure 2.1.12:*** *Process of creating group3*

1. **Let 4 users be in the first group, 3 users in a second group and the last 2 users in the third group**

**Commands:**

sudo adduser user1 group1

sudo adduser user2 group1

sudo adduser user3 group1

sudo adduser user4 group1

sudo adduser user5 group2

sudo adduser user6 group2

sudo adduser user7 group2

sudo adduser user8 group3

sudo adduser user9 group3

The “sudo adduser” command is used for adding existing users to existing user groups. It is done by specifying a target user followed by the target group that the target user should be added to. For instance, “sudo adduser userx groupx” adds the user “userx” to the “groupx” group. **Figure 2.2.1** to **Figure 2.2.9** show the processes of distributing all users that were created previously into the user groups that were created.

**Text

Description automatically generated**

***Figure 2.2.1:*** *Process of adding user1 to group1*

***Text

Description automatically generated***

***Figure 2.2.2:*** *Process of adding user2 to group1*

***Text

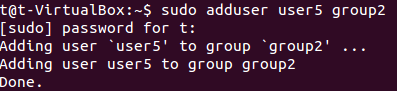
Description automatically generated***

***Figure 2.2.3:*** *Process of adding user3 to group1*

***Text

Description automatically generated***

***Figure 2.2.4:*** *Process of adding user4 to group1*

******

***Figure 2.2.5:*** *Process of adding user5 to group2*

***Text

Description automatically generated***

***Figure 2.2.6:*** *Process of adding user6 to group2*

***Text

Description automatically generated***

***Figure 2.2.7:*** *Process of adding user7 to group2*

***Text

Description automatically generated***

***Figure 2.2.8:*** *Process of adding user8 to group3*

***Text

Description automatically generated***

***Figure 2.2.9:*** *Process of adding user9 to group3*

1. **Create 1 file accessible by all of the 9 users above**

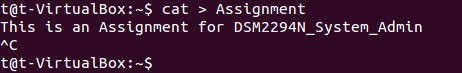
**Commands:**

cat > Assignment

chmod a+rwx Assignment

ls -l Assignment

The “cat > [filename]” command allows the administrator to create a new .txt file. After writing file contents, administrators can press “CTRL+C” to close and save the .txt file. **Figure 2.3.1** shows the process of creating a new .txt file and adding text content into the file.

****

***Figure 2.3.1:*** *Process of creating new .txt file with content in file*

Then, the “chmod a+rwx Assignment” command is used to change the file permission of the newly created Assignment.txt file. The “a+rwx” parameter tells the command to modify file permissions for all users and allow them to read, write, and execute the file, giving them full access to the file.

******

***Figure 2.3.2:*** *Command used to modify file permissions for all users*

Lastly, the command “ls -l Assignment” is used to list out the current file permissions of the Assignment.txt file to check whether the changes for Assignment.txt is valid. In **Figure 2.3.3**, the first column, “-rwxrwxrwx” provides information about file permissions to the administrator. Firstly, “–“ indicates that the item is a file, and 3 columns of “rwx” refers to the read, write, execute permission for owner, group, and other users respectively. As seen in **Figure 2.3.3**, the Assignment.txt file is accessible by all 9 users created earlier because all users have read, write and execute access regardless of which user group they are in.

******

***Figure 2.3.3:*** *Checking file permissions of the Assignment.txt file*

1. **Create 1 folder for each of the group**

**Commands:**

sudo mkdir Group1

sudo chgrp group1 Group1

sudo mkdir Group2

sudo chgrp group2 Group2

sudo mkdir Group3

sudo chgrp group3 Group3

The command “sudo mkdir [folder name]” creates a new directory and the command “sudo chgrp [group name] [file/folder name]” assigns a target user group to a specified file or directory. By combining both commands, administrators can create new directories and assign a specific user group to the new directory to use. It modifies a directory to belong to the specified group and folder permissions can be configured for the assigned user group instead of assigning permissions to users individually. **Figure 2.4.1** to **Figure 2.4.4**show the processes involvedto create group-specific directories.

**A screenshot of a computer

Description automatically generated with low confidence**

***Figure 2.4.1:*** *Process of creating new directory and assigning to group1*

***Text

Description automatically generated***

***Figure 2.4.2:*** *Process of creating new directory and assigning to group2*



***Figure 2.4.3:*** *Process of creating new directory and assign to group3*

**Commands:**

sudo chmod o= Group1

sudo chmod o= Group2

sudo chmod o= Group3

The “chmod” command is then used to modify the access permissions of the created folders. The full command “sudo chmod o= [folder name]” specifically makes it so that users who are not the folder owner, and users who do not long belong to the owner group of the specific folder are unable to access it. **Figures 2.4.4** to **2.4.5** show the process of removing access permissions to the folders for users other than the folder owners and the members of the owner group.

A screenshot of a computer

Description automatically generated with low confidence

***Figure 2.4.4:*** *Process of removing access permissions for other users to the created folders*

***A screenshot of a computer

Description automatically generated with medium confidence***

***Figure 2.4.5:*** *Results of removing access permission for other users*

1. **Create 1 more folder called public where all users can have access to it**

Using the “sudo mkdir” and “chmod” commands, the “public” folder can be created and file permissions for it can be modified. In this case, “a+rwx” should be used to enable read, write and execute access for all users including the owner, members of the owner group, and other users to access the file. The figures below show the creation of the “public” folder and the file permissions of the folder after modification.

******

***Figure 2.5.1:*** *Process of creating a public directory and giving access to all users*

******

***Figure 2.5.2:*** *File permissions of the “public” folder*

1. **Show how access permissions affect users with read, write, and execute modes and also user, group and other categories using different CHMOD command syntax.**

A sample text file and a sample directory are first created to show how access permissions affect user access to files and directories. The file only has read permission set for the owner and no permissions set for other user types. On the other hand, the directory contains subdirectories and only has read and execute permissions set for the owner and no permissions set for other user types.



***Figure 2.6.1:*** *File permissions for testFile.txt*

******

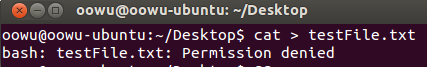
***Text

Description automatically generated***

***Figure 2.6.2:*** *Folder contents and permissions for testDir/ directroy*

* 1. **Add permission (+)**

When the owner tries to write to the testFile.txt file using any text editor, the system denies owner access to the file as they do not have write permission. In this example, the system denies owner access to the file when the “cat” command is used to write contents to the file.



***Figure 2.6.3:*** *Owner access denied to testFile.txt when trying to write the file*

File contents can only be accessed and modified after adding write permissions for the owner to the testFile.txt file. The figures below show that the owner is able to modify file contents after changing testFile.txt owner’s file permissions using the “chmod” command.

**Command:**

sudo chmod u+w testFile.txt

Text

Description automatically generated

***Figure 2.6.4:*** *Write access given to owner for testFile.txt*

Text

Description automatically generated

***Figure 2.6.5:*** *Owner becomes able to write to testFile.txt*

* 1. **Remove permission (–)**

Using the “testDir” directory, owners can view subdirectories inside the directory because the execute permission is set for the owner.

When the owner’s execute permission is removed and the owner tries to open the directory, the system denies the owner access to the directory’s contents. The owner can no longer view text files and view contents of subdirectories in the main directory. The following figure shows the scenarios when the owner does and does not have execute permission respectively.

**Command:**

sudo chmod o-x testDir

***Text

Description automatically generated***

A picture containing graphical user interface

Description automatically generated

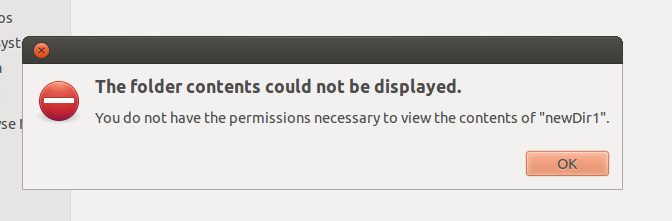
***Figure 2.6.6:*** *Directory contents are accessible when owner has execute permission*

Text

Description automatically generated

Graphical user interface, application, website

Description automatically generated



***Figure 2.6.7:*** *Owner access to directory contents is denied when owner does not have execute permissions*

* 1. **Assign permission (=)**

If another user from the same user group of testFile.txt who is not the file owner attempts to read and write to the testFile.txt file, the system denies them access to the file because they do not have read or write permissions set for their user group. To simulate this, the testFile.txt’s user group is changed to “group1” instead, and another user who is not the file owner but belongs to the same user group is used to access the file.



Graphical user interface, application

Description automatically generated

***Figure 2.6.8:*** *User group of testFile.txt and the logged in user is changed*

Graphical user interface, text, application

Description automatically generated

***Figure 2.6.9:*** *testFile.txt is inaccessible by User1*

To enable file access to testFile.txt for other users in the “group1” user group, the testFile.txt owner’s file permissions can be copied and assigned to the user group permissions. By doing this, all members in “group1” will then be able to read and write to the file.

**Command:**

sudo chmod g=u testFile.txt

Text

Description automatically generated

***Figure 2.6.10:*** *Assigning the same owner file permissions for user group to testFile.txt*

Graphical user interface, text, application

Description automatically generated

***Figure 2.6.11:*** *testFile.txt becomes accessible using User1*

* 1. **Permission in octal numbers**

Aside from using symbolic characters to set permissions for files and directories, administrators can also use octal values to denote a specific set of permissions to configure permissions for each user type sequentially. The following table shows the numerical representation of each permission combination that administrators can specify for the “chmod” command.

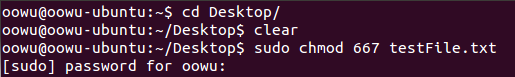
|  |  |
| --- | --- |
| **Numeric value** | **Permissions set in RWX format** |
| **0** | **- - -** |
| **1** | **- - x** |
| **2** | **- w -** |
| **3** | **- w x** |
| **4** | **r - -** |
| **5** | **r – x** |
| **6** | **r w -** |
| **7** | **r w x** |

***Table 2.6.1:*** *Octal representation of permission sets*

Using testFile.txt, if administrators want to maintain the current file permissions and enable all access for other users that is neither the file owner nor a member of the file’s user group, “sudo chmod 667 testFile.txt” is the command that should be used to produce the desired result. **Figure 2.6.3.1** shows the result of using the given command.

**Command:**

sudo chmod 667 testFile.txt





***Figure 2.6.12:*** *Changing file permissions using octal representations*

**Part 3: Backup Directory Creation and Management**

1. **Using Linux commands, create a directory called testwww at /var. Then create any file inside that directory. Then create another directory at root (/) called backup.**

**Commands:**

sudo mkdir /var/testwww

cd /var/testwww

sudo vi testfile.txt

:wq (to write and quit from new file)

sudo mkdir /backup

**Explanation:**

Text

Description automatically generated

***Figure*** ***3.1.1:*** *Creating the /testwww directory in /var directory*

The **mkdir** command is used to create the new */testwww* directory inside the */var* directory. Since */var* is a system root directory that is being modified, the system will prevent the user from creating directories if they do not have administrative rights. For users to authorize themselves, users need to use the **sudo** command before their input commands to grant administrative access to system files and directories. In this case, **sudo** was used to grant administrative privileges to the **mkdir** command. After executing the command, the */testwww* directory is successfully created within the */var* directory as shown in ***Figure*** ***3.1.1***.

Then, the **cd** command is used to change the present working directory of the current terminal session. The command is used to redirect user from the home directory (~) to the */testwww* directory path, which is */var/testwww*.

Text

Description automatically generated

***Figure*** ***3.1.2:*** *Changing working directories to /var/testwww directory*

To create a text file, the **vi** command isused to create a new file using the Vi text editor. It allows users to read and write files within the command line interface. Since we are creating a new file within a system directory, the **sudo** command is needed again to grant write access privilege for creating the file in the directory. In the Figure below, a new text file named *testFile.txt* is created in the */testwww* directory.

Text

Description automatically generated

Text

Description automatically generated with medium confidence

***Figure*** ***3.1.3:*** *Creation of text file using Vi editor*

When creating the file using Vi editor, the user will be redirected to Vi editor’s interface to write into the new file. Using Vi editor’s command mode, the **:wq** command is used to save the file and close Vi editor from the command line interface without writing any content to the text file.

Text

Description automatically generated

***Figure*** ***3.1.4:*** *Creating /backup directory in root directory*

Lastly, a */backup* directory is created in the root directory (/) using the same method that was used to create the */var/testwww* directory.

1. **Use the Tar command to perform backup on the testwww directory and stored the backup file at the backup directory under root.**

**Commands:**

sudo tar -cvpzf /backup/backup.tar.gz /var/testwww

**Explanation:**

**Text

Description automatically generated**

***Figure*** ***3.2.1:*** *Creating backup.tar.gz archive in /backup directory*

The **tar** command is used to create archives of files and directories. The archive file format, *.tar.gz*, indicates that tar was used to archive multiple file components into a single file and GZIP was used to perform compression to reduce resultant file sizes (WinZip, 2022). Since the root directory is being modified again, the **sudo** command is needed for the **tar** command.

To create a backup archive file in the */backup* directory created earlier, the **tar** command needs to specify 3 different arguments: the **tar** command options, the backup file name, and the path of the directory for the archiving process.

The **tar** command utilizes 5 options to create a new archive. Each options are mapped to their functionalities as described below:

|  |  |
| --- | --- |
| **Option** | **Description** |
| **c** | Create or overwrite an archive |
| **v** | Expose archiving process verbosely in the command line |
| **p** | Preserve file and directory permissions |
| **z** | Use compression on archive |
| **f** | Allow users to give a name to the backup file |

***Table 3.2.1:*** *Explanations for**command options*

After declaring **tar** command options, a filename for the archive which uses the *.tar.gz* file format and the start path are also specified for the command. In this case, the file name for the archive is *backup.tar.gz* and the directory path to start the archiving process will be */var/testwww*. After the process is done, *backup.tar.gz* is successfully created in the */backup* directory.

1. **Use the same command as above to perform an automatic task in which the backup will occur every Saturday at 3.00am.**

**Commands:**

sudo crontab -e

**In crontab file:**

0 3 \* \* 6 sudo tar -cvpzf /backup/backup.tar.gz /var/testwww

**Explanation:**

**Text

Description automatically generated**

***Figure*** ***3.3.1:*** *Command for opening crontab file for adding cron jobs*

To schedule an automatic task or also called a cron job, the **crontab** command is used to specify cron jobs that should be executed by their specified time. The **crontab -e** command opens a text editor for users to edit additional cron jobs that they want. Since the cron job that should be executed is a command that modifies the system root directory, the **sudo** command is required to allow cron jobs to be executed using administrator’s credentials instead of normal user’s.

**Text

Description automatically generated**

***Figure*** ***3.3.2:*** *Crontab file*

To specify a cron job, the users need to include the actual minutes, hours, day of month, month, and day of week successively for the cron job, as well as the command to execute. ‘\*’ is used to ignore unnecessary date and time fields. In this case, ignoring other fields, the *minute* field is set to 0, the *hour* field is set to 3, and the *day of week* field is set to 6 to indicate that the cron job for backing up */var/testwww* data should execute at exactly 3.00am every Saturday.

**Text

Description automatically generated**

***Figure*** ***3.3.3:*** *Adding the backup command as a cron job in the crontab file*

**Text

Description automatically generated**

***Figure*** ***3.3.4:*** *Output for**successfully adding cron job*

1. **Create a directory called recover at root. Use tar command to extract the backup file at the backup directory and save it at recover directory.**

**Commands:**

sudo mkdir /recover

sudo tar -xvpzf /backup/backup.tar.gz -C /recover

**Explanation:**

Text

Description automatically generated

***Figure*** ***3.4.1:*** *Creating the /recover directory*

First, a */recover* directory is created using the **sudo mkdir** command that was used to create the */backup* directory. Then, the **tar** command is used again to extract contents of a specified archive into the destination path. As shown in ***Figure 3.4.2***, the **tar** command is not only used for creating archives, but also used for extracting their contents. This is accomplished by replacing the **c** option in the **tar** command with the **x** option. While the **c** option tells the **tar** command to create archive files, the **x option** allows the extraction of a given archive file.

Aside from command options in the **tar** command used for extracting archives, target archive filenames are also a required parameter to specify the archive file that the **tar** utility needs to extract from. Optionally, users can change and specify the destination path of the extraction using the **-C** flag. When opted out, the results of the extraction are inserted into the present working directory of the current terminal session. In this case, */backup/backup.tar.gz* was specified as the target archive to extract from and the extraction destination path, */recover*, is set for the -**C** flag to allow extraction results to be inserted into the */recover* directory in the root directory.

Text

Description automatically generated ***Figure*** ***3.4.2:*** *Extracting the archive file into the /recover directory and the output result*

**Part 4: Process Management**

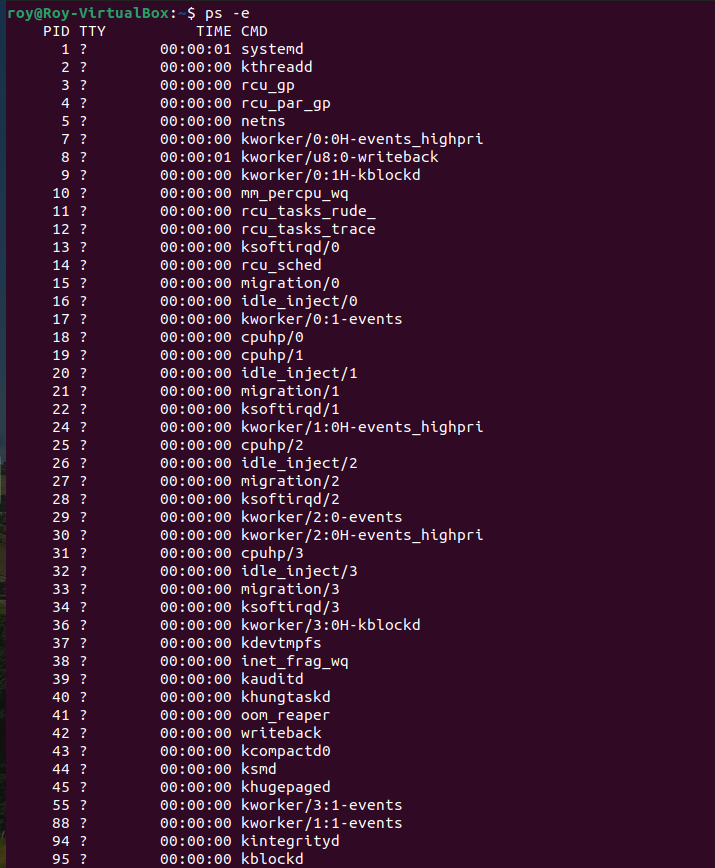
1. **Use the Linux command ps to show the current running process.**



***Figure 4.1.1：****The current running process*

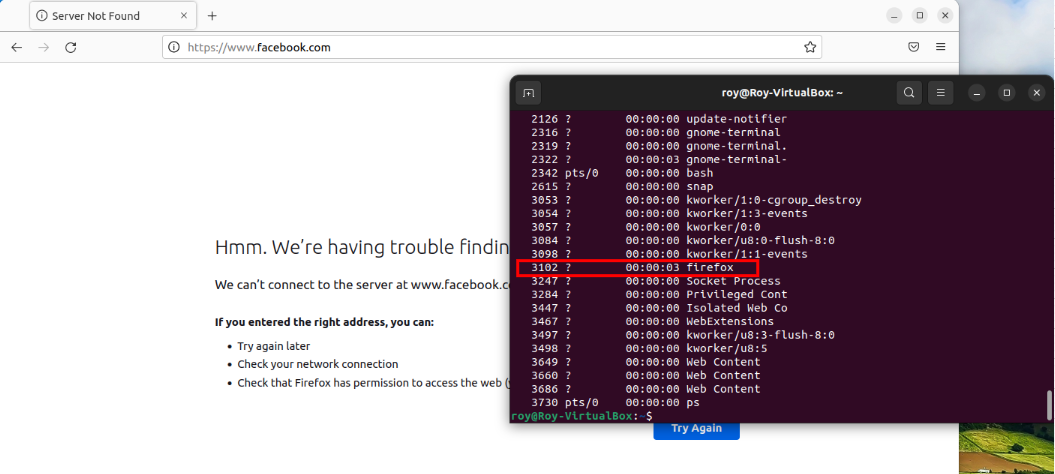
After typing “ps” in the Oracle VM VirtualBox terminal, it will print out the currently running process within the virtual machine. Process Identification numbers (PID) are being assigned automatically and uniquely for each task that is being performed such as browsing the Internet or launching a browser. TTY is the terminal type that the user used to log into. Time is the total amount of time used in minutes and seconds for the process that is currently running. CMD is the name of the command used to run the process.

1. **Give an example on how to terminate a process using kill command.**



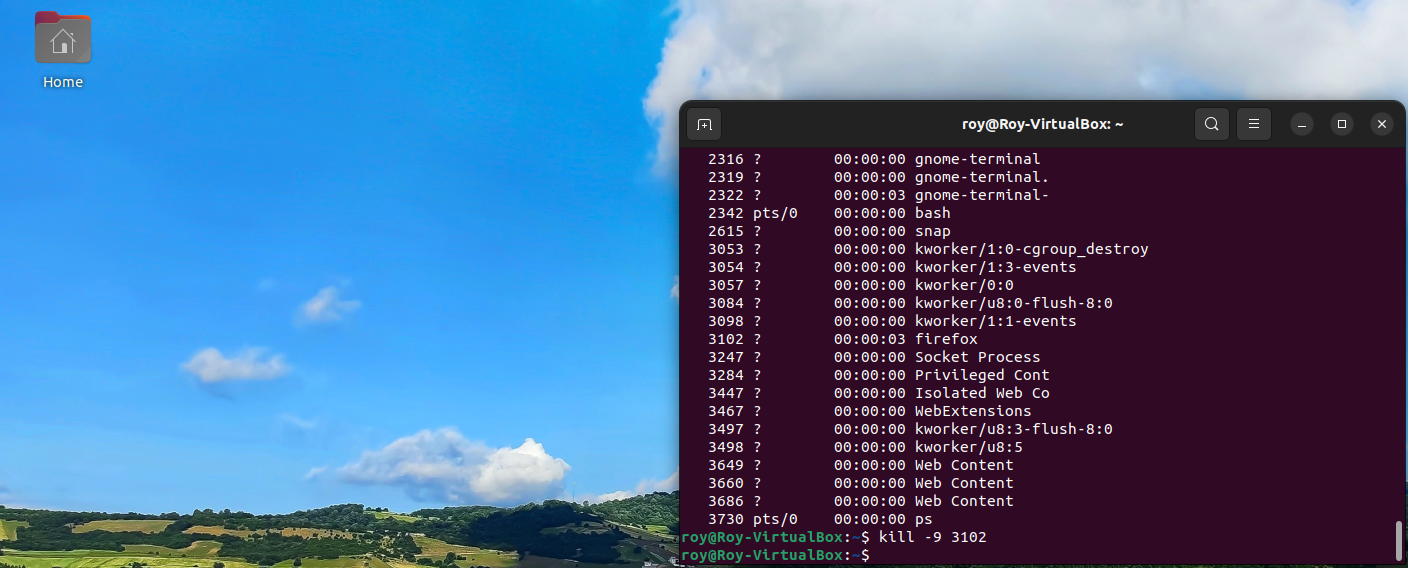
***Figure 4.2.1：*** *Listing of all current running process*

In order to terminate a running process using the kill command, firstly the user must type “ps -e” to print out all the currently running processes.



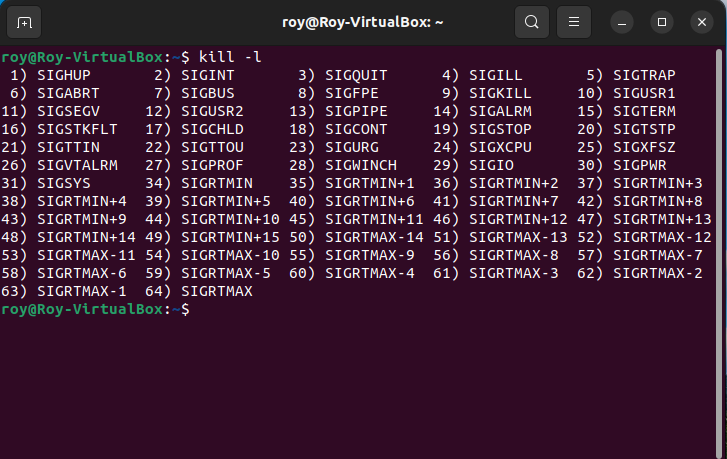
***Figure 4.2.2：****Choosing process to terminate*

For example, Mozilla Firework is opened, and the user wanted to terminate the process using the kill command. After entering “ps –e”, search for the “Firefox” process and refer to its Process Identification Number (PID), which is “3102”.



***Figure 4.2.3：*** *Firefox process terminated*

After identifying the “Firefox” PID, type in “kill -9 3102” and enter. After the command is triggered, it will then kill the process of “Firefox”. As shown in **Figure 4.2.3**, the Firefox program has been terminated using the kill command.



***Figure 4.2.4：*** *Types of kill command*

There are numerous types of kill command that can be used in Ubuntu. Referring to **Figure 4.2.4**, the kill command that was used in **Figure 4.2.3** is number 9, called SIGKILL or signal 9. SIGKILL is type of kill command used by Linux to quickly force terminate a process. SIGKILL is a powerful kill command that is often used by system administrators in terminating a process. This type of kill command cannot be blocked or ignored.

**Part 5: Networking Concepts**

* **0204677 Lim Zhe Yuan**



The network layer is the 3rd layer in the Open Systems Interconnection (OSI) model which handles complex routing and communication between two different networks. When a data transfer occurs over a network, this layer is responsible for handling the segmentation and reassembly of data, called packets, that passed the transport layer – the 4th layer. It also performs logical addressing which uniquely identifies each device operating over a network (Kanade, 2022). In terms of routing, the network layer also determines the optimal route for data transmission (Kanade, 2022) using dynamic routes. The routes that are determined by the layer changes based on multiple factors, such as but not limited to unstable network conditions, priority of service, and logical-to-physical address mappings (Microsoft, 2022). One of the most significant and more familiar protocol that belongs to the network layer is the Internet Protocol, which is the standard for routing packets across interconnected networks (IBM Corporation, 2010) in the present.



Before explaining about classful addressing, one should first understand about IP addresses. IP addresses are unique addresses that are assigned to hosts so that they can be identified across a network. In IPv4, IP addresses are 32-bit addresses that have an address space of 232 (Kumar and Miglani, 2021).

**[x].[x].[x].[x]**

**x can be any value between 0-255**

**[] = each segment equals 8 bit**

**e.g. 192.168.0.1**

Classful addressing is the categorisation of IP addresses based on the number of bits that are occupied for the network and host IDs. It determines the category that a bit in an IP address belongs to and also the total number of possible networks and hosts per network for a particular class. There are 5 main sub-classes for IP addresses: Class A, B, C, D and E. The following table specifies the leading higher order bits which will always be set for the first few bits of an IP, IP range, number of bits for networks and hosts, and total number of networks and hosts per network for each class (Kumar and Miglani, 2021):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Leading bits** | **Number of bits for network ID** | **Number of bits for network ID** | **Total number of networks** | **Total number of hosts per network** | **IP Range** |
| A | 0 | 8 | 24 | 27 | 224 | 0.0.0.0 – 127.255.255.255 |
| B | 10 | 16 | 16 | 214 | 216 | 128.0.0.0 – 191.255.255.255 |
| C | 110 | 24 | 8 | 221 | 28 | 192.0.0.0 – 223.255.255.255 |
| D | 1110 | Not defined | Not defined | Not defined | Not defined | 224.0.0.0 – 239.255.255.255 |
| E | 1111 | Not defined | Not defined | Not defined | Not defined | 240.0.0.0 – 255.255.255.255 |

Class D and E are not used when assigning IP addresses to devices because they are reserved for multi-casting and experimental purposes respectively. According to the rules of classful addressing, IP addresses should not have all of their host bits set to either 1 or 0 as they denote either the network ID or broadcasting address of an IP address. The following special IP addresses will also be unavailable for assignment as they are reserved for their individual purposes (Kumar and Miglani, 2021): -

**169.254.0.0 – 169.254.0.16**: Link local addresses

**127.0.0.0 – 127.0.0.8**: Loop-back addresses

**0.0.0.0 – 0.0.0.8**: used to communicate within the current network.



A subnet, short for subnetwork, is a network obtained from dividing an IP address into smaller segments. By subnetting IP addresses, networks can be turned into smaller, private chunks of subnetworks for categorization. It allows easier routing and management within a large network and prevents too many devices from clogging up a network’s bandwidth (Heddings, 2020). Subnets are made using a subnet mask, which is a bitmask used to identify the subnet of an IP address by applying bitwise AND operations on the IP address using the mask (Solarwinds, 2022). By increasing the bits occupied by the network ID, more subnets can be created at the expense of reducing the total number of hosts per network.



IP addresses are divided into two versions: IPv4 and IPv6. IPv4 and IPv6 are Internet Protocol version 4 and 6 respectively, with IPv6 being the newer version of Internet Protocol. IPv6 was developed to deal with the exhaustion of IPv4 addresses and is better than IPv4 in terms of complexity and efficiency (Samdare, 2022; Agrawal, 2021).

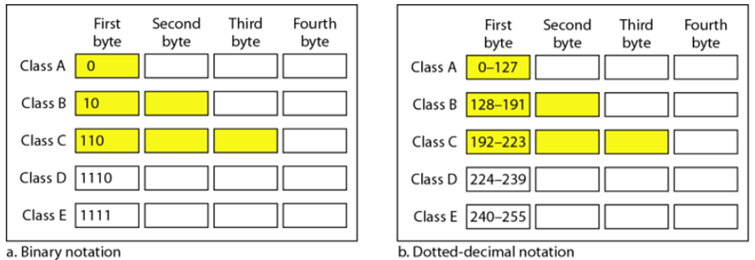
In IPv4, IP addresses are 32 bits in length, represented in decimal, and can generate up to 4.29 x 109 address spaces. IP addresses are divided into multiple classes using classful addressing and provides a broadcast message transmission scheme. Encryption and authentication facilities are not provided and security features for the address are dependent on external applications. It only supports manual and DHCP address configurations. An example of an IPv4 address would be 66.94.29.13 (Samdare, 2022).

Alternatively in IPv6, IP addresses are 128 bits in length, represented in hexadecimal, and can generate up to 3.4×1038 address spaces that is significantly larger than IPv4’s available address spaces. Although IPv6 do not separate their IP addresses into classes, multicast and anycast message transmission schemes are available for use. Encryption and authentication facilities are provided and IPSec is a built-in security feature for IPv6 addresses. It also supports auto and renumbering address configurations. An example of an IPv6 address would be 2001:0000:3238:DFE1:0063:0000:0000:FEFB (Samdare, 2022).

* **0205666 Roy Lim Chen Rui**

**Any ONE layer of the Open Systems Interconnection (OSI) model. (5 marks)**

One of the OSI reference models which sought my attention and brought interest for me to know more about it is Transport Layer (Host Layer) or also called as the heart of OSI model. The role of Transport Layer are Segmentation and Reassembly. However, Transport layer is responsible in breaking a message into a smaller data or known as segments and transporting the segments from end to end making sure that the segments arrive safely at the receiver’s end then only reassembles the data into readable text. Moreover, its duties are to work with application layer and network layer by providing services to application layer and getting services from network layer. Furthermore, Transport Layer provides error-free end to end channel for both connection-oriented and connectionless services. For connection-oriented services, it keeps on repeating the three phases of process which are connection establishments, data transfer, and termination. In this type of transmission, the receiver devices will send an acknowledgement back to the source once the packet is being transferred over to the receiver safely. Hence, this transmission is more reliable and secure than connectionless service. Other than that, for connectionless service, is a single-phase process that includes data transfer. Meaning that, the receiver will not be able to acknowledge the recipient of the packet that is being sent to the user. Connectionless service allows much faster transmission of packet between end-to-end users. In conclusion, connection-oriented service is still more reliable than connectionless service.



***Diagram 1: Classful Addressing Table***

**Classful addressing. (5 marks)**

Classful Addressing is an IPv4 addressing method that divides addresses into five different groups which are Class A, Class B, Class C, Class D, and Class E. Each class has four bytes and is divided into two parts which are network id and host id. Class A has the least network and most host id. But for class B, the network id and host id are equal. Class C has more networks compared to classes A and B but has least host id. Class D is a multicast address and class E is reserved for future purposes. In order to understand the binary notation into dotted-decimal notation is. For example, **Class A** has 1 byte as the network id, and it is 0. Therefore, the range for class A is from 0 to 7 bytes after first byte sum up together which is 64+32+16+8+4+2+1 equal 127 and the range for class A is 0 to 127. An example of cla**ss B**, has 2 bytes as a network and the first byte is 1 which is 128 and the 6 bytes after which are 32+16+8+4+2+1 equal 63, and sum up with 128 will get 191 which is the maximum range for class B. Hence, the range for class B is 128 to 191. For **Class C**, 3 bytes are being used in the front with 2 one meaning that 128 + 64 equals 192 is the starting point of the range and sum up 5 bytes after it 16+8+4+2+1 equals 31. Therefore, 192+31 equals 223 is the maximum range for class C. For Class **D**, 4 bytes used in front with 3 ones sum up 128+64+32 will be getting 224 as starting off the range and sum up 4 bytes behind 8+4+2+1 equal 15. Therefore, 224+15 equals 39, and the range for class D is 224 to 239. Lastly for **Class E**, 4 bytes are being used with all one. Hence, 128+64+32+16 are 240 as starting and sum up 4 bytes behind 8+4+2+1 equal 15. Therefore, 240+15 equals 255 and the range for class D is 240 to 255.

**Subnets. (5 marks)**

A subnet or called a subnetwork is a piece of a large network that is broken into smaller and easily manageable networks using the strategy called subnetting. The role played by subnetting is to convert the IP networks into multiple and smaller network segments. In order to minimize the traffic, control the traffic so it does not have to flow through unnecessary routes and increase the network speeds. Most organizations will use the subnet strategy to subdivide large networks into smaller and more efficient subnetworks. Moreover, each subnet allowed those devices that are connected to the subnet to communicate with each other, while the router is used to allow communication between the subnets. The number of sizes of the subnet depends on the connectivity requirement. Furthermore, the subnetting strategy is essential to be used when a single network wanted to allocate over numerous segments of the local area network. Other than that, the subnets are designed to reallocate IP addresses. Meaning that, it divides a single network into multiple subnets that allows the system administrator to use one block of addresses on multiple physical networks. Not only that, subnets can be used for lowering the network congestion by placing the traffic that is often being shared into the same subnet so that it can lower down the network congestion and also every computer or server on the network will not see data packets from every other computer. Lastly, subnets can also improve the network security. It allows network administrator to quarantine compromised sections, and by doing so, it can help to reduce network-wide threats and also make it difficult for unauthorized user to move around an organization’s network.

**IPv4 vs IPv6. (5 marks)**

IPv4 is the fourth version of the internet protocol and IPv6 is the sixth version of the internet protocol. IPv6 is the newest version of the internet protocol which is better than IPv4 version four in terms of complexity and efficiency. IPv4 is widely used in uniquely identifying devices on a network using an addressing system and establishes rules for computer networks operating on the principle of exchanging packets. Moreover, IPv4 uses 32 bit of address scheme which allows to store 4.19 billions of addresses using the formula 2^32. Because of the increasing number of users that connect to the internet day by day, it has reached the maximum IPv4 address. Hence, the newest version of internet protocol which is version 6 IPv6 is being developed to fulfil the need for more internet addresses. IPv6 is the latest version of internet protocol which is deployed in 1999 with the aim to resolve issues that is encountered by IPv4. Not only that, IPv6 is also developed to resolves the barriers to allows multiple devices to connect to the internet. Furthermore, IPv6 uses 128 bit of address which support 340 undecillion unique addresses using the formula 2^128. For IPv4, checksum field is available. Whereas in IPv6 checksum field is not available. In IPv4, it has broadcast Message Transmission Scheme. Whereas IPv6 multicast and anycast message transmission scheme is available. In IPv4 Encryption and Authentication facility not provided while in IPv6 Encryption and Authentication are provided. Lastly, IPv4 has a header of 20 to 60 bytes while IPv6 has header of 40 bytes fixed.

* **0205430 Tan Peng Heng**

1. **Explain One OSI Model (Transport Model)**

Open Systems Interconnection (OSI) model contains 7 layer which are, physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and application layer. Below was the explanation of the transport layer in the OSI model.

According to Imperva (2022), the transport layer will transfer the data in the session layer and breaks the data into few segments and transfer to network layer and proceed the OSI model until it used the physical layer to transfer the data to the receiver and when the receiver received the data, it will pass through the OSI model again until the transport layer then it will joining back the broken segments together and pass back to the session layer to continue the progress. The responsibility of the transport layer is to send data, error control, checking data was received incorrectly or not if not then it will request again for sending data.

1. **Classful addressing**

According to Petryschuk (2021), classful addressing was an IPV4 addressing architecture that divides the addresses space into five classes. Firstly, Class A and Class B contains more host ID compared to network ID so normally it has been used by large organizations. Besides that, Class C have a smaller address space for hosts, so it normally been used by home or office network. Lastly, Class D are reserved for multi-casting and Class E are reserved for experimental and research purposes. Due to IPV4 address limitation, the first eight bits in the IP address defined the host of a network maximum up to 254 networks and IPV4 addresses was exhausted because the limitation of IPV4 address space and hungry of IP address. Diagram 5.1 shown about the differences of class A, B and C.

Table

Description automatically generated

***Diagram 5.1:*** *The differences of Class A, B, C*

1. **Subnets**

According to Ferguson (2021), subnetwork or known as subnet is a part of subnetwork of IP address. Subnets divided large network to smaller network from the host ID part based on the Class of the network and the size of the subnets will cover is depends on the connectivity requirements, network engineer, administrator, and others. Subnet may be segmented into smaller subnets and giving the flexibility to create smaller subnets for things like point-to-point links to support for more devices.

Subnet can be used for reallocating IP addresses, relieving network congestion, and improving network security. Subnet with reallocating IP addresses will be using the subnets that divided from the Host ID based on the Classes of the IP addresses and manage to let many physical networks available in same addresses block. Furthermore, subnet that used to improving network security is also a good example for explaining subnet since the subnet divided the big network into smaller and smaller which can reduce the network-wide threats and making more difficult to get allocate by the hackers.

1. **IPV4 vs IPV6**

According to Pramatarov (2018), the different of Internet Protocol Version 4 (IPV4) and Internet Protocol Version 6 (IPV6) was mainly the improvement of internet protocol from the version of 4 to 6. Internet Protocol (IP) can be defined into two types which are Private and Public IP addresses, it provides the way of connection between devices, and it has its own algorithms of data travels from hosts to its destination.

In terms of IPV4 address is a 32bit addressing method and it will be having four blocks of segments and each segment can be form by 0 to 254 of numbers and it divided the blocks and numbers with dots. For example, 192.168.0.1. Based on math calculations Maximum of IPV4 can be up to 4,294,967,296 networks available for users so with this limitation of networks IPV4 is exhausted due the hunger of IPV4 running out the available networks so IPV6 is invented to be overcome this problem.

IPV6 addresses is a latest version of IP. IPV6 used 128bit addressing method and it has eight blocks of segments in total and it divided all by colon mark which similar different from IPV4. For example, “2001:0db8:0000:0042:0000:8a2e:0307:7334”. IPV6 used four hexadecimal (hex) digits to form in each of the block which makes IPV6 have more combinations and networks available for the users. Diagram 5.2 shown the differences between IPV4 and IPV6.

Graphical user interface, text, application, email

Description automatically generated

***Diagram 5.2:*** *The differences between IPV4 and IPV6*

Furthermore, IPV6 has a faster speed compared to IPV4 due to IPV6 has enough global addresses so it does not need Network Address Translation (NAT) to transits data but IPV4 need to deal with NAT. Besides that, IPV6 has a lower security since IPV6 included IPSec without taking any extra steps compared to IPV4, it makes the address scanning for the IPV6 more difficult, and the support of end-to-end encryption can all prevents the man-in-the-middle attacks or other attacks.

Lastly, IPV6 is a 128bit address network which better than IPV4 32bit address network and it has a better routing with fragmentation of packets. Besides that, IPSec will secure and prevents the attacks from the hackers. Out of that, IPV6 will be provides enough IPs for a long time although there was 30% of IPV6 network is getting used by the world’s internet users but it was still more empty network ready to be use.

* **0205096 Thor Wen Zheng**

1. **Any One OSI Layer (Transport Layer)**

The Open Systems Interconnection (OSI) model, created by the International Organization for Standardization (ISO), is a standard reference model that defines all aspects of data communication in computer networks. The OSI model is divided into 7 layers, each consisting of different protocols performing various functions that depend on one another to complete the data transfer process between 2 or more networked devices (Microsoft, 2017). One of the 7 OSI layers is the **Transport Layer**, which is arguably the most important layer.

The Transport Layer is the 4th layer in the OSI model, in between the Network Layer (3rd layer) and the Session Layer (5th layer). The data transferred in this layer is measured in units of segments. The primary function of the Transport Layer is to ensure that messages transferred between hosts are error-free, in correct sequence, and have no duplications or loss. It ensures error-free message transmission from Layers 1 through 3 by segmenting messages, correctly reassembling messages at the receiving end of the transmission, performing flow control and traffic control, and more. The Transport Layer also consists of communication protocols including the connection-oriented Transmission Control Protocol (TCP) and connectionless User Datagram Protocol (TCP) (Microsoft, 2017; Techopedia, 2021).

Message segmentation is necessary because the size of data packets may usually be larger than the maximum acceptable packet size supported in the network. Furthermore, message segmentation ensures that when a certain message fails to send, not the entire message fails, because it is likely that the transmission failure occurred after some message segments have already been transmitted and received. The remaining segments that failed can then be resent, instead of needing to resend the entire message. Subsequently, message reassembly must be done correctly such that no data is lost in the process of transmission and the message is in correct sequence (Bouftira, 2016; Microsoft, 2017). Additionally, flow control ensures that the rate of data transfer between the sender and receiver is acceptable for both sides to prevent buffer overruns or buffer underruns. Traffic control is done to detect network data congestion caused by issues such as overloaded network node or restricted bandwidth and take appropriate action to solve these congestion issues (Techopedia, 2021).

1. **Classful Addressing**

An Internet Protocol (IP) address is a unique numerical label used as an identifier for devices connected to a network. In the early stages of the Internet, only the first octet in an IPv4 address was used to identify a network, resulting in only 254 possible networks, each consisting of 16,777,216 IP addresses. As the world obviously had more than 254 organizations that need a computer network, and very few networks that actually need 16.7 million addresses, **classful addressing** was introduced (Petryschuk, 2021). Classful addressing is an IPv4 classification system that breaks down IP addresses into 5 classes, including Classes A, B, C, D, and E (Microsoft, 2017). The following table shows the IP range, network/host portions, total number of networks, and total number of usable addresses for each class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **IP Range (1st octet)** | **Network/Host Portions** | **Total Number of Networks** | **Total Number of Usable Addresses per Network** |
| A | 0 – 127 | Net.Host.Host.Host | 27 = 128 | 224 – 2 = 16,777,214 |
| B | 128 – 191 | Net.Net.Host.Host | 214 = 16,384 | 216 – 2 = 65,534 |
| C | 192 – 223 | Net.Net.Net.Host | 221 = 2,097,151 | 28 – 2 = 254 |
| D | 224 – 239 | N/A | N/A | N/A |
| E | 240 - 255 | N/A | N/A | N/A |

***Table 5.1:*** *IPv4 classful addressing*

Class A addresses are only given to the largest organizations such as government agencies, Internet Service Providers (ISP), universities, and big corporations because they require the most addresses for their tremendous number of networked devices. Class B addresses are given to large-scale organizations that do not require as many addresses as the largest organizations using Class A. Class C, which is the most common, is given to anyone who requires a small network that does not require the address amount provided in Class A or B. Class D and E are not used by regular networks as Class D is used for multicasting, which is the transmission of data to multiple hosts, while Class E was reserved for future use (Microsoft, 2017).

There are also restrictions on certain specific IP addresses. For example, the address 127.0.0.1 is the localhost address reserved for loopback testing. The first address, x.x.x.0, and the last address, x.x.x.255, in every network cannot be used as the first address is used to identify an entire network while the last address is used as the broadcast address. Therefore, the number of usable addresses in a network is always 2 less than the total calculatable amount. Additionally, IPv4 addresses can be further categorized into public and private IP addresses. Public IP addresses are exposed on the Internet and can be communicated with. Private IP addresses are those which are hidden from the Internet or other networks, behind a firewall or IP proxy (Microsoft, 2017). The table below shows the ranges of private addresses in Classes A, B, and C.

|  |  |  |
| --- | --- | --- |
| **Class** | **Start of IP Range** | **End of IP Range** |
| A | 10.0.0.0 | 10.255.255.255 |
| B | 172.16.0.0 | 172.31.255.255 |
| C | 192.168.0.0 | 192.168.255.255 |

***Table 5.2:*** *Private IPv4 addresses for Classes A, B, and C*

1. **Subnets**

In the early days of IPv4, addresses in each class specifically only had a portion of bits fixed for the network, and another portion for the host address. For example, in a Class A address “69.0.0.1”, 69 represents the network, while 0.0.1 represents the host. This means the first octet is reserved for the network, while the remaining 3 octets are reserved for the host. As a result, this network has approximately 16.7 million usable addresses. This becomes a problem when the organization who owns the network does not use all 16.7 million addresses, as millions of addresses could be wasted. Furthermore, when a host wants to communicate with another host within this Class A network, it may need to go through millions of connected devices to locate the correct destination device, causing network congestion and reduced network speed (CBT Nuggets, 2017). To solve these issues, the concept of **subnets** was applied.

A subnet is a logical subdivision of a network, or simply a sub-network. By dividing a large network into smaller subnets, network performance can be drastically improved. For example, when a host wants to communicate with another host in the same subnet, the data is transmitted only within the subnet, thus reducing congestion outside the subnet. When the host desires to communicate with another host in a different subnet, the data transmitted only needs to go through the routers or gateways of other subnets, which is significantly lesser than the number of hosts in a network with no subnets. It is also much easier to maintain network security as there are lesser access points. Additionally, network administrators can have an easier time managing smaller subnets and expanding the overall network without wasting any addresses (CBT Nuggets, 2017).

Every network has a default subnet mask based on its IP address class, as shown in **Table x**. In order to divide a network, the subnet mask of the network must be modified. This is done by “borrowing” bits from the host bits and converting them into network bits, also known as masked bits. The additional “borrowed” bits can then be used to define subnets of the network. The more masked bits, the more subnets in a network, but with lesser hosts per subnet (Microsoft, 2017). For example:

* 69.0.0.0 is a Class A network
* The default subnet mask is 255.0.0.0, with bits of 11111111 00000000 00000000 00000000
* 5 bits is borrowed from the host bits, resulting in subnet mask of 255.248.0.0, with bits of 11111111 11111000 00000000 00000000
* 5 bits were borrowed, 2^5 is equal to 32, the first and last subnet cannot be used, so this results in 30 possible subnets
* There are 19 host bits remaining, 2^19 is equal to 524,288, the first and last addresses cannot be used, so this results in 524,286 hosts per subnet

|  |  |  |
| --- | --- | --- |
| **Class** | **Decimal** | **Binary** |
| A | 255.0.0.0 | 11111111 00000000 00000000 00000000 |
| B | 255.255.0.0 | 11111111 11111111 00000000 00000000 |
| C | 255.255.255.0 | 11111111 11111111 11111111 00000000 |

***Table 5.3:*** *Default subnet masks of Classes A, B, and C*

1. **IPv4 vs IPv6**

An IP address is a numerical label used to uniquely identify devices in a network (Microsoft, 2017). To provide some background on IP address versions, there are primarily 2 versions of IP addresses, namely Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) addresses. There were other IP versions including IPv1, IPv2, IPv3, and IPv5, but they were either named differently and used for purposes other than addressing or were used as drafts for subsequent versions used today (Weissman, 2022). Although bound to be replaced by IPv6, IPv4 is still the most commonly used communications protocol.

An **IPv4** address consists of 32 bits, divided into 4 octets. An IPv4 address is usually expressed in dotted decimal format with 4 portions, each ranging from 0 to 255. For example, 192.168.69.0 is a Class C IPv4 address. As IPv4 only has 32 bits, it can have only 4,294,967,296 possible unique addresses. Because of the rapid growth of the Internet and address limitations of IPv4, the available IPv4 addresses had quickly become depleted, leading to IPv4 address exhaustion. Several methods such as subnetting, Network Address Translation (NAT), and Classless Interdomain Routing (CIDR) were used to slow down the exhaustion of IPv4 addresses, which were still not sufficient (Microsoft, 2017).

**IPv6** was introduced to address the inherent issues of IPv4. An IPv6 address consists of 128 bits, represented using hexadecimal numbers. It has a total of 32 hexadecimal numbers, divided into 8 groups, each consisting of 4 numbers. As IPv6 uses 128 bits, there can be a total of 340 undecillion (340 with 36 trailing zeros) addresses, which may never run out in the foreseeable future. An example of IPv6 address is 2001:4860:0000:2001:0000:0000:0000:0068, which can be abbreviated by removing leading 0s and representing multiple groups of consecutive 0s as a double colon, resulting in the address 2001:4860:0:2001::68. There are 3 types of IPv6 addresses, including unicast, anycast, and multicast. Unicast addresses are those that define a single address of a single host; global unicast addresses are routable public addresses exposed to the Internet; link-local unicast addresses are autoconfigured internal addresses in a network. Anycast addresses are assigned to single groups of hosts, with packets sent to only the destination host. Multicast addresses are also assigned to single groups of hosts, but data packets are sent to all hosts in the group. Furthermore, IPv6 incorporates Internet Protocol Security (IPSec), an advanced network security protocol suite used for packet authentication and encryption (Microsoft, 2017). IPv6 not only solves the issue of IP address exhaustion, but these numerous characteristics and advanced features of IPv6 make it far superior to IPv4 in terms of convenience and security. However, modern devices are typically built with dual IP stack, meaning that they support both IPv4 and IPv6 (Microsoft, 2017).

**References**

Agrawal, A. (2021) *Internet Protocol version 6 (IPv6)*. Available from <https://www.geeksforgeeks.org/internet-protocol-version-6-ipv6/?ref=lbp> [accessed 10 November 2022].

Bouftira (2016) *07: Transport layer – Segmentation, packaging and reassembly part 1*. Computer Networks For All. Available from <http://www.cnt4all.com/2016/08/07-transport-layer-segmentation.html> [accessed 24 November 2022].

CBT Nuggets (2017) *5 Subnetting Benefits*. NetworkComputing. Available from <https://www.networkcomputing.com/data-centers/5-subnetting-benefits> [accessed 25 November 2022].

Ferguson, K. (2021) *What is a subnet (subnetwork)? Definition from SearchNetworking.* TechTarget. Available from <https://www.techtarget.com/searchnetworking/definition/subnet>. [accessed 25 November 2022].

Heddings, A. (2020) *What are Subnets, and How Do They Affect My Network?* Available from <https://www.howtogeek.com/devops/what-are-subnets-and-how-do-they-affect-my-network/> [accessed 10 November 2022].

IBM Corporation (2010) *Networking on z/OS*. Available from <https://www.ibm.com/docs/en/zos-basic-skills?topic=review-network-layer-layer> [accessed 10 November 2022].

Imperva (2022) *What is OSI Model | 7 Layers Explained | Imperva*. Available from <https://www.imperva.com/learn/application-security/osi-model/#:~:text=What%20Is%20the%20OSI%20Model> [accessed 8 November 2022].

Infoblox (n.d). *What is Layer 4 of the OSI Model: Transport Layer? | DDI (Secure DNS, DHCP, IPAM).* Available from <https://www.infoblox.com/glossary/layer-4-of-the-osi-model-transport-layer/#:~:text=Layer%204%20of%20the%20OSI%20Model%3A%20Transport%20Layer%20provides%20transparent> [accessed 25 November 2022].

Janardhan, J. (2020) *The Transport Layer of OSI Model.* Available from <https://www.tutorialspoint.com/The-Transport-Layer-of-OSI-Model>. [accessed 25 November 2022].

Kanade, V. (2022) *What Is The OSI Model? Definition, Layers, and Importance*. Available from <https://www.spiceworks.com/tech/networking/articles/what-is-osi-model/> [accessed 10 November 2022].

Kumar, M. and Miglani, G. (2021) *Introduction of Classful IP Addressing.* Available from <https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/> [accessed 10 November 2022].

Margaret (2021) *IPv4 vs IPv6: Which is Faster?* FS Community. Available from <https://community.fs.com/blog/ipv4-vs-ipv6-whats-the-difference.html> [accessed 25 November 2022].

Microsoft (2017) *Exam 98-366 Networking Fundamentals*, 2nd edition. New York, USA: John Wiley & Sons Inc.

Microsoft (2022) *Windows Network Architecture and the OSI Model*. Available from <https://learn.microsoft.com/en-us/windows-hardware/drivers/network/windows-network-architecture-and-the-osi-model> [accessed 10 November 2022].

Petryschuk, S. (2021) *Classful and Classless Addressing Explained*. Auvik. Available from <https://www.auvik.com/franklyit/blog/classful-classless-addressing/> [accessed 24 November 2022].

Pramatarov, M. (2018) *IPv4 vs IPv6 and where did IPv5 go?* ClouDNS. Available from <https://www.cloudns.net/blog/ipv4-vs-ipv6-internet-protocol/> [accessed 15 November 2022].

Prep, F. (n.d.). *IP Addressing | Introduction and Classful Addressing.* FACE Prep. Available from <https://www.faceprep.in/computer-networks/ip-addressing--introduction-and-classful-addressing/> [accessed 25 November 2022].

Samdare, B. (2022) *Differences between IPv4 and IPv6*. Available from <https://www.geeksforgeeks.org/differences-between-ipv4-and-ipv6/> [accessed 10 November 2022].

Solarwinds (2022) *Subnetting: What Is a Subnet? How It Works.* Available from <https://www.solarwinds.com/resources/it-glossary/subnetting> [accessed 10 November 2022].

Techopedia (2021) *Transport Layer.* Janalta Interactive. Available from <https://www.techopedia.com/definition/9760/transport-layer> [accessed 24 November 2022].

Thiyari, K. and Pandey, H. (2019) *Layers of OSI Model.* GeeksForGeeks. Available from <https://www.geeksforgeeks.org/layers-of-osi-model/> [accessed 25 November 2022].

Tutorialspoint (n.d.). *Classful Vs Classless Addressing.* Available from <https://www.tutorialspoint.com/classful-vs-classless-addressing> [accessed 25 November 2022].

Weissman, J.S. (2022) *IPv4, IPv6 … What Happened to All the Other Numbers?*. American Registry for Internet Numbers. Available from <https://www.arin.net/blog/2022/05/12/other-ip-version-numbers/> [accessed 29 November 2022].

Williams, L. (2019) *IPv4 vs IPv6: What’s the Difference?* Guru99.com. Available from <https://www.guru99.com/difference-ipv4-vs-ipv6.html> [accessed 25 November 2022].

WinZip (2022) *What is a TAR GZ file extension?* Available from <https://www.winzip.com/en/learn/file-formats/tar-gz/> [accessed 10 November 2022].

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MARKING RUBRIC**  **DSM2294 SYSTEMS ADMINISTRATION AND MANAGEMENT**  **Assignment (100 Marks, Weighted @ 30%)** | | | | | | | | |
| 1. ***Student ID: 0204677***   ***Student Name: Lim Zhe Yuan*** 2. ***Student ID: 0205666***   ***Student Name: Roy Lim Chen Rui*** | | | | 1. ***Student ID: 0205430 Student Name: Tan Peng Heng*** 2. ***Student ID: 0205096 Student Name: Thor Wen Zheng*** | | | | |
| **LEARNING OUTCOME** | **MARKING CRITERIA** | **SCALE** | | | | | | |
| **Fail**  **(0-49)** | **3rd Class**  **(50-59)** | | **2nd Lower Class**  **(60-69)** | **2nd Upper Class**  **(70-79)** | **1st Class**  **(80-100)** | **Task Marks**  **(Max. 100)** |
| ***Group Work (80 Marks)*** | | | | | | | | |
| CLO1: Install and manage system resources effectively | Part 01: Linux Setup  (10 Marks) | *Linux was poorly setup with little-to-no user documentation with screen shots created.* | *Linux was adequately setup with basic user documentation with screen shots created.* | | *Linux was fairly-well setup with some user documentation with screen shots created.* | *Linux was well setup with most user documentation with screen shots created.* | *Linux was very well setup with all user documentation with screen shots created.* |  |
| CLO2: Administer user accounts and groups to limit access | Part 02: File Permissions (40 Marks) | *File permissions was poorly assigned.* | *File permissions was adequately assigned.* | | *File permissions was fairly well assigned.* | *File permissions was well assigned.* | *File permissions was clearly and properly assigned.* |  |
| Part 03: Backup Directory Creation and Management  (20 Marks) | *Backup directory was poorly created with little-to-no backup and recovery functions implemented and/or automatic scheduler successfully implemented.* | *Backup directory was adequately created with basic backup and recovery functions implemented and/or automatic scheduler successfully implemented.* | | *Backup directory was fairly well created with some backup and recovery functions implemented and/or automatic scheduler successfully implemented.* | *Backup directory was well created with most backup and recovery functions implemented and/or automatic scheduler successfully implemented.* | *Backup directory was very well created with all backup and recovery functions implemented and/or automatic scheduler successfully implemented.* |  |
| Part 04: Process management  (10 Marks) | *Terminate process was poorly handled.* | *Terminate process was adequately handled.* | | *Terminate process was fairly well handled.* | *Terminate process was well handled.* | *Terminate process was very well handled.* |  |
| ***Individual Work (20 Marks)*** | | | | | | | | |
| CLO3: Configure network services for intranet and internet | Part 05: Networking Concepts  (20 Marks) | *Networking concepts was poorly described and explained, with little-to-no range of references included.* | *Networking concepts was adequately described and explained, with basic range of references included.* | | *Networking concepts was fairly well described and explained, with fair range of references included.* | *Networking concepts was well described and explained, with good range of references included.* | *Networking concepts was very well described and explained, with excellent range of references included.* |  |
| **Total (100%)** | | | | | | | |  |
| **Weighted Marks @ 30%** | | | | | | | |  |