Experiment No.1

Introduction of operating System:

An operating system (OS) is essential for managing and facilitating interactions between a computer's hardware and software. It provides a structured environment for executing programs and handling tasks like file management. In the context of a file experiment, the OS plays a crucial role by allowing users to create, modify, and delete files, and by organizing these files into directories or folders. It also manages file permissions, which control access rights and ensure that only authorized users can perform certain actions on files. Additionally, the OS employs a file system, such as NTFS, FAT32, or ext4, to systematically organize and store files on storage devices. During your experiment, you will observe how the OS efficiently handles these file operations and maintains an organized structure for data management.

Function of Operating System:

Certainly! Here are the key functions of an operating system relevant to a file experiment:

1. File Creation and Deletion: Allows you to create new files and remove existing ones.

2. File Organization: Structures files in directories or folders for easy access and management.

3. File Permissions: Controls access rights, specifying who can read, write, or execute files.

4. File System Management: Uses a file system (like NTFS, FAT32, or ext4) to organize and store files on storage devices.

5. File Access and Retrieval: Manages how files are accessed and retrieved from storage efficiently.

Services of Operating System:

Here are the key services provided by an operating system:

- **1. Process Management: Handles the creation, scheduling, and termination of processes.**

2. Memory Management: Allocates and manages the computer's memory resources.

3. File Management: Oversees file creation, deletion, and organization.

4. Device Management: Manages input and output devices, including drivers and interfaces.

5. User Interface: Provides a user interface (command-line or graphical) for interaction with the system.

6. Security and Access Control: Enforces user authentication and controls access to system resources.

Need for linux operating system:

Linux operating system is widely used for several compelling reasons:

1. Open Source: Linux is open source, meaning its source code is freely available for anyone to view, modify, and distribute. This fosters innovation and customization.

2. Stability and Reliability: Known for its stability and reliability, Linux is often used in environments where uptime is critical, such as servers and embedded systems.

3. Security: Linux is designed with robust security features and has a strong community of developers who quickly address vulnerabilities and provide patches.

4. Cost-Effective: Linux is free to use, reducing costs associated with licensing fees compared to proprietary operating systems.

5. Flexibility and Customization: Linux can be tailored to suit specific needs, from lightweight distributions for older hardware to powerful configurations for advanced users and servers.

6. Performance: Linux typically has a smaller footprint and can be optimized for performance, making it suitable for a wide range of hardware.

7. Community Support: A vibrant community of users and developers provides extensive support and resources, including forums, documentation, and user guides.

8. Compatibility: Linux supports a wide range of hardware architectures and offers compatibility with various software applications, including many open-source tools and enterprise solutions.

History Of linux Operating System:

Linux originated in 1991 when Linus Torvalds, a Finnish student, released the first version of the Linux kernel as a free and open-source alternative to proprietary operating systems. Initially developed as a personal project, Linux quickly garnered support from a global community of developers. By adhering to the principles of open source, it allowed continuous improvements and adaptations. Over the years, Linux evolved into a robust and versatile operating system used across various domains, from personal computers to servers and embedded systems. Its development has been driven by contributions from both individual programmers and large organizations, leading to a diverse ecosystem of distributions tailored for different needs.

Different Services and Application Of Linux Operating system :

Here are some key services and applications of the Linux operating system:

1. Web Servers: Powers popular web servers like Apache and Nginx.

2. Database Management: Supports database systems such as MySQL, PostgreSQL, and MongoDB.

3. File Servers: Provides file sharing services with tools like Samba and NFS.

4. Network Management: Manages network services with tools like iptables and NetworkManager.

5. Development Environment: Used by developers for programming with support for various programming languages and development tools.

6. System Administration: Offers robust tools for system monitoring and management, such as top, htop, and system.

7. Security: Provides strong security features and tools, including SELinux and AppArmor.

8. Virtualization: Supports virtualization platforms like KVM, Docker, and VirtualBox.

9. Desktop Environments: Offers various desktop environments like GNOME, KDE, and XFCE for user interfaces.

10. Embedded Systems: Powers embedded systems and IoT devices due to its flexibility and low resource requirements.

Experiment No. 2

Aim : To get the function of command of linux command .

1.Date :- date command is used to display the system date and time. date command is also used to set date and time of the system.

Syntax- date

```
tryhackme@linux1:~$ date
Wed Sep 25 15:20:41 UTC 2024
tryhackme@linux1:~$
```

2.Whoami :- The whoami command in Linux displays the username of the currently logged-in user.

Syntax-whoami

```
tryhackme@linux1:~$ whoami
tryhackme
tryhackme@linux1:~$
```

3.Whatis:- display one-line manual page descriptions

Syntax-whatis

```
tryhackme@linux1:~$ whatis whoami
whoami (1)           - print effective userid
```

4.Man :- an interface to the system reference manuals

Syntax-Man argument

```
tryhackme@linux1:~$ man whatis
WHATIS(1)                  Manual pager utils                  WHATIS(1)

NAME
       whatis - display one-line manual page descriptions

SYNOPSIS
       whatis [-dlv?V] [-r|-w] [-s list] [-m system[,...]] [-M path] [-L lo-
       cale] [-C file] name ...

DESCRIPTION
       Each manual page has a short description available within it.  whatis
       searches  the manual page names and displays the manual page descrip-
       tions of any name matched.

       name may contain wildcards (-w) or be a regular expression (-r).  Us-
       ing  these  options,  it may be necessary to quote the name or escape
       (\) the special characters to stop the shell from interpreting them.

       index databases are used during the search, and are  updated  by  the
       mandb  program.  Depending on your installation, this may be run by a
       periodic cron job, or may need to be run manually  after  new  manual
       pages have been installed.  To produce an old style text whatis data-
       base from the relative index database, issue the command:

       whatis -M manpath -w '*' | sort > manpath/whatis

       where manpath is a manual page hierarchy such as /usr/man.

OPTIONS
       -d, --debug
              Print debugging information.

       -v, --verbose
              Print verbose warning messages.
```

5.Pwd :- print name of current/working directory

Syntax- pwd

```
tryhackme@linux1:~$ pwd
/home/tryhackme
```

6.**mkdir:-** print name of current/working directory

Syntax-mkdir foldername

```
tryhackme@linux1:~$ mkdir prince
tryhackme@linux1:~$ ls
access.log  folder1  folder2  folder3  folder4  prince
tryhackme@linux1:~$
```

7.**rmdir:-** remove empty directories

Syntax-rmdir foldername

```
tryhackme@linux1:~$ rmdir folder2
tryhackme@linux1:~$ ls
access.log  folder1  folder3  folder4  prince
tryhackme@linux1:~$
```

8.**ls:-** list directory contents

Synatx- ls

```
tryhackme@linux1:~$ ls
access.log  folder1  folder3  folder4  prince
```

9.**Ls-l :-** list directory contents with detail

Syntax- ls-l

```
tryhackme@linux1:~$ ls -l
total 80
-rw-rw-r-- 1 tryhackme tryhackme 65522 May 10  2021 access.log
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder1
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder3
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder4
drwxrwxr-x 2 tryhackme tryhackme  4096 Sep 26 06:26 prince
```

10.**Cat :-** Concatenate files and print on the standard output

Syntax- cat>>foldername

```
tryhackme@linux1:~$ cat >> Prince
Hello i am Prince
```

13.Su:- run a command with substitute user and group ID

Syntax-su

```
tryhackme@linux1:~$ su
Password:
200su: Authentication failure
tryhackme@linux1:~$ 2004
2004: command not found
```

14.Sudo:- execute a command as another user.

Syntax-sudo

```
tryhackme@linux1:~$ sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
            [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
            prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
            prompt] [-T timeout] [-u user] file ...
```

15.ps:- report a snapshot of the current processes**.**

Syntax-ps

```
tryhackme@linux1:~$ ps
    PID TTY          TIME CMD
   1028 pts/0    00:00:00 bash
   1089 pts/0    00:00:00 ps
```

16.Df:- report file system disk space usage

Syntax-df

```
tryhackme@linux1:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root       10089288 2799128   7273776  28% /
devtmpfs          469512       0    469512   0% /dev
tmpfs             477836       0    477836   0% /dev/shm
tmpfs              95568     860     94708   1% /run
tmpfs               5120       0      5120   0% /run/lock
tmpfs             477836       0    477836   0% /sys/fs/cgroup
/dev/loop0         25856   25856         0 100% /snap/amazon-ssm-agent/7993
/dev/loop2         56704   56704         0 100% /snap/core18/1885
/dev/loop1         28800   28800         0 100% /snap/amazon-ssm-agent/2012
/dev/loop3         57088   57088         0 100% /snap/core18/2829
/dev/loop4         94080   94080         0 100% /snap/lxd/24061
/dev/loop5        106752  106752         0 100% /snap/core/17200
/dev/loop7         72320   72320         0 100% /snap/lxd/16922
/dev/loop6         65536   65536         0 100% /snap/core20/2318
tmpfs              95564       0     95564   0% /run/user/1001
```

17.Touch :- Create File

Syntax-touch filename

```
tryhackme@linux1:~$ touch Princefile2
tryhackme@linux1:~$ ls
Prince  Princefile2  access.log  folder1  folder3  folder4  prince
```

18.Cal-show the calender of the given name.

Syntax-cal

```
tryhackme@linux1:~$ cal september 2024
   September 2024
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

19.**VI-Vi IMproved, a programmer's text editor**

Syntax-VIM


VIM - Vi IMproved
version 8.1.1847
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable
Become a registered Vim user!
type :help register<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info

20.Nano-


[Welcome to nano. For basic help, type Ctrl+G.]
Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell