Linux

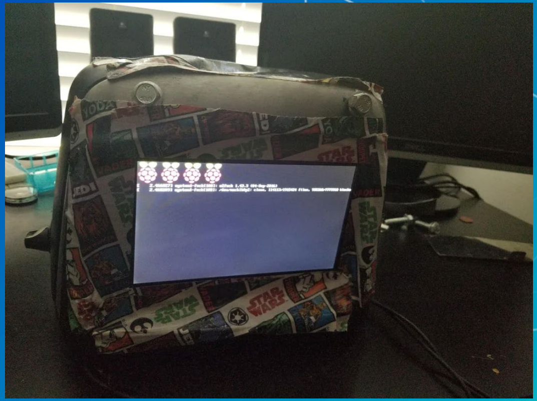
UBNetDef, Fall 2023
Dikshit Khandelwal (DK)
Steffi Yeh

Agenda

- Linux Basics
  - What is Linux? What is Kernal? What is Linux Distribution
  - Terminal
  - Commands – What Am I ? & Get Help!
- File System
  - Navigate File System
  - Interact with Files
  - Text Editors
- In class Activity – Linux CTF
- Users & Groups
- File Permission
- Others

What is Linux?

- You may have heard of Linux being talked about by other students in the context of “kernel space memory management”.
- It’s not that complicated.



What is a Linux?

- Specifically: Linux is an operating system.
  - The bit of software that communicates between the hardware and the applications.
- It’s found everywhere.
  - Operating systems
  - Embedded devices
  - Supercomputers
  - My dog runs Linux.
- More generally: Linux is a group of operating systems (called “distributions”) that all use the Linux kernel.

What is Kernel?

- The kernel is the core component of an operating system that manages and controls all aspects of the system's operations
- Roles of Kernel: Input/Output (I/O) Management, Memory Management, Processor Management

Distributions

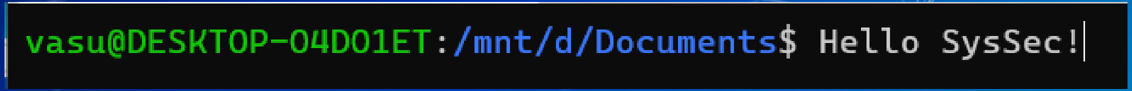
- There are countless different distributions (shortened to “distros”)
- 2 major families:
  - Debian based
    - Includes Debian, Ubuntu, Kali, Mint, Pop
  - Red Hat based
    - Includes Red Hat, Fedora, CentOS, Rocky
- Other distributions include:
  - RedstarOS (리눅스가 최고다)
  - Arch
  - OpenSuse
  - Gentoo
  - Feel free to ask SecDev what they use!

The Terminal

- Another way to interact with your system.
- Most GUI activity can be done here faster.
- When have we used a terminal in class?

The Terminal

- Running without a GUI (headless) mean systems can be more lightweight
- There are several common command line interpreters, or shells
  - bash, zsh, sh, csh, fish, (and many more)
- Typically, you will see a prompt in your shell that gives you some information about your current session, often including your current directory
  - You can customize your prompt via a configuration file (such as `~/.bashrc`)



```
vasu@DESKTOP-O4DO1ET:/mnt/d/Documents$ Hello SysSec!
```

User Hostname Current Directory Type Here

“Command Line” “CLI”

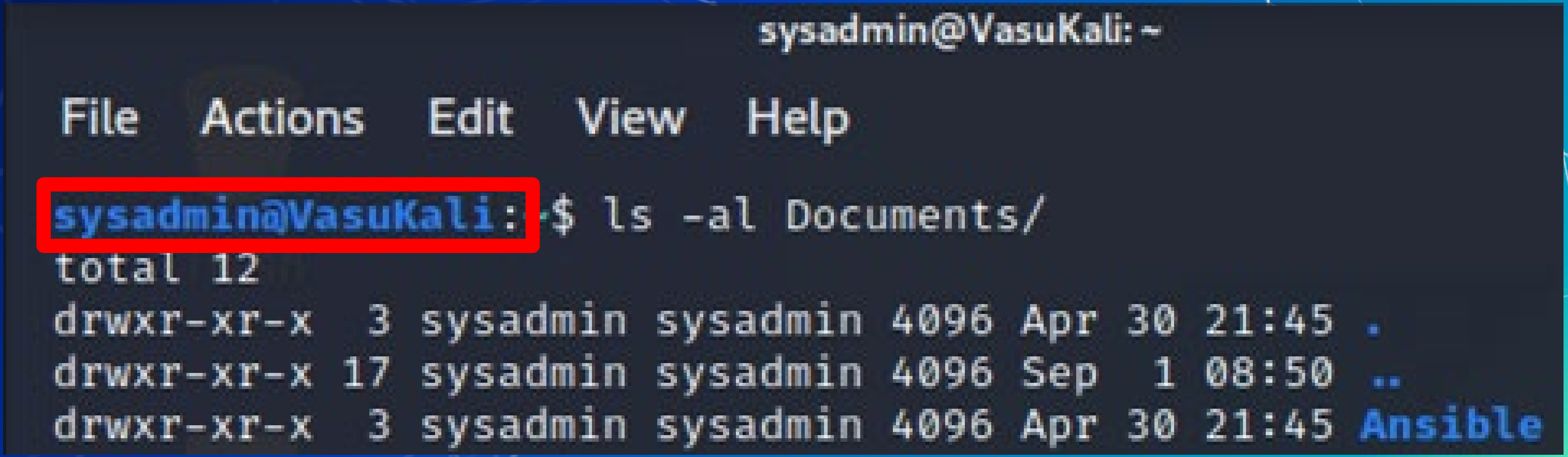
“Shell” “Bash”

“Terminal”

“Hacking Window”

Terminal

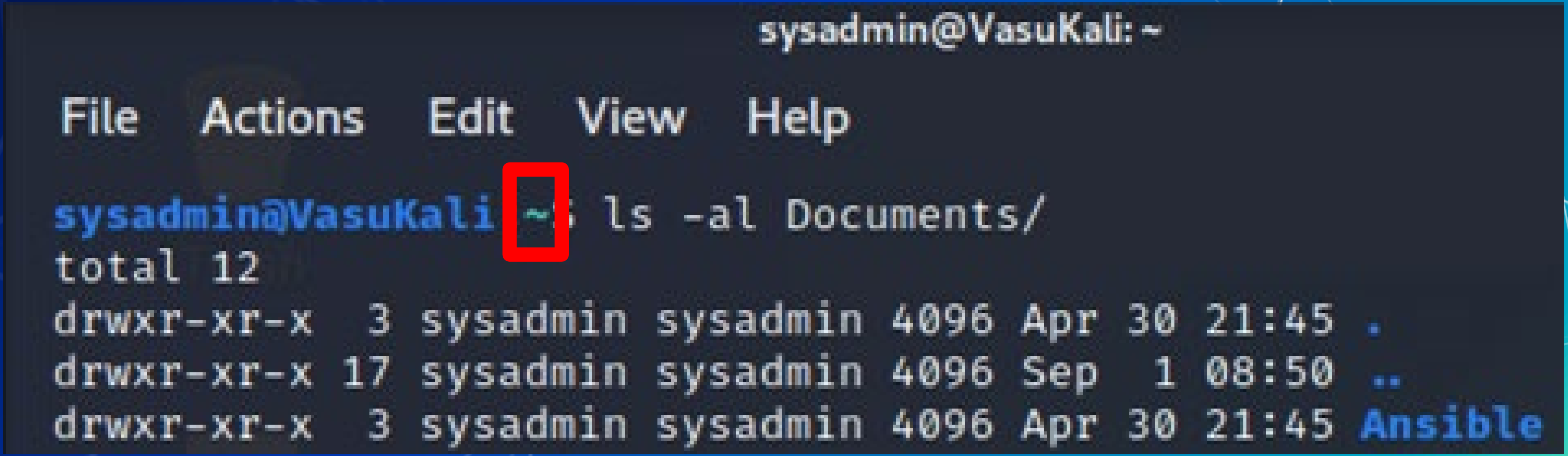
- `sysadmin`: The username of the current user logged in
- `VasuKali`: The hostname of the machine



```
sysadmin@VasuKali:~$ ls -al Documents/
total 12
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 .
drwxr-xr-x 17 sysadmin sysadmin 4096 Sep  1 08:50 ..
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 Ansible
```

Terminal

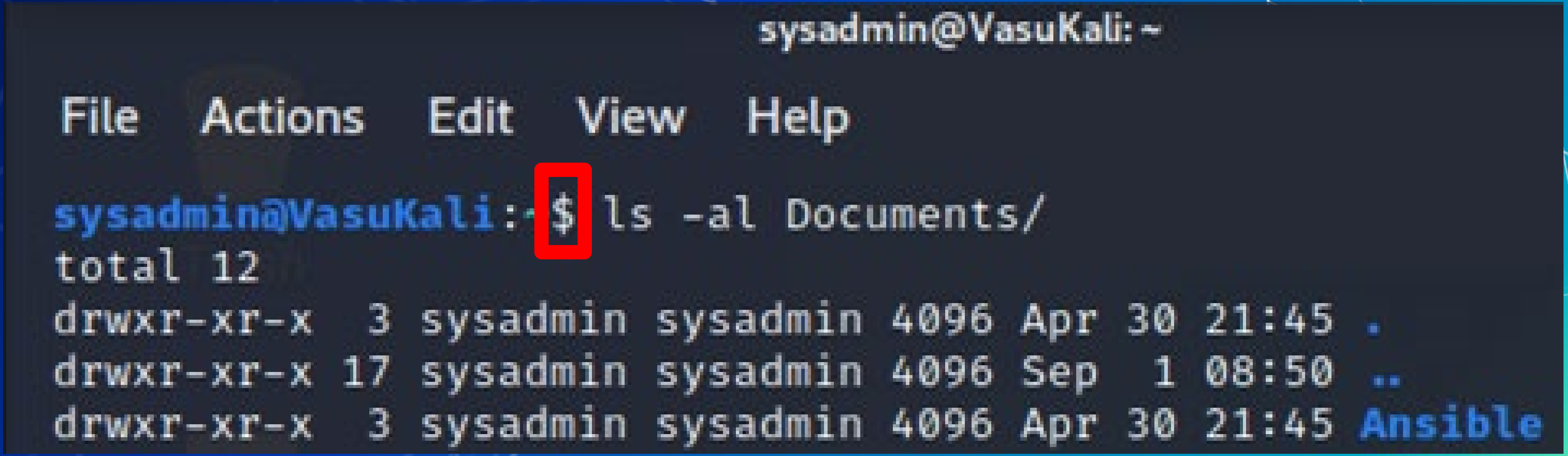
- ~ : Home directory shortcut



```
sysadmin@VasuKali:~$ ls -al Documents/
total 12
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 .
drwxr-xr-x 17 sysadmin sysadmin 4096 Sep  1 08:50 ..
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 Ansible
```

Terminal

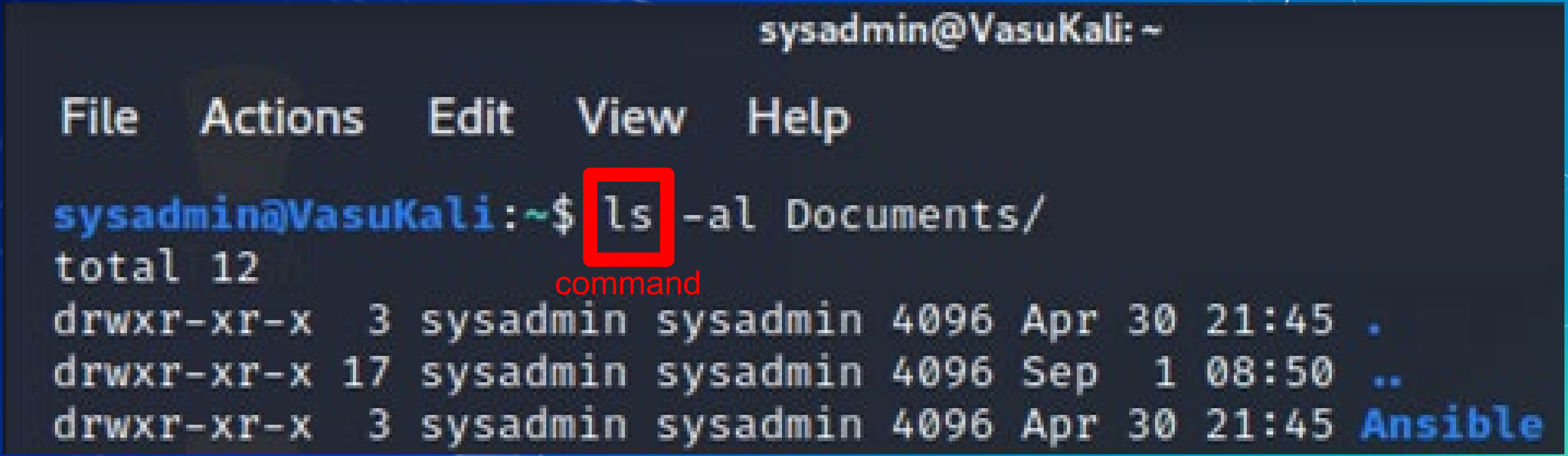
- $:The prompt symbol.
- Denotes the end of the command prompt
    - User’s keyboard input will appear next



```
sysadmin@VasuKali:~$ ls -al Documents/
total 12
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 .
drwxr-xr-x 17 sysadmin sysadmin 4096 Sep  1 08:50 ..
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 Ansible
```

Commands

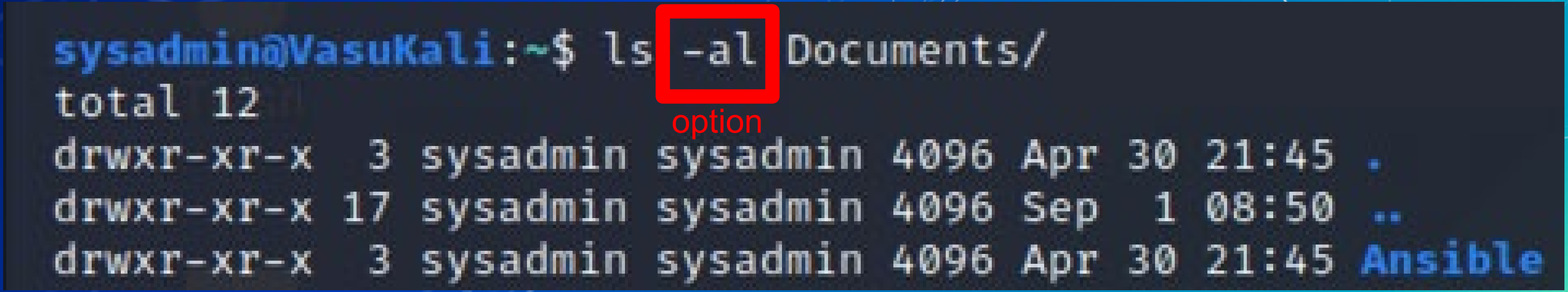
- Command
  - An instruction given by a user telling a computer to do something



```
sysadmin@VasuKali:~$ ls -al Documents/
total 12
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 .
drwxr-xr-x 17 sysadmin sysadmin 4096 Sep  1 08:50 ..
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 Ansible
```

Commands

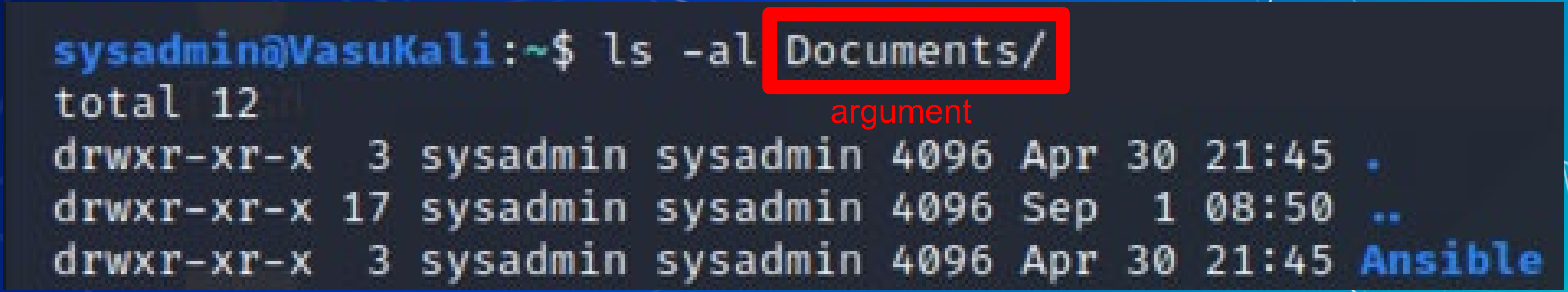
- Option
  - may follow after commands
  - Could be one or more to modify what the command does
  - Start with one/two dashes (ex: -p, --print) in order to differentiate them from arguments



```
sysadmin@VasuKali:~$ ls -al Documents/
total 12
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 .
drwxr-xr-x 17 sysadmin sysadmin 4096 Sep  1 08:50 ..
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 Ansible
```

Commands

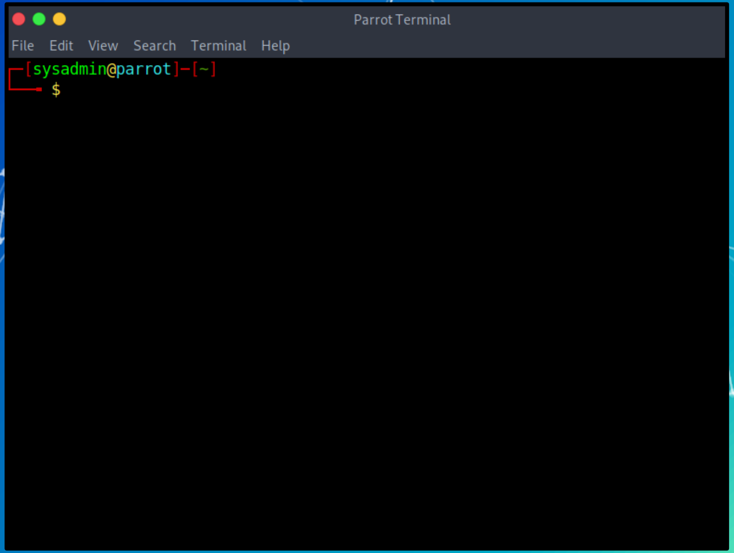
- Argument
  - File name referenced
  - Presented in < > in this presentation



```
sysadmin@VasuKali:~$ ls -al Documents/
total 12
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 .
drwxr-xr-x 17 sysadmin sysadmin 4096 Sep  1 08:50 ..
drwxr-xr-x  3 sysadmin sysadmin 4096 Apr 30 21:45 Ansible
```

What am I?

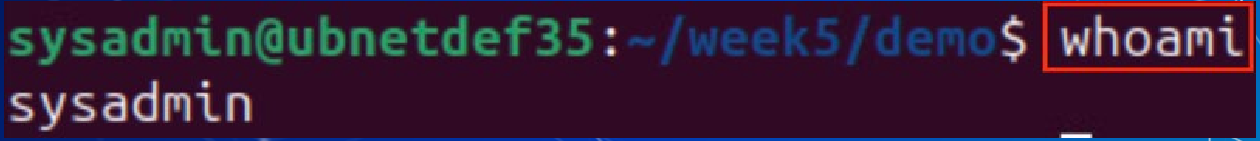
- Now that we've opened up the terminal, we can start to get our bearings on the system
- `whoami` : Current user
- `pwd` : Where you are
- `hostname` : Name of system you are on
- `ip a` : What is your network information
- `ps -aux` : What is running
- `clear` : clears the screen



```
┌─[sysadmin@parrot]─[~]
└──╼ $
```

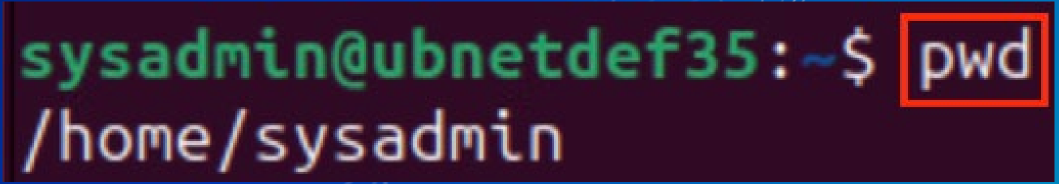
What am I?

- Whoami : show current user
  - Check which current account you are currently using in the terminal



```
sysadmin@ubnetdef35:~/week5/demo$ whoami
sysadmin
```

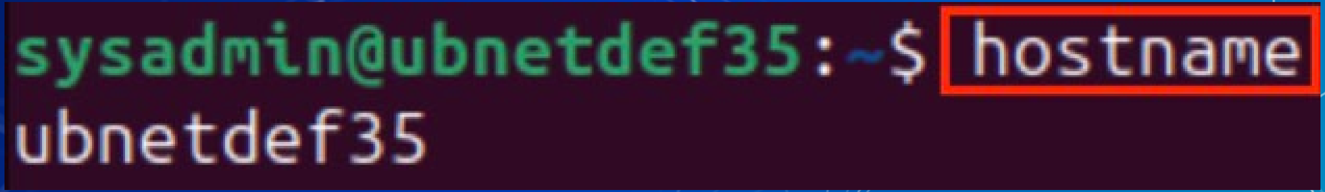
- pwd : Print Working Directory
  - Displays the full path of the current working directory



```
sysadmin@ubnetdef35:~$ pwd
/home/sysadmin
```

What am I?

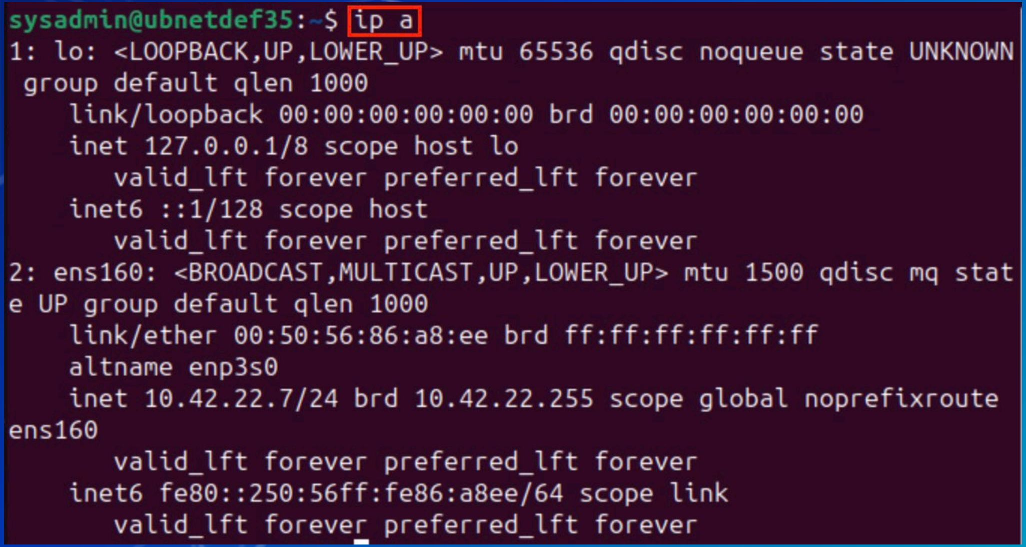
- `hostname` : Name of system you are on
  - System hostname—the unique name that identifies a device on the network



```
sysadmin@ubnetdef35:~$ hostname
ubnetdef35
```

What am I?

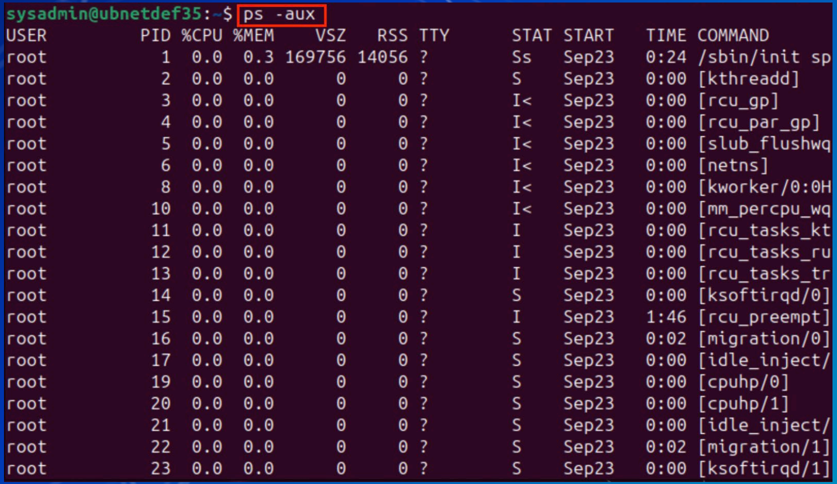
- `ip a`: What is your network information
  - lo: loopback interface, use for local communication
  - eth0: Ethernet network interface



```
sysadmin@ubnetdef35:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
 group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq stat
e UP group default qlen 1000
    link/ether 00:50:56:86:a8:ee brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 10.42.22.7/24 brd 10.42.22.255 scope global noprefixroute
ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe86:a8ee/64 scope link
       valid_lft forever preferred_lft forever
```

What am I?

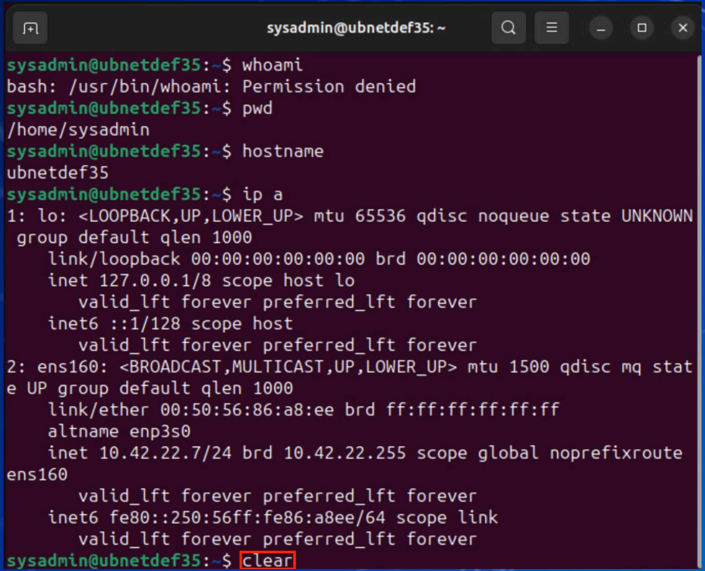
- `ps -aux` : process status
  - Shows (**a**)ll the processes
  - With (**u**)sernames
  - Including processes not started from the terminal (**x**)



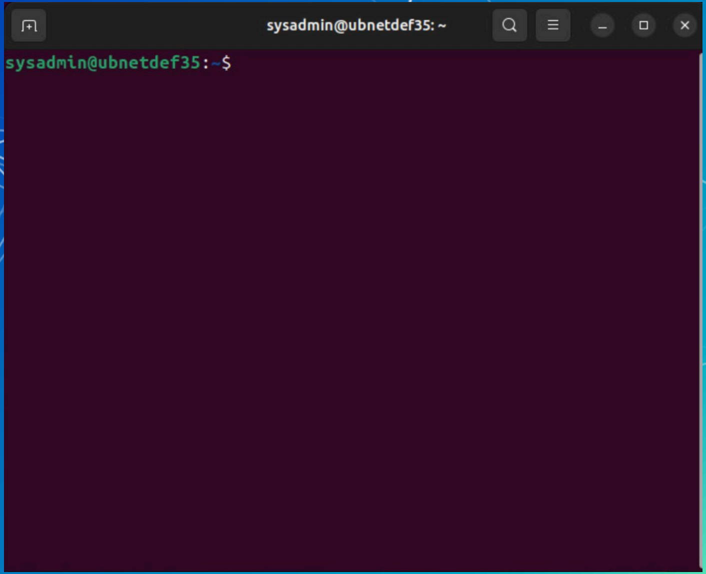
```
sysadmin@ubnetdef35:~$ ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.3 169756 14056 ?        Ss   Sep23   0:24 /sbin/init sp
root           2  0.0  0.0      0     0 ?        S    Sep23   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   Sep23   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   Sep23   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   Sep23   0:00 [slub_flushwq
root           6  0.0  0.0      0     0 ?        I<   Sep23   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   Sep23   0:00 [kworker/0:0H
root          10  0.0  0.0      0     0 ?        I<   Sep23   0:00 [mm_percpu_wq
root          11  0.0  0.0      0     0 ?        I    Sep23   0:00 [rcu_tasks_kt
root          12  0.0  0.0      0     0 ?        I    Sep23   0:00 [rcu_tasks_ru
root          13  0.0  0.0      0     0 ?        I    Sep23   0:00 [rcu_tasks_tr
root          14  0.0  0.0      0     0 ?        S    Sep23   0:00 [ksoftirqd/0]
root          15  0.0  0.0      0     0 ?        I    Sep23   1:46 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    Sep23   0:02 [migration/0]
root          17  0.0  0.0      0     0 ?        S    Sep23   0:00 [idle_inject/
root          19  0.0  0.0      0     0 ?        S    Sep23   0:00 [cpuhp/0]
root          20  0.0  0.0      0     0 ?        S    Sep23   0:00 [cpuhp/1]
root          21  0.0  0.0      0     0 ?        S    Sep23   0:00 [idle_inject/
root          22  0.0  0.0      0     0 ?        S    Sep23   0:02 [migration/1]
root          23  0.0  0.0      0     0 ?        S    Sep23   0:00 [ksoftirqd/1]
```

What am I?

- `clear` : clears the screen
  - Does not clear the history



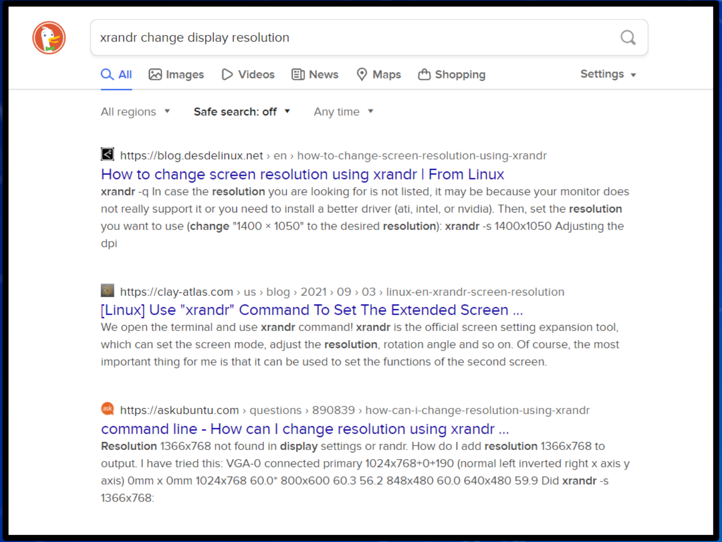
```
sysadmin@ubnetdef35:~$ whoami
bash: /usr/bin/whoami: Permission denied
sysadmin@ubnetdef35:~$ pwd
/home/sysadmin
sysadmin@ubnetdef35:~$ hostname
ubnetdef35
sysadmin@ubnetdef35:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
 group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq stat
e UP group default qlen 1000
    link/ether 00:50:56:86:a8:ee brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 10.42.22.7/24 brd 10.42.22.255 scope global noprefixroute
ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe86:a8ee/64 scope link
       valid_lft forever preferred_lft forever
sysadmin@ubnetdef35:~$ clear
```



```
sysadmin@ubnetdef35:~$
```

Commands? Memorization?

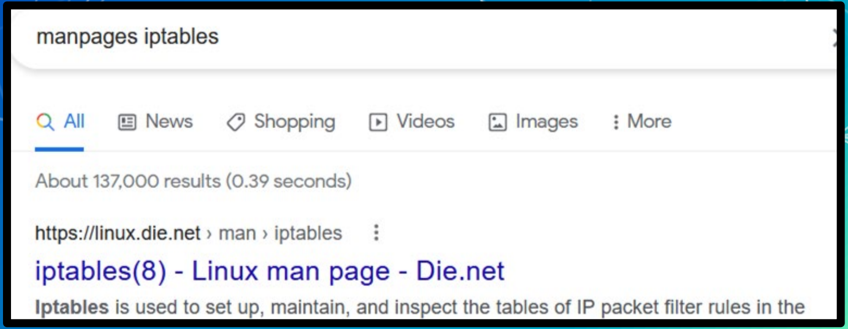
- **Look it up.** It’s what I do, it’s what Ken Smith does, it’s what everyone does.
  - Best way to learn/troubleshoot anything linux related
- This lecture covers ~20/30 of the most important/useful commands



https://blog.desdelinux.net › en › how-to-change-screen-resolution-using-xrandr
How to change screen resolution using xrandr | From Linux
xrandr -q In case the **resolution** you are looking for is not listed, it may be because your monitor does not really support it or you need to install a better driver (ati, intel, or nvidia). Then, set the **resolution** you want to use (**change** "1400 × 1050" to the desired **resolution**): **xrandr** -s 1400x1050 Adjusting the dpi

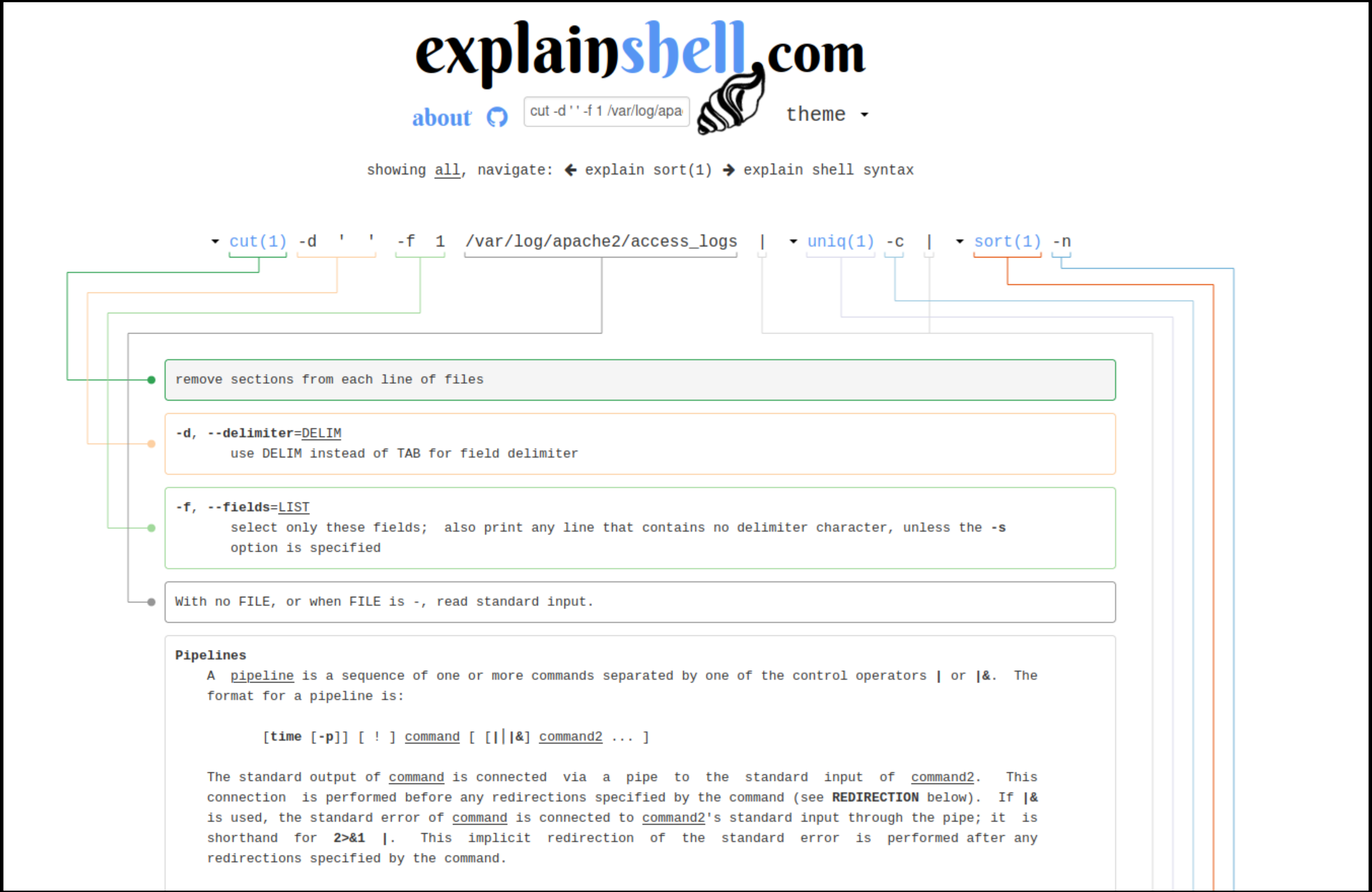
https://clay-atlas.com › us › blog › 2021 › 09 › 03 › linux-en-xrandr-screen-resolution
[Linux] Use "xrandr" Command To Set The Extended Screen ...
We open the terminal and use **xrandr** command! **xrandr** is the official screen setting expansion tool, which can set the screen mode, adjust the **resolution**, rotation angle and so on. Of course, the most important thing for me is that it can be used to set the functions of the second screen.

https://askubuntu.com › questions › 890839 › how-can-i-change-resolution-using-xrandr
command line - How can I change resolution using xrandr ...
Resolution 1366x768 not found in **display** settings or randr. How do I add **resolution** 1366x768 to output. I have tried this: VGA-0 connected primary 1024x768+0+190 (normal left inverted right x axis y axis) 0mm x 0mm 1024x768 60.0* 800x600 60.3 56.2 848x480 60.0 640x480 59.9 Did **xrandr** -s 1366x768:



About 137,000 results (0.39 seconds)

https://linux.die.net › man › iptables
iptables(8) - Linux man page - Die.net
Iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the



showing all, navigate: ← explain sort(1) → explain shell syntax

```
cut(1) -d ' ' -f 1 /var/log/apache2/access_logs | uniq(1) -c | sort(1) -n
```

remove sections from each line of files

-d, --delimiter=DELIM
    use DELIM instead of TAB for field delimiter

-f, --fields=LIST
    select only these fields; also print any line that contains no delimiter character, unless the **-s** option is specified

With no FILE, or when FILE is -, read standard input.

Pipelines

A pipeline is a sequence of one or more commands separated by one of the control operators **|** or **|&**. The format for a pipeline is:

```
[time [-p]] [ ! ] command [ [|||&] command2 ... ]
```

The standard output of command is connected via a pipe to the standard input of command2. This connection is performed before any redirections specified by the command (see **REDIRECTION** below). If **|&** is used, the standard error of command is connected to command2's standard input through the pipe; it is shorthand for **2>&1 |**. This implicit redirection of the standard error is performed after any redirections specified by the command.

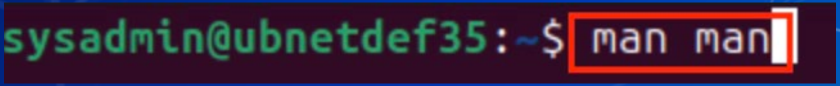
Information Commands

If you're stuck and the suffix `--help` isn't helping,

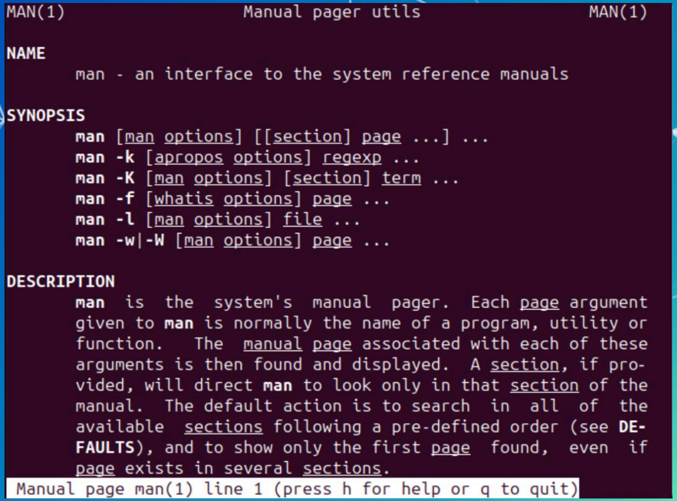
- `man` – Manual
  - Syntax: man <command>
- `whatis` – displays one-line manual page description
  - Syntax: whatis <command>

Information Commands

- man
  - Manual
  - Fully detailed description of what each command suffix does.
  - Syntax: `man <tool>`



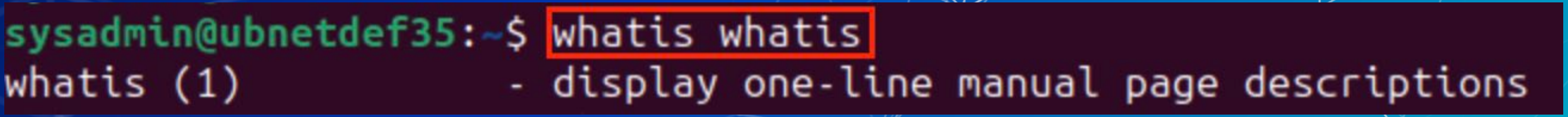
```
sysadmin@ubnetdef35:~$ man man
```



```
MAN(1)                    Manual pager utils                    MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...

DESCRIPTION
       man  is  the  system's  manual  pager.  Each page argument
       given to man is normally the name of a program, utility or
       function.   The  manual page associated with each of these
       arguments is then found and displayed.  A section, if pro-
       vided, will direct man to look only in that section of the
       manual.  The default action is to search  in  all  of  the
       available  sections following a pre-defined order (see DE-
       FAULTS), and to show only the first page  found,  even  if
       page exists in several sections.
 Manual page man(1) line 1 (press h for help or q to quit)
```

Information Commands

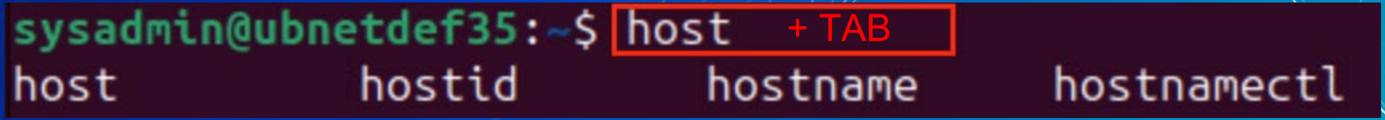
- whatis
  - Fully detailed description of what each command suffix does.



```
sysadmin@ubnetdef35:~$ whatis whatis
whatis (1)           - display one-line manual page descriptions
```

Tab Tab Tab Tab Tab Tab Tab Tab Tab Tab Tab…

- Many shells use tab to autocomplete or suggest autocompletion
- This is so useful it gets its own slide



```
sysadmin@ubnetdef35:~$ host
host            hostid          hostname        hostnamectl
```

+ TAB

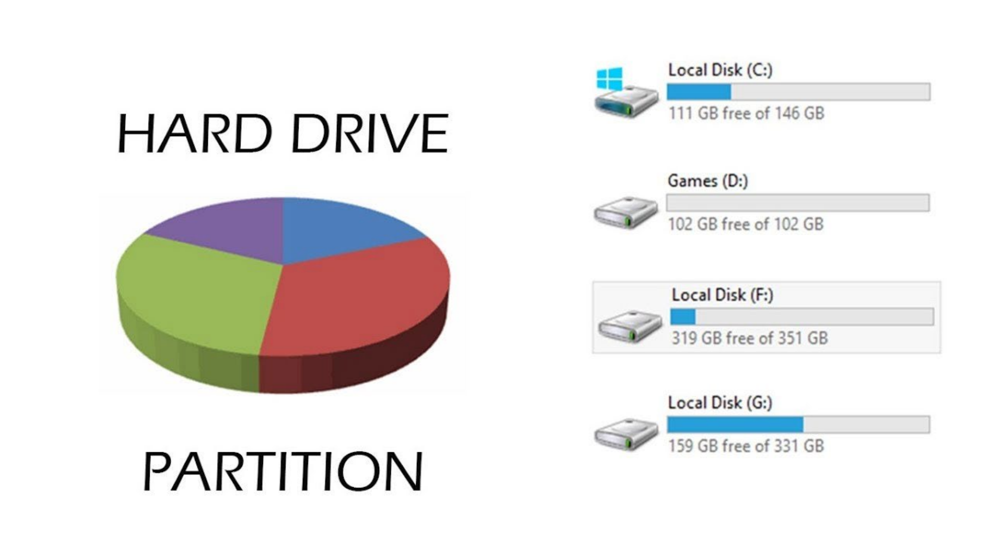
Questions ?

Agenda

- Linux Basics
  - What is Linux? What is Kernal? What is Linux Distribution
  - Terminal
  - Commands– What Am I ? & Get Help!
- File System
  - Navigate File System
  - Interact with Files
  - Text Editors
- In class Activity – Linux CTF
- Users & Groups
- File Permission
- Others

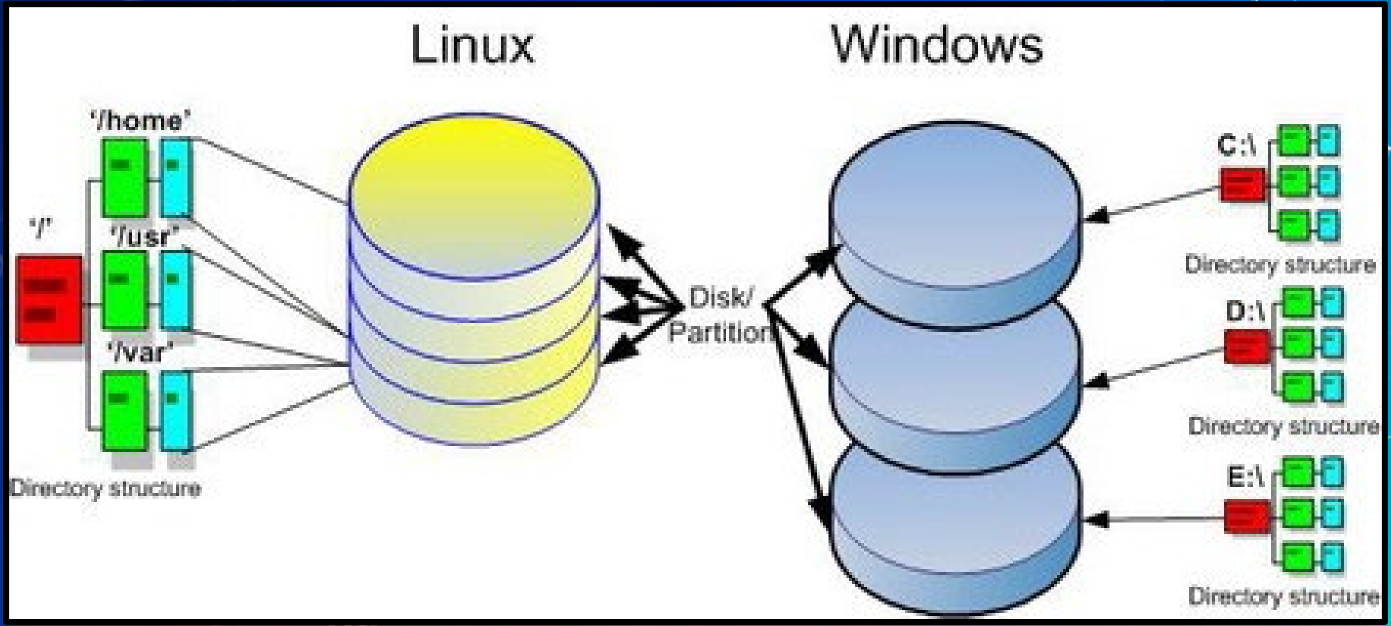
Disk Partition

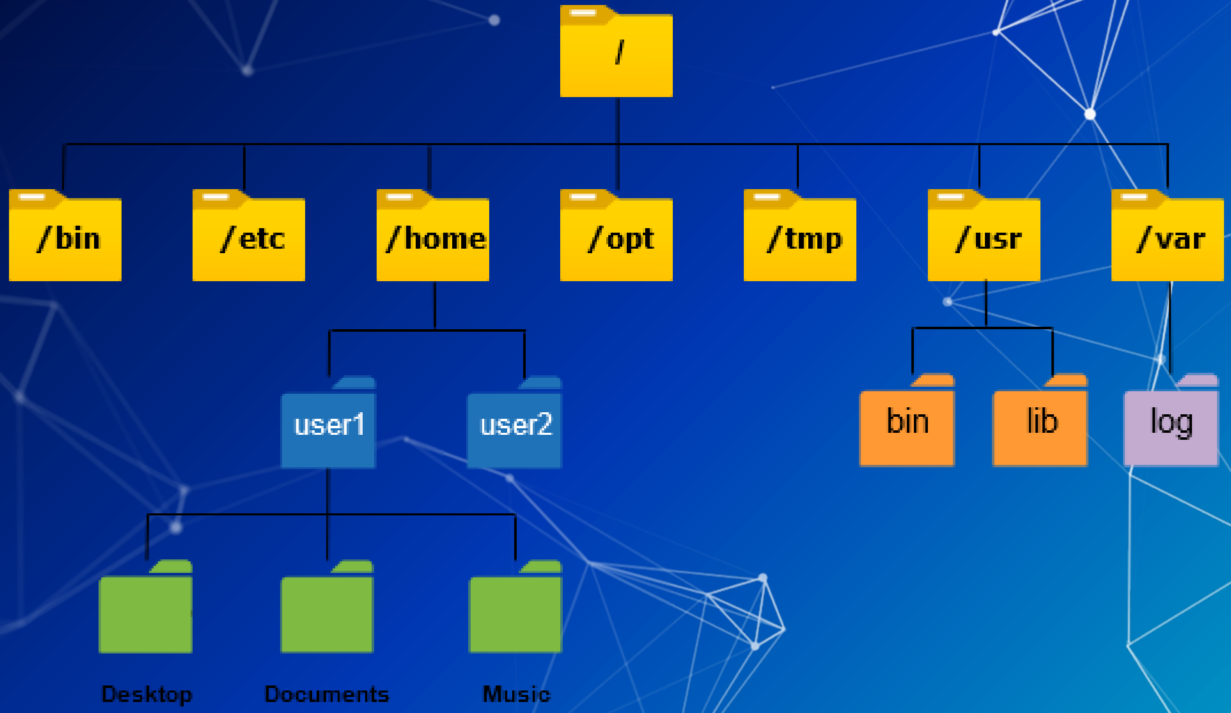
- Divisions of storage devices, like hard drives or SSDs, into isolated sections that function as separate logical units.



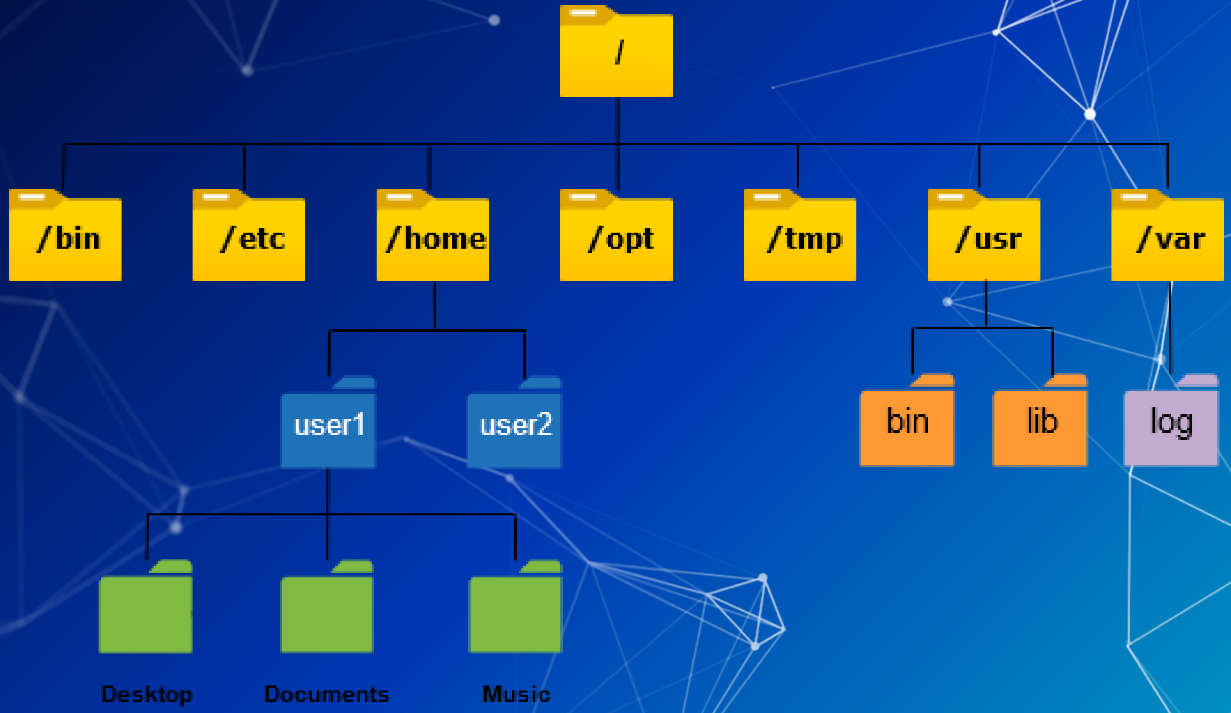
Understanding the filesystem

- Everything is built of the / (root) directory
- Everything is a file

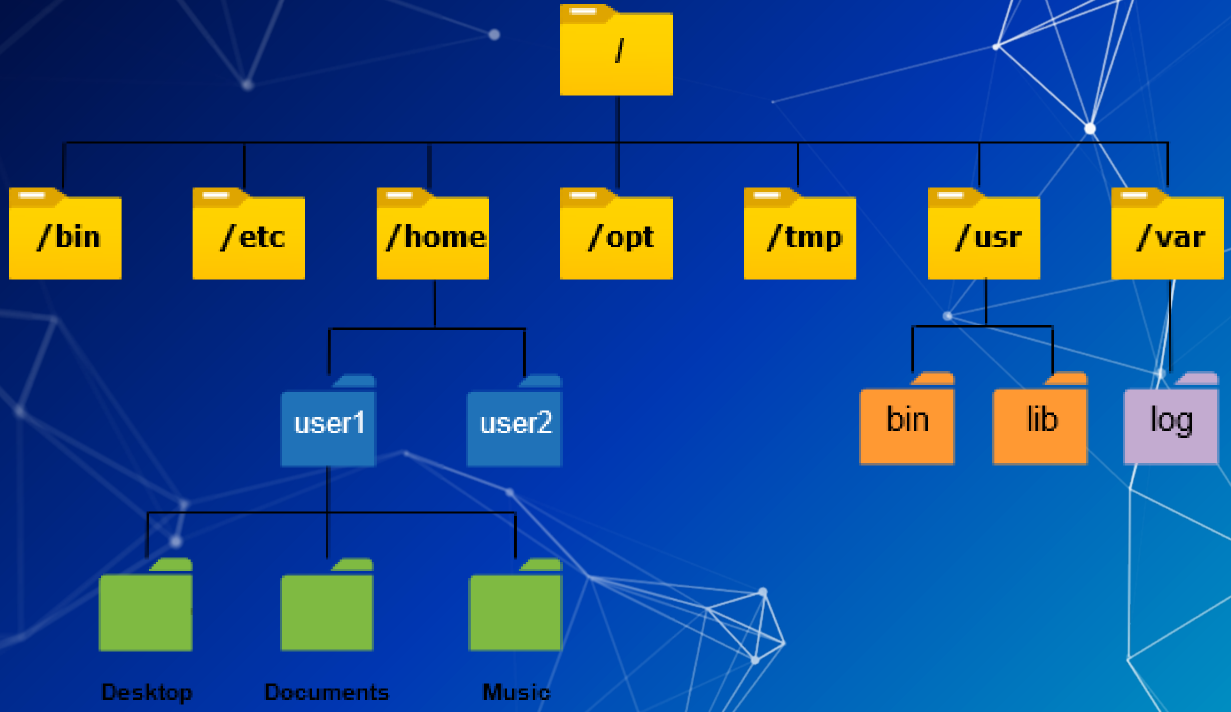




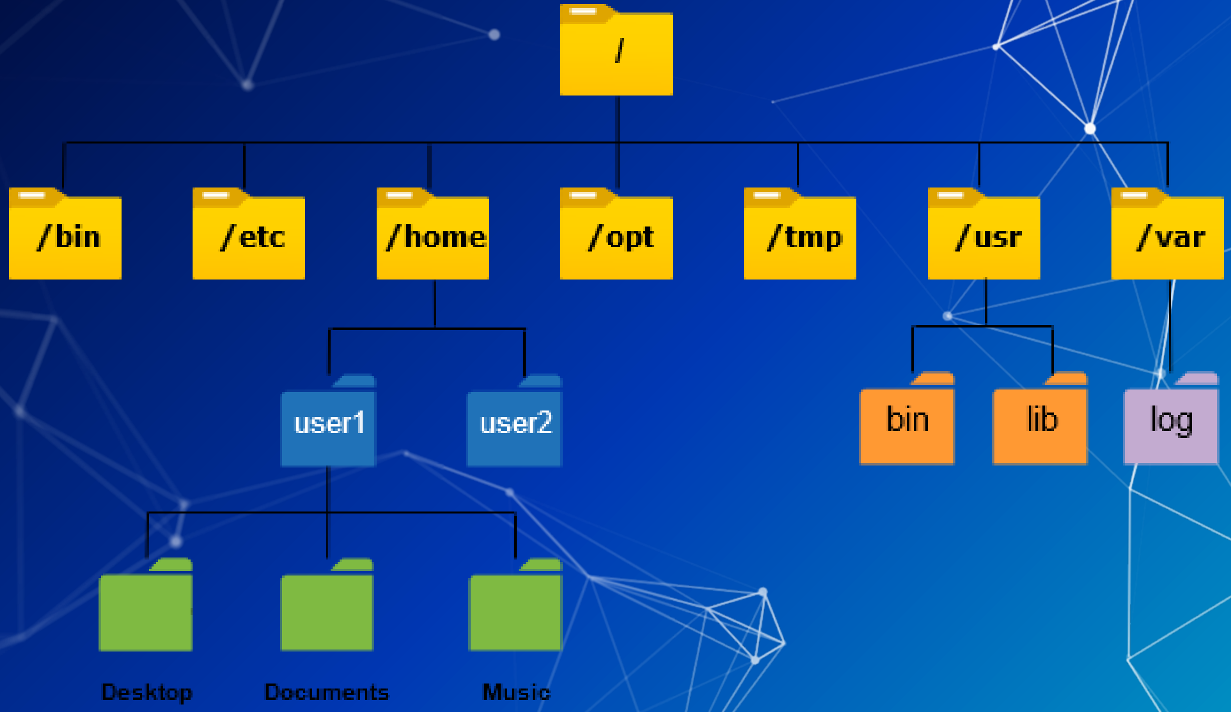
- / (`root`) root directory of the entire system hierarchy.
  - Everything starts at root.
  - Nothing is higher than root.



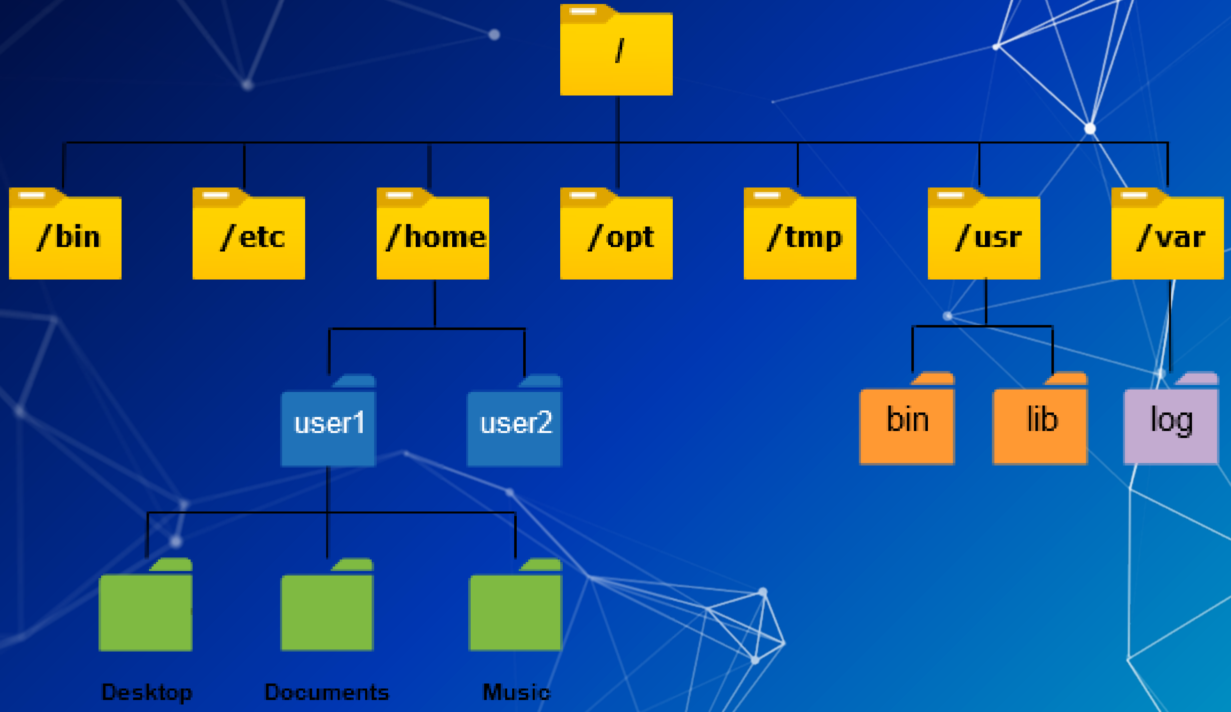
- `/bin/` essential command binaries
  - `whoami`, `pwd`, `cp` are all stored here



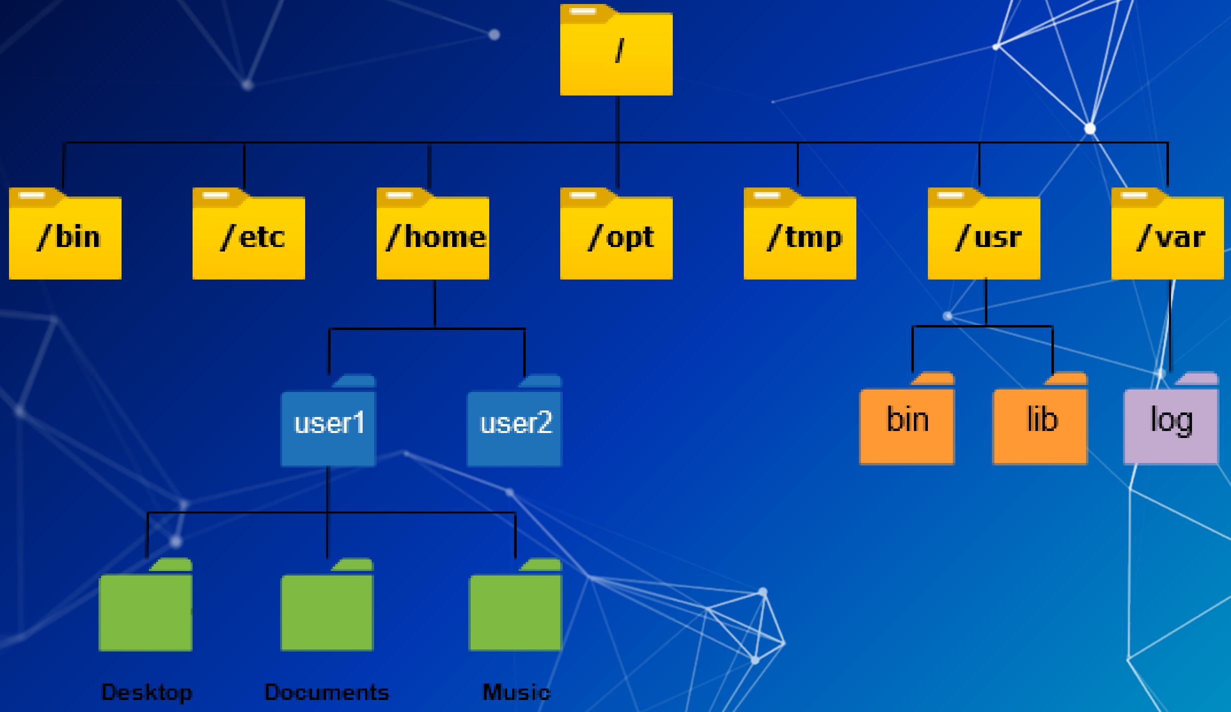
- `/etc/` specific system-wide configuration files
  - We edited the network configuration file in here for HW02



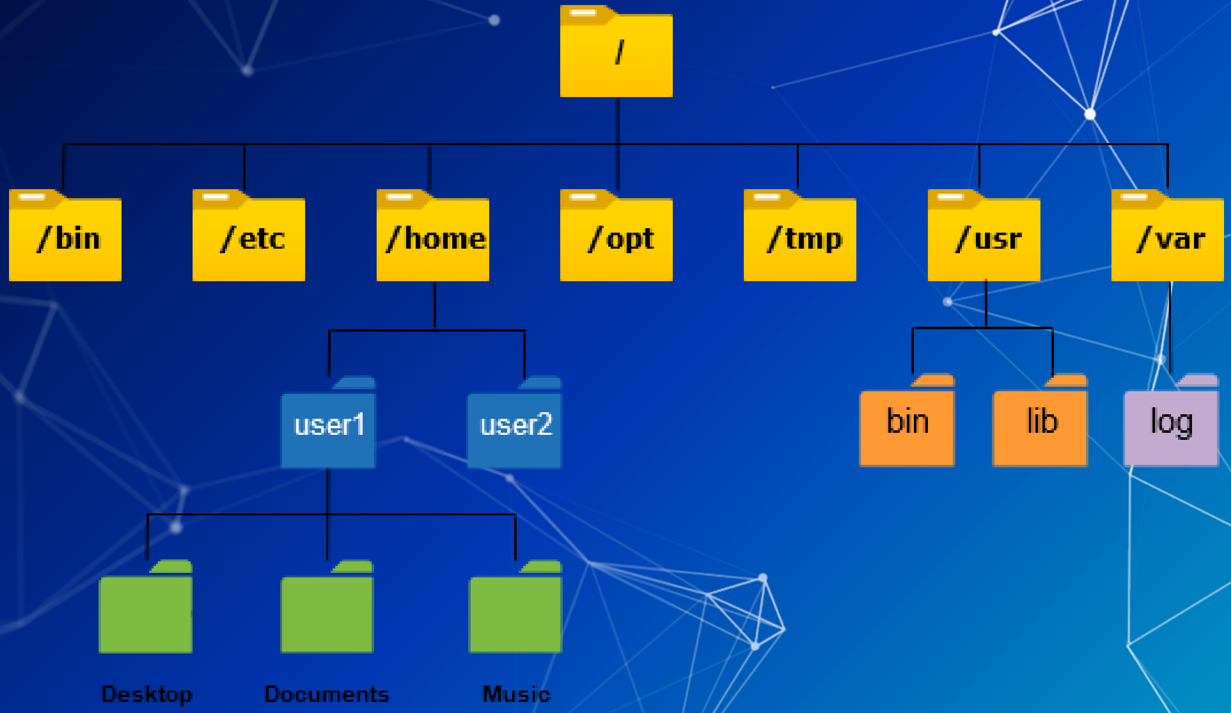
- /home/ Users' home directories, containing saved files, personal settings, etc.



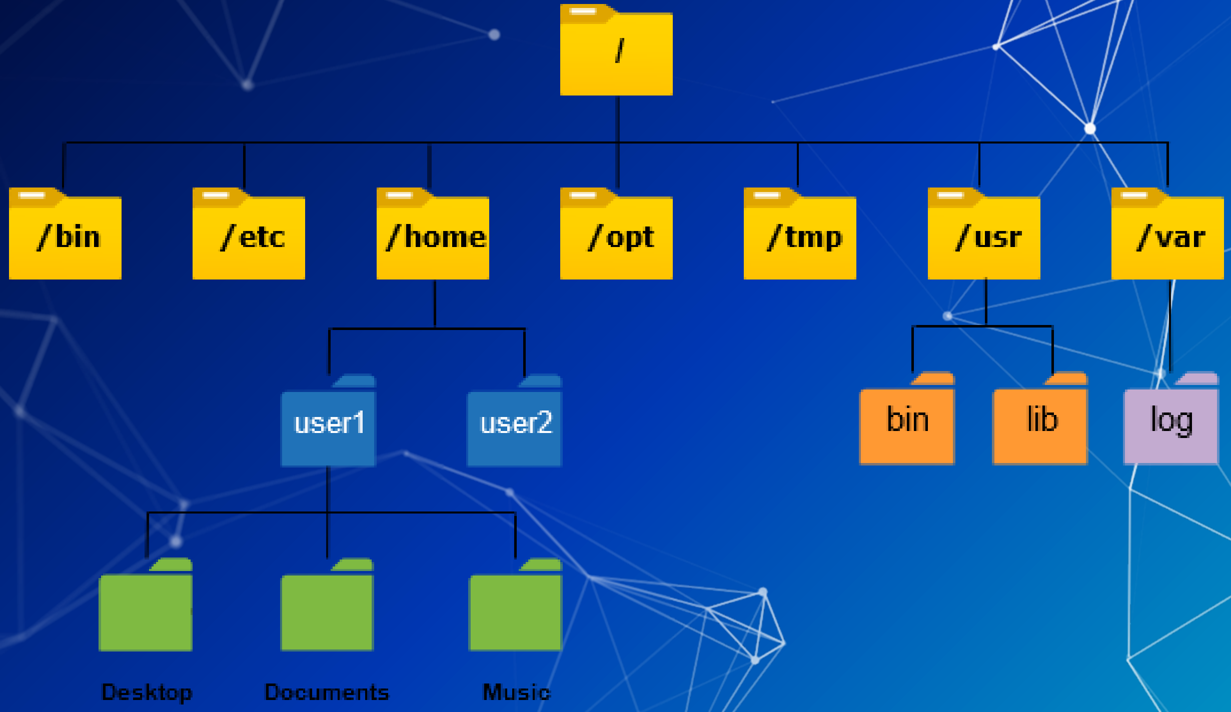
- `/opt/` Additional software and addons



- /tmp/ Temporary files
  - Typically not saved after reboots



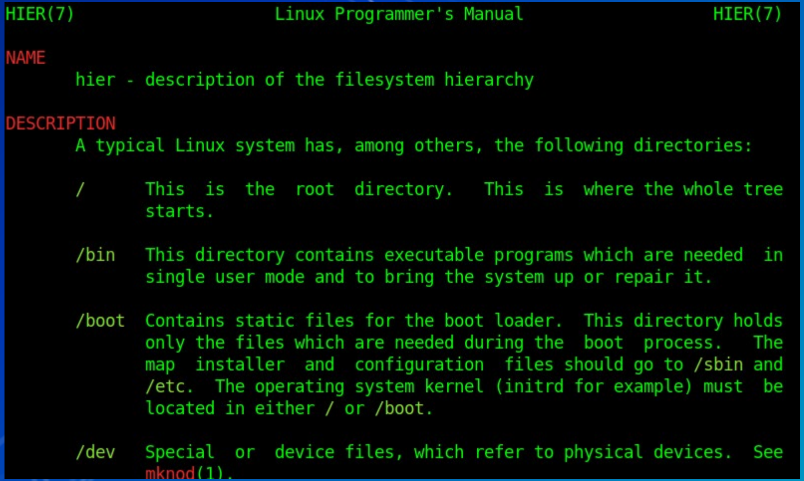
- /usr/ user level binaries and applications



- /var/ Variable files - content of the file is expected to continually change during normal operation of the system
  - System logs are stored here

Linux FHS

- There are more key paths on the filesystem that we haven't covered
- These are specified in the Filesystem Hierarchy Standard (FHS)
- You can access that information from your terminal with `man hier`
- https://refspecs.linuxfoundation.org/fhs.shtml



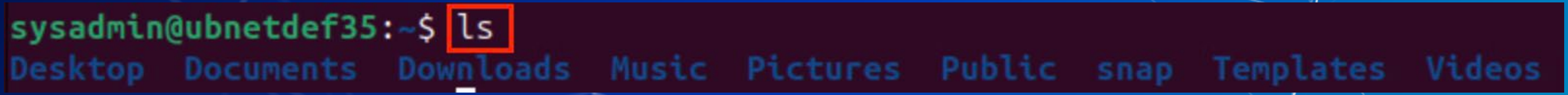
```
HIER(7)                   Linux Programmer's Manual                  HIER(7)

NAME
       hier - description of the filesystem hierarchy

DESCRIPTION
       A typical Linux system has, among others, the following directories:

       /      This  is  the  root  directory.   This  is  where the whole tree
              starts.

       /bin   This directory contains executable programs which are needed  in
              single user mode and to bring the system up or repair it.

       /boot  Contains static files for the boot loader.  This directory holds
              only the files which are needed during the  boot  process.   The
              map  installer  and  configuration  files should go to /sbin and
              /etc.  The operating system kernel (initrd for example) must  be
              located in either / or /boot.

       /dev   Special  or  device files, which refer to physical devices.  See
              mknod(1).
```

Questions ?

How do we navigate the file system?

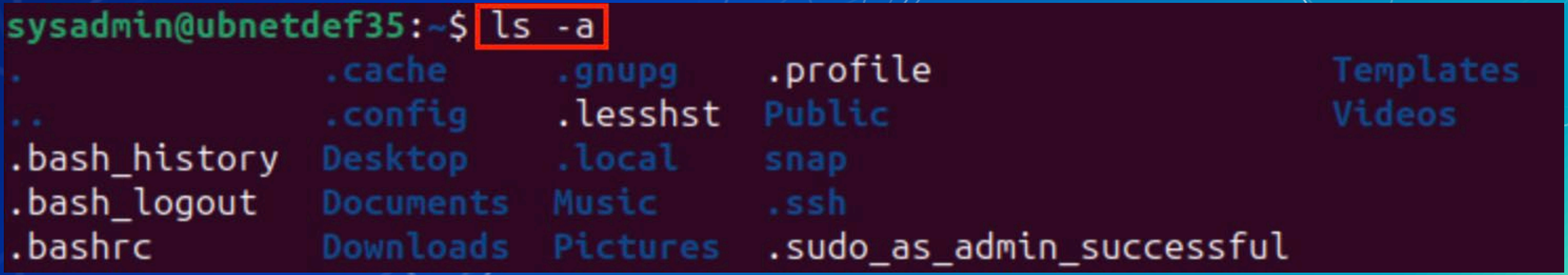
Navigating Directories

- `ls` - list files and directories in the current directory



```
sysadmin@ubnetdef35:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
```

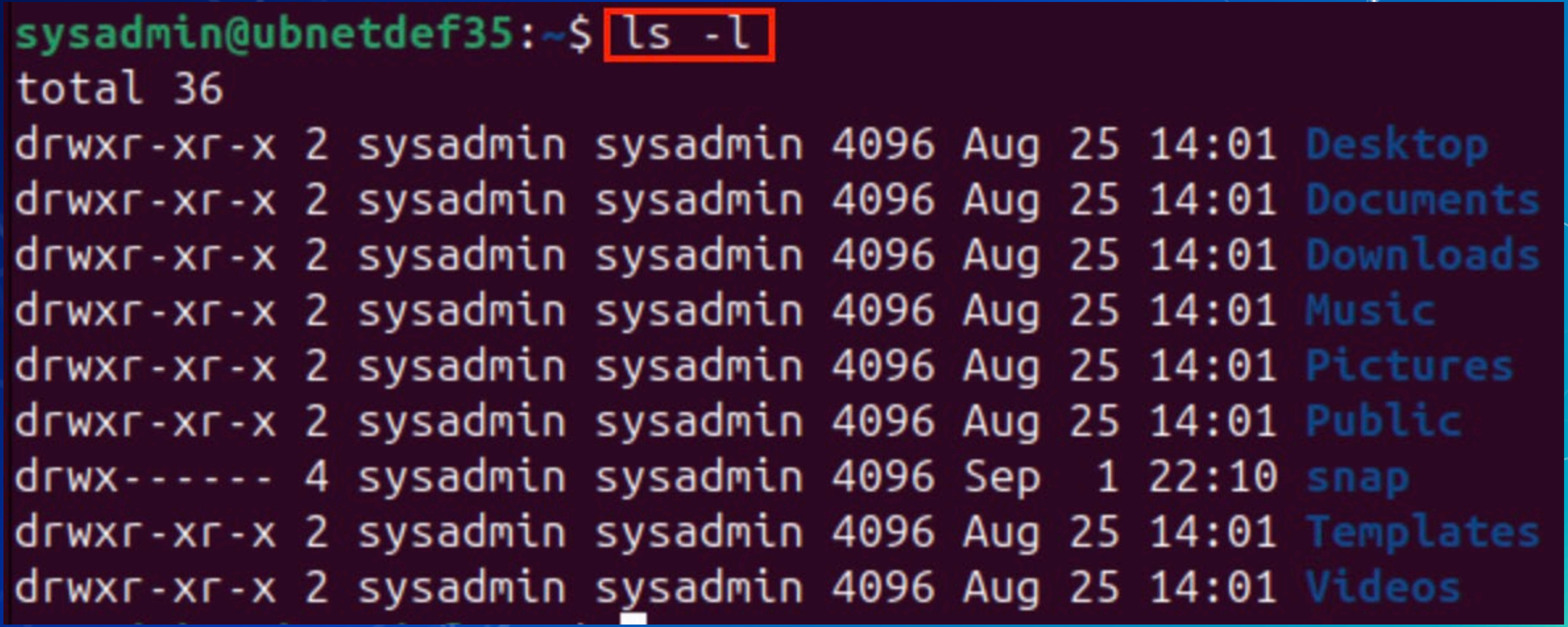
- `ls -a` : shows hidden files and directories
  - Files or directories that start with “ . ” are hidden.



```
sysadmin@ubnetdef35:~$ ls -a
.              .cache     .gnupg    .profile                   Templates
..             .config    .lesshst  Public                     Videos
.bash_history  Desktop    .local    snap
.bash_logout   Documents  Music     .ssh
.bashrc        Downloads  Pictures  .sudo_as_admin_successful
```

Navigating Directories

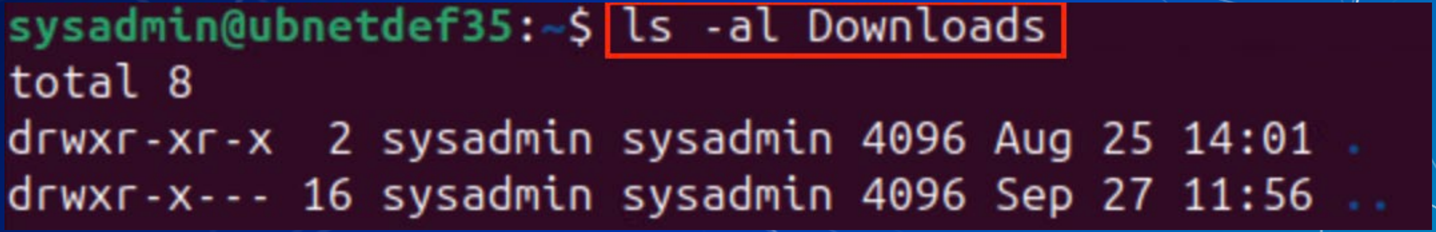
- `ls -l`: provides additional information such as file permissions, owner, group, file size, and modification date



```
sysadmin@ubnetdef35:~$ ls -l
total 36
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Desktop
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Documents
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Downloads
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Music
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Pictures
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Public
drwx------ 4 sysadmin sysadmin 4096 Sep  1 22:10 snap
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Templates
drwxr-xr-x 2 sysadmin sysadmin 4096 Aug 25 14:01 Videos
```

Navigating Directories

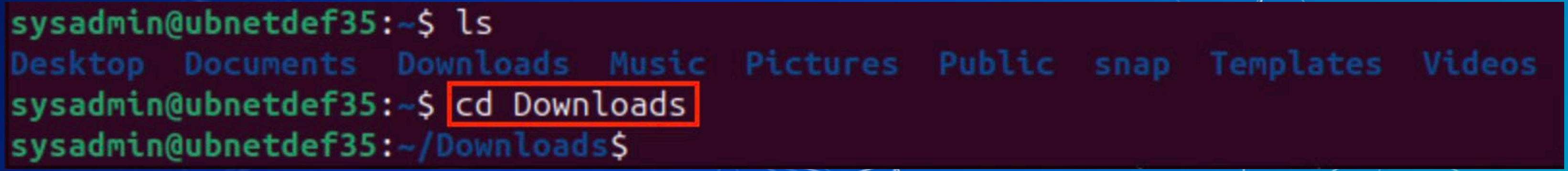
- `ls /path/to/directory`: list Files and Directories in a Specific Directory



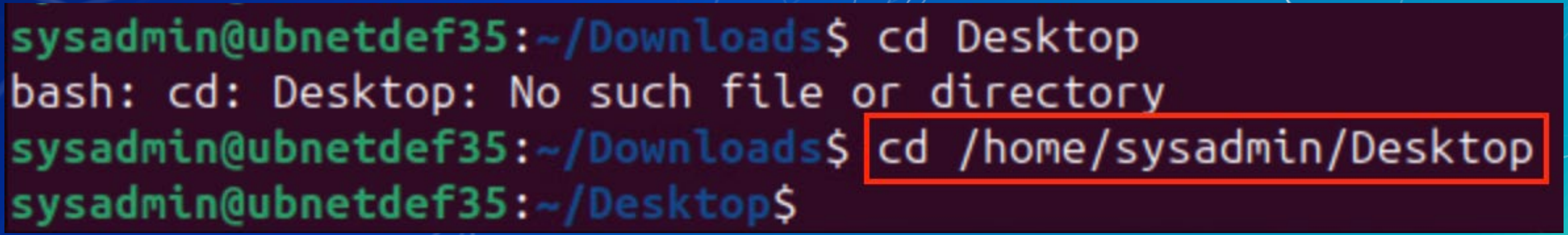
```
sysadmin@ubnetdef35:~$ ls -al Downloads
total 8
drwxr-xr-x  2 sysadmin sysadmin 4096 Aug 25 14:01 .
drwxr-x--- 16 sysadmin sysadmin 4096 Sep 27 11:56 ..
```

Navigating Directories

- cd - change directory: changes working directory
    - Syntax: `cd <relative/absolute path>`



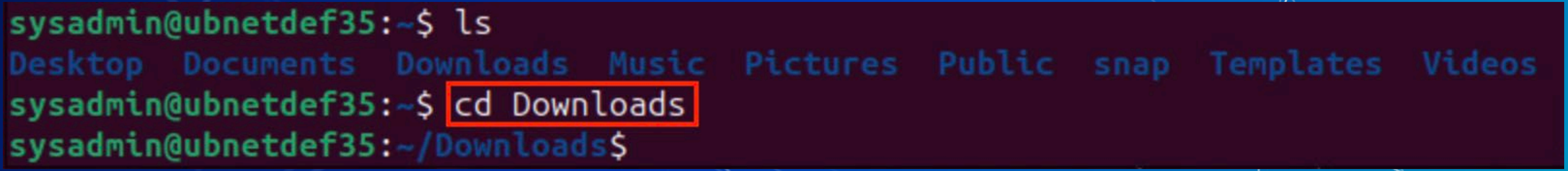
```
sysadmin@ubnetdef35:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
sysadmin@ubnetdef35:~$ cd Downloads
sysadmin@ubnetdef35:~/Downloads$
```



```
sysadmin@ubnetdef35:~/Downloads$ cd Desktop
bash: cd: Desktop: No such file or directory
sysadmin@ubnetdef35:~/Downloads$ cd /home/sysadmin/Desktop
sysadmin@ubnetdef35:~/Desktop$
```

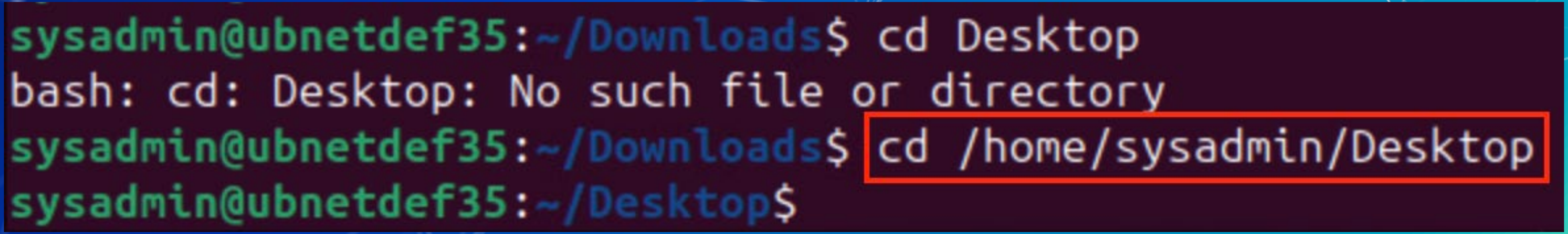
Relative vs Absolute Paths

- Relative Path
  - specifies the location of a file or directory relative to the current working directory
  - Start from the current directory you are in



```
sysadmin@ubnetdef35:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
sysadmin@ubnetdef35:~$ cd Downloads
sysadmin@ubnetdef35:~/Downloads$
```

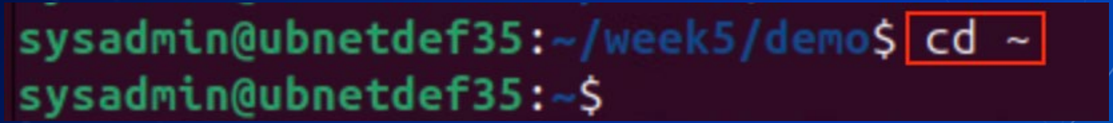
- Absolute Locations
  - Provides the complete and exact location of a file or directory
  - Start from the root directory of the file system ("/")



```
sysadmin@ubnetdef35:~/Downloads$ cd Desktop
bash: cd: Desktop: No such file or directory
sysadmin@ubnetdef35:~/Downloads$ cd /home/sysadmin/Desktop
sysadmin@ubnetdef35:~/Desktop$
```

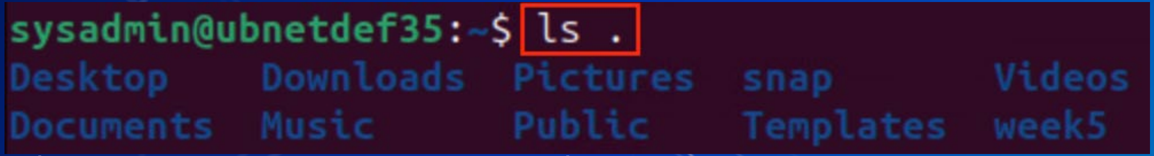
Shortcuts

- ~ Current user’s “home” directory (shortcut)



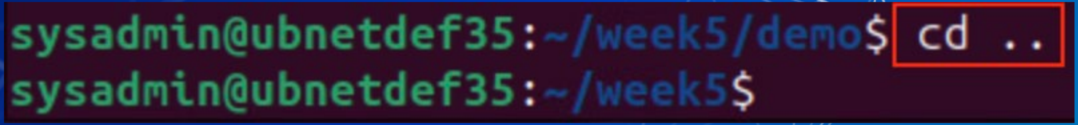
```
sysadmin@ubnetdef35:~/week5/demo$ cd ~
sysadmin@ubnetdef35:~$
```

- . The current directory



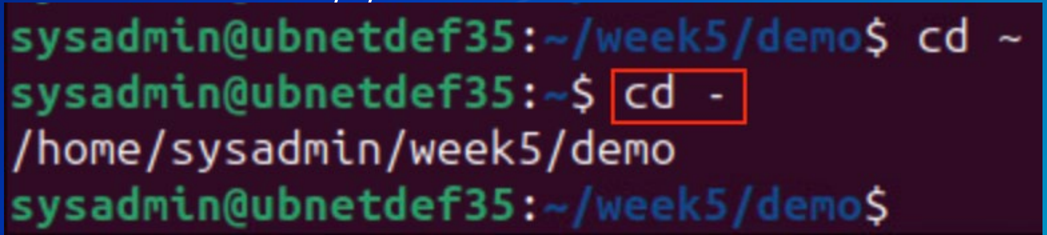
```
sysadmin@ubnetdef35:~$ ls .
Desktop     Downloads  Pictures  snap       Videos
Documents   Music      Public    Templates  week5
```

- . . The parent to your current directory

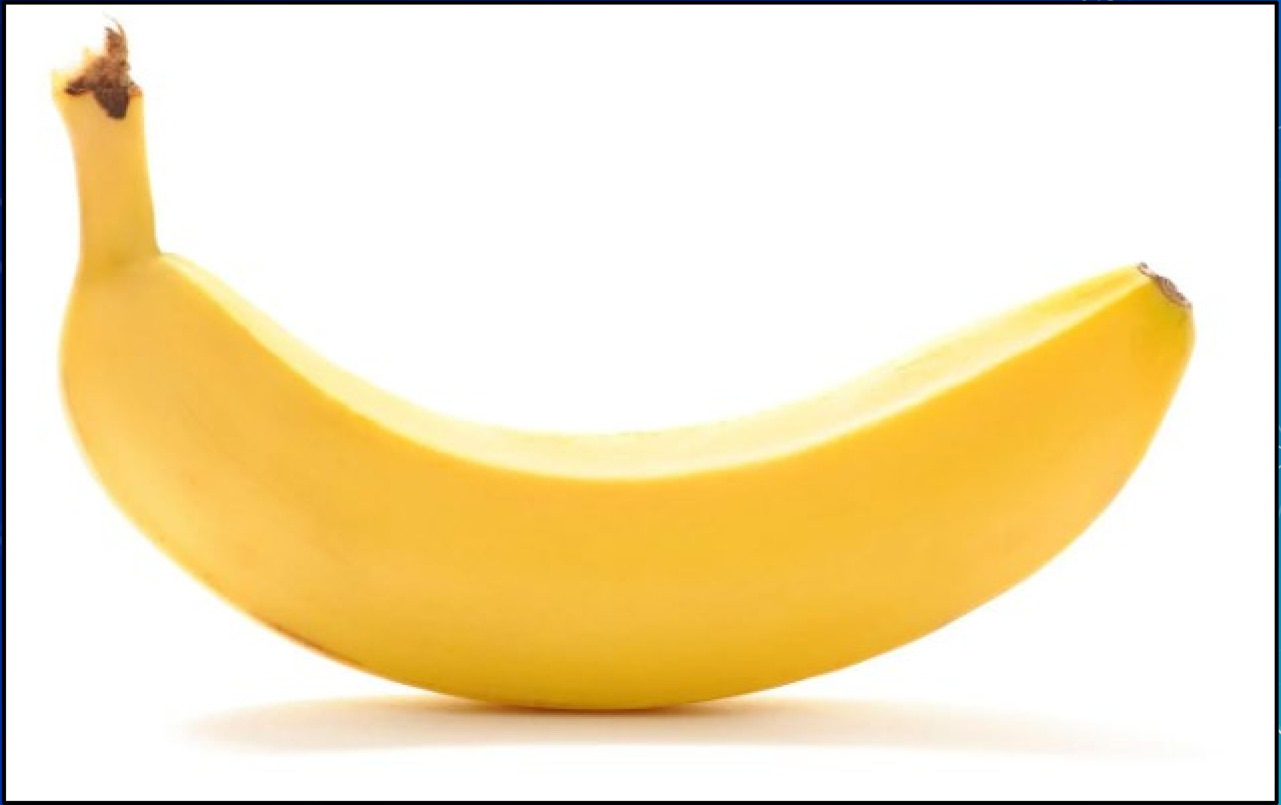


```
sysadmin@ubnetdef35:~/week5/demo$ cd ..
sysadmin@ubnetdef35:~/week5$
```

- – The last directory you went to

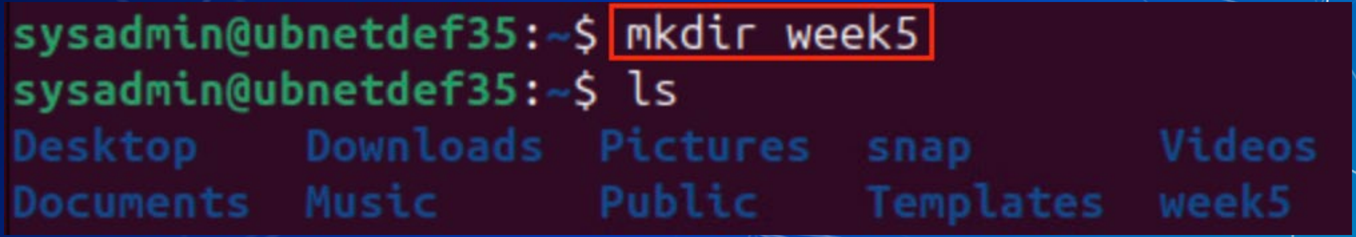


```
sysadmin@ubnetdef35:~/week5/demo$ cd ~
sysadmin@ubnetdef35:~$ cd -
/home/sysadmin/week5/demo
sysadmin@ubnetdef35:~/week5/demo$
```



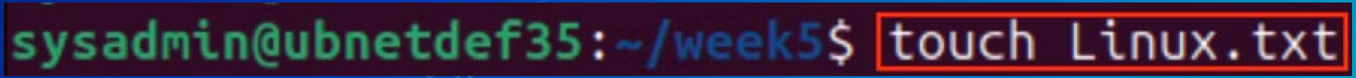
Interacting with files

- mkdir – Make Directory
    - Syntax: mkdir <Directory name>



```
sysadmin@ubnetdef35:~$ mkdir week5
sysadmin@ubnetdef35:~$ ls
Desktop     Downloads  Pictures  snap       Videos
Documents   Music      Public    Templates  week5
```

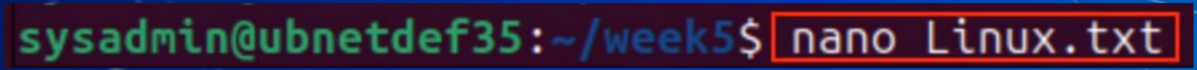
- touch
    - Syntax: touch <filename>
    - Creates an empty file with the filename provided



```
sysadmin@ubnetdef35:~/week5$ touch Linux.txt
```

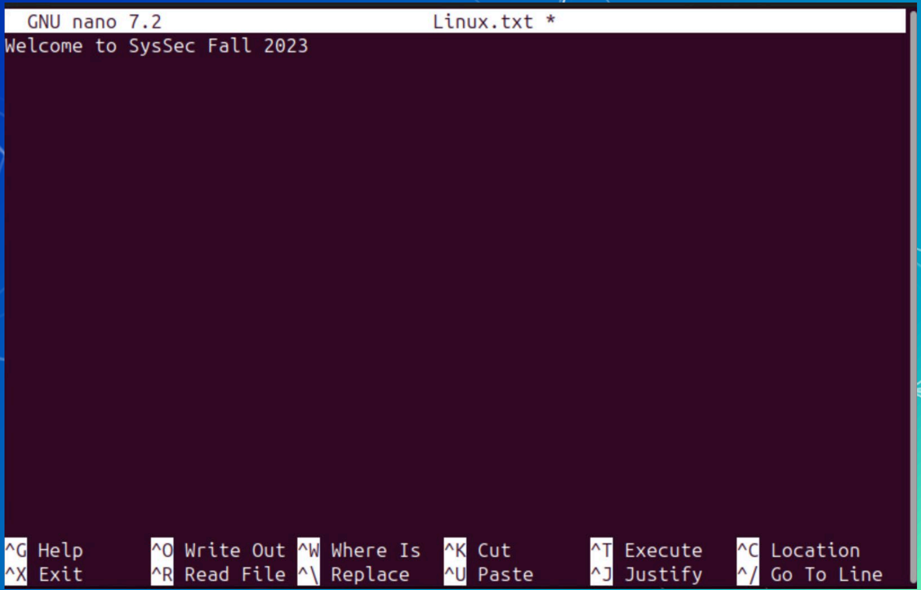
Interacting with files

- nano
  - Syntax: `nano <filename>`



```
sysadmin@ubnetdef35:~/week5$ nano Linux.txt
```

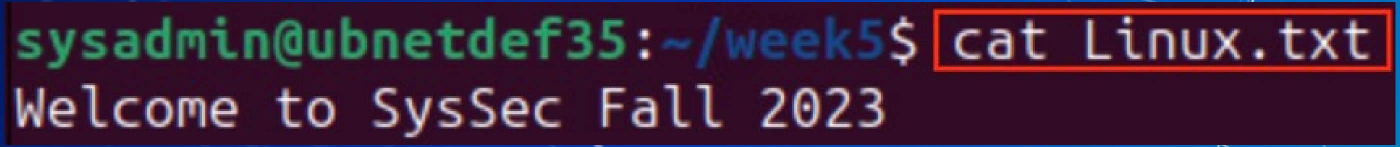
  - Exit: `Ctrl + X`
    `+ Press Y when prompted for buffer`
    `+ Press Enter`



```
  GNU nano 7.2                             Linux.txt *
Welcome to SysSec Fall 2023












^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

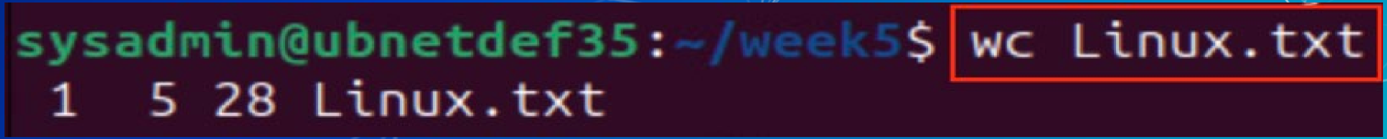
Interacting with files

- cat
  - Syntax: `cat <filename>`
  - Displays the contents of the file in the terminal.



```
sysadmin@ubnetdef35:~/week5$ cat Linux.txt
Welcome to SysSec Fall 2023
```

- wc: Word Count
  - Syntax: `wc <filename>`
  - Counts the number of lines, words and characters in a text file

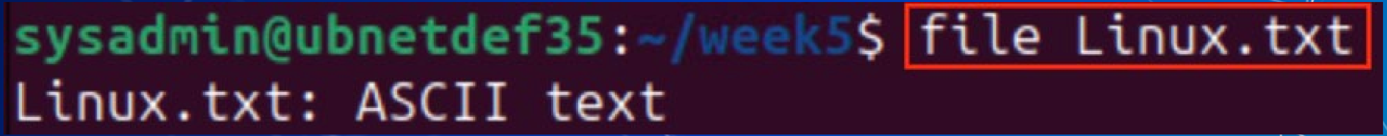


```
sysadmin@ubnetdef35:~/week5$ wc Linux.txt
 1  5 28 Linux.txt
```

-l (Line) -w (Words) -c (Characters)

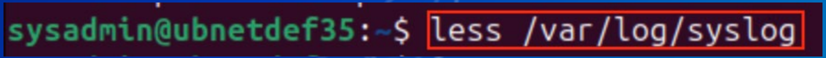
Interacting with files

- `file`
  - Syntax: `file <filename>`
  - Tells you the file type of the file

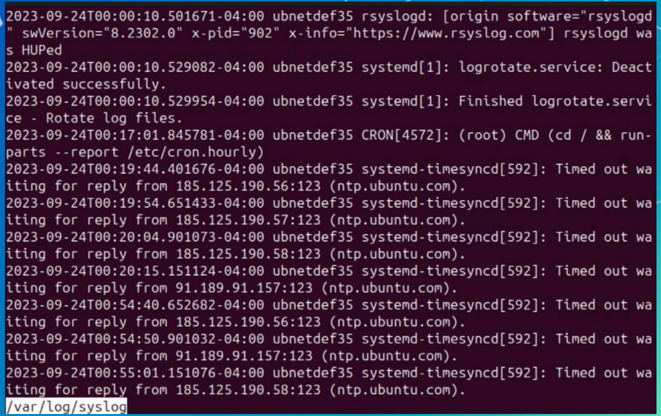


```
sysadmin@ubnetdef35:~/week5$ file Linux.txt
Linux.txt: ASCII text
```

- `less`
  - Syntax: `less <filename>`
  - Provides a scrollable version of cat
  - Use keyboard arrows to scroll up and down
  - Press `spacebar` : go to next page
  - Press `b` : back to last page
  - Press `q` to exit



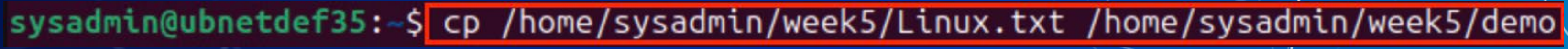
```
sysadmin@ubnetdef35:~$ less /var/log/syslog
```



```
2023-09-24T00:00:10.501671-04:00 ubnetdef35 rsyslogd: [origin software="rsyslogd
" swVersion="8.2302.0" x-pid="902" x-info="https://www.rsyslog.com"] rsyslogd wa
s HUPed
2023-09-24T00:00:10.529082-04:00 ubnetdef35 systemd[1]: logrotate.service: Deact
ivated successfully.
2023-09-24T00:00:10.529954-04:00 ubnetdef35 systemd[1]: Finished logrotate.servi
ce - Rotate log files.
2023-09-24T00:17:01.845781-04:00 ubnetdef35 CRON[4572]: (root) CMD (cd / && run-
parts --report /etc/cron.hourly)
2023-09-24T00:19:44.401676-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 185.125.190.56:123 (ntp.ubuntu.com).
2023-09-24T00:19:54.651433-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 185.125.190.57:123 (ntp.ubuntu.com).
2023-09-24T00:20:04.901073-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 185.125.190.58:123 (ntp.ubuntu.com).
2023-09-24T00:20:15.151124-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 91.189.91.157:123 (ntp.ubuntu.com).
2023-09-24T00:54:40.652682-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 185.125.190.56:123 (ntp.ubuntu.com).
2023-09-24T00:54:50.901032-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 91.189.91.157:123 (ntp.ubuntu.com).
2023-09-24T00:55:01.151076-04:00 ubnetdef35 systemd-timesyncd[592]: Timed out wa
iting for reply from 185.125.190.58:123 (ntp.ubuntu.com).
/var/log/syslog
```

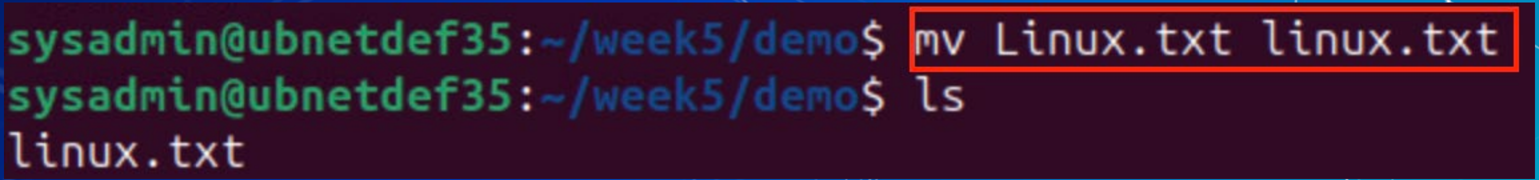
Interacting with files

- cp: Copy
  - Syntax: `cp </path/to/source> </path/to/destination>`



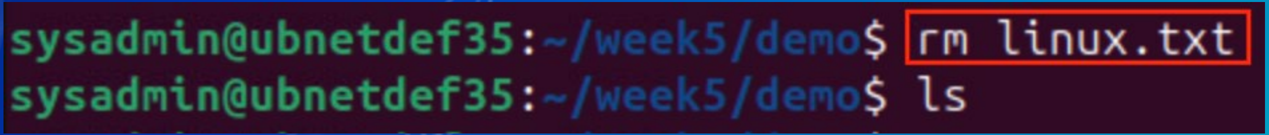
```
sysadmin@ubnetdef35:~$ cp /home/sysadmin/week5/Linux.txt /home/sysadmin/week5/demo
```

- mv: Move
  - Syntax: `mv </path/to/source> </path/to/destination>`
  - You can use this to rename files as well



```
sysadmin@ubnetdef35:~/week5/demo$ mv Linux.txt linux.txt
sysadmin@ubnetdef35:~/week5/demo$ ls
linux.txt
```

- rm: remove
  - Syntax: `rm <filename>`
  - Deletes the file for good. No recovery.



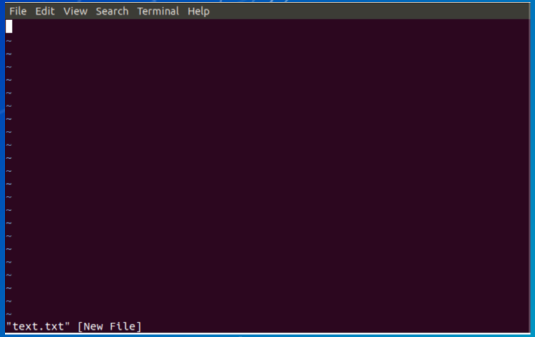
```
sysadmin@ubnetdef35:~/week5/demo$ rm linux.txt
sysadmin@ubnetdef35:~/week5/demo$ ls
```

Text Editors

- Syntax is <text editor name> <file> for anything

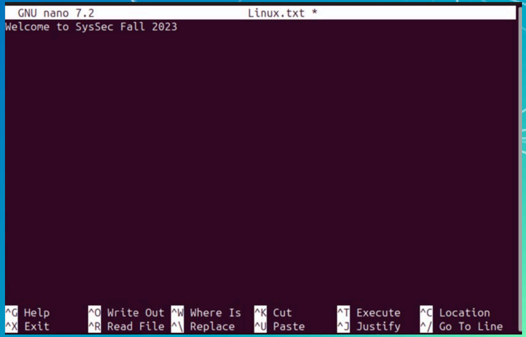
Editors

- `vim` - Very powerful editor with an unconventional workflow, can be hard for beginners
  - There are many good tutorials
  - Often times the default text editor
- `nano` - Pretty standard text editor, easier to use
  - Arrow keys to move and you can type, **ctrl + x** to exit and save
- `emacs` / `gedit` - Use the built in GUI text editor
  - Just like good ol' notepad
  - Emacs does have a CLI interface



```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"text.txt" [New File]
```

vim



```
  GNU nano 7.2                         Linux.txt *
Welcome to SysSec Fall 2023

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

nano

`find`

- `Find` is very powerful, useful, and complex for finding files
- Basic syntax:
  - `find <search directory> <options>`
  - `-name <name>` or `-iname <name>` (case insensitive)
    - supports wildcards such as “hello*” which might match “hello_world.txt”

`grep`

- `grep` is also a really powerful tool for searching inside files
  - `grep <pattern/word> <file>`
- It uses the power of regular expressions (regex) to do its magic
- Find text in large files
    - Log files...?
  - Filter unwanted text away
  - You can send output of other commands to it!

In Class Activity

Linux CTF

Activity – Linux CTF

- You have a vm named `LinuxCTF` with hidden files on it.
- `Username: ctfuser Password: ctfuser`
- `Open web browser and go to linuxctf.org:8000`
  - `The port is required!`
- `Username: Team## Password: Team##`
- `Use the commands we learned to find all the flags.`

Break

10 Minutes

Agenda

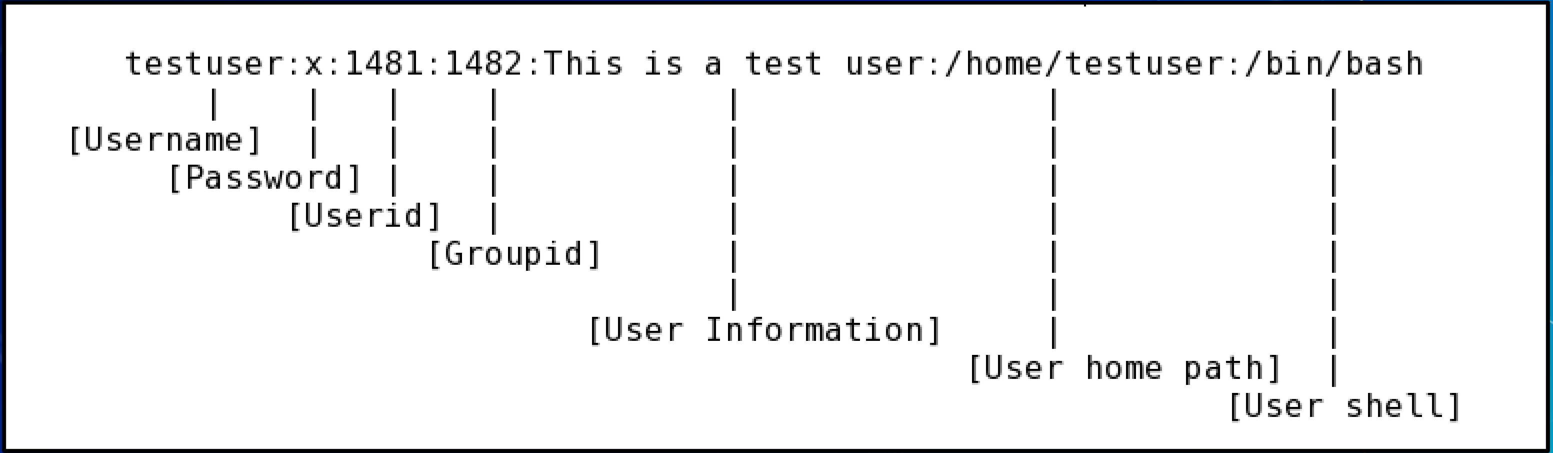
- Linux Basics
  - What is Linux? What is Kernal? What is Linux Distribution
  - Terminal
  - Commands– What Am I ? & Get Help!
- File System
  - Navigate File System
  - Interact with Files
  - Text Editors
- In class Activity – Linux CTF
- Users & Groups
- File Permission
- Others

Users and Groups

Users and Groups

- Linux systems have many users
  - One user per service
  - Stored in /etc/passwd
- Linux systems also have groups
  - Stored in /etc/group
- Every user has a User Identification number (UID)
- Groups also have unique Group Identification numbers (GIDs)
- The root user has a UID of 0
  - Root can do anything

/etc/passwd

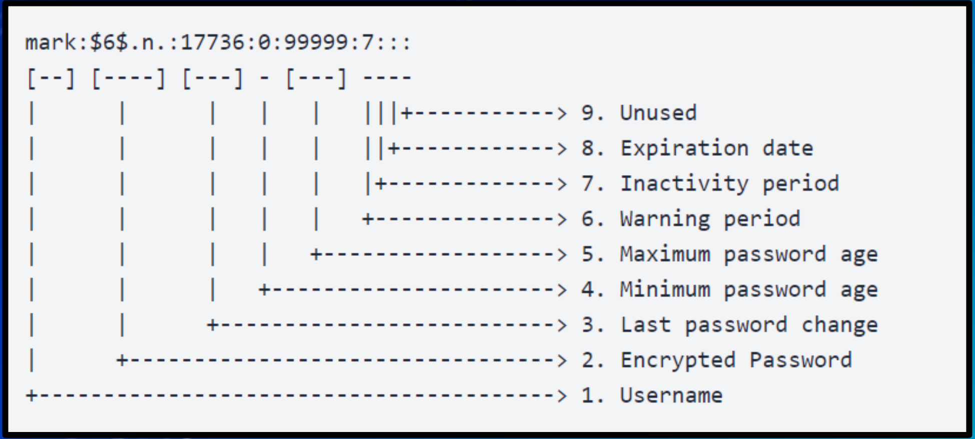


```
   testuser:x:1481:1482:This is a test user:/home/testuser:/bin/bash
       |    |   |    |           |               |             |
[Username]  |   |    |           |               |             |
     [Password] |    |           |               |             |
           [Userid]  |           |               |             |
                  [Groupid]      |               |             |
                                 |               |             |
                          [User Information]     |             |
                                             [User home path]  |
                                                          [User shell]
```

- Notice the x instead of the password?

/etc/shadow

- Encrypted passwords formally stored in /etc/passwd
- Now stored in /etc/shadow which is only readable by root



```
mark:$6$.n.:17736:0:99999:7:::
[--] [----] [---] - [---] ----
|      |      |   |   |   |||+------------> 9. Unused
|      |      |   |   |   ||+-------------> 8. Expiration date
|      |      |   |   |   |+--------------> 7. Inactivity period
|      |      |   |   |   +---------------> 6. Warning period
|      |      |   |   +-------------------> 5. Maximum password age
|      |      |   +-----------------------> 4. Minimum password age
|      |      +---------------------------> 3. Last password change
|      +----------------------------------> 2. Encrypted Password
+-----------------------------------------> 1. Username
```

Adding users

- `useradd`: Add a user to the system
  - Syntax: `useradd -c “<comment>” -m (create homdir) -s <shell> -g <primary group> -G <other groups> <username>`
  - Need to create password with passwd <username>
  - This is complicated and sucky

- `adduser` is interactive!
  - It is a wrapper around useradd
  - Handles creating the home directory, shell, password, etc
  - Not available on all systems
  - Syntax: `adduser <username>`

`userdel` and `deluser`

- `userdel` and `deluser` delete the user
- Like `useradd` and `adduser`, `deluser` is a wrapper around `userdel`

- Syntax: `deluser <username>`
  - The `-r` flag will also delete the user's home directory

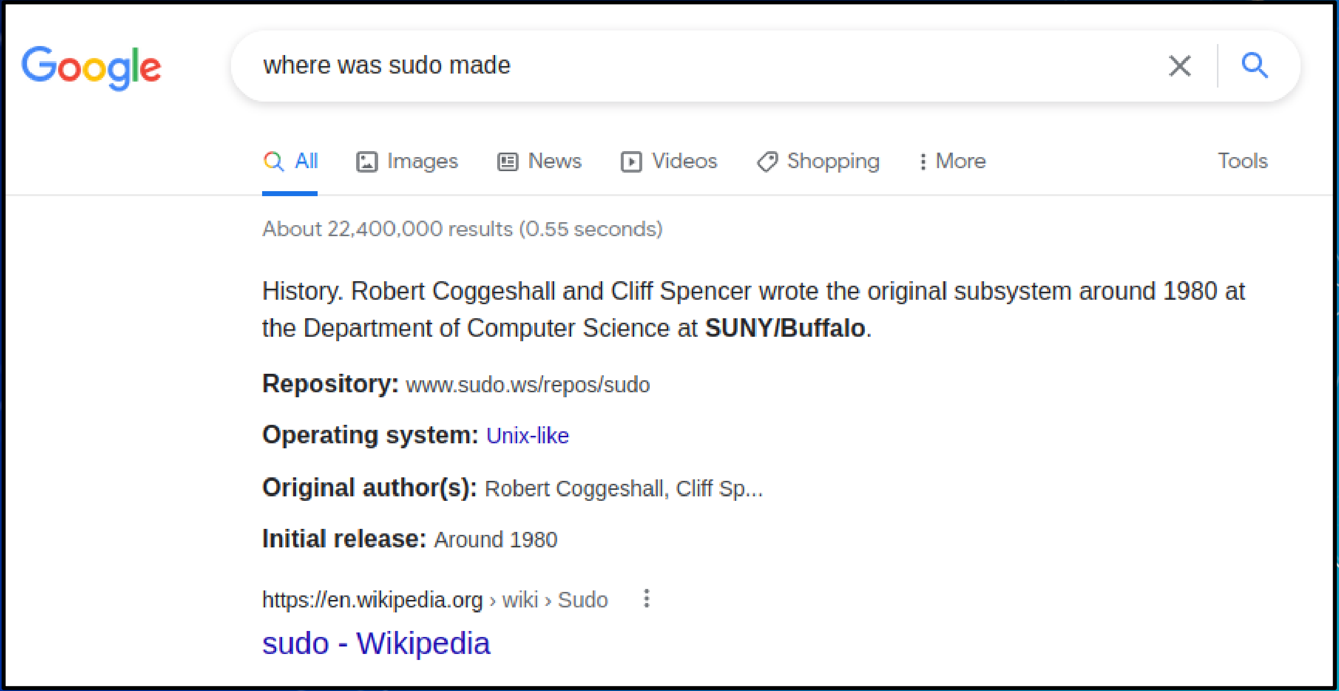
Administrative Right and Users

- The root user has full access to every part of the system
- Other users can access “root permissions” with the sudo command
- `sudo`: super user do
  - Syntax: `sudo <command>`
  - This will run the command with sudo permissions
  - To use `sudo` you must be in the sudo group
- Limit others users sudo access by editing the sudoers file
  - This is a special file, and must be edited with the `visudo` command

Administrative Right and Users

- You can switch users with `su`
- `su`: switch user
  - Syntax: `su <username>`
  - Typing `su` without a username will switch you into the root user

Fun fact about sudo:



where was sudo made

About 22,400,000 results (0.55 seconds)

History. Robert Coggeshall and Cliff Spencer wrote the original subsystem around 1980 at the Department of Computer Science at **SUNY/Buffalo**.

Repository: www.sudo.ws/repos/sudo

Operating system: Unix-like

Original author(s): Robert Coggeshall, Cliff Sp...

Initial release: Around 1980

https://en.wikipedia.org › wiki › Sudo

sudo - Wikipedia

Groups!

- Group name
- Password (usually unused)
- GID (Group ID)
- List of accounts which belong to the group
- All groups found in `/etc/group`
- Like security groups in Windows, Linux groups can also be used to grant users different privileges.

Fun with groups!

- `groupadd` and `groupdel` add/delete groups
  - Syntax: `groupadd <group name>`
  - Syntax: `groupdel <group name>`
- `usermod` lets you add/remove users to a group
  - Syntax: `usermod -G <Group> <username>`
- `getent` will let you see which users are part of a group
  - Syntax: `getent group <groupname>`

Agenda

- Linux Basics
  - What is Linux? What is Kernal? What is Linux Distribution
  - Terminal
  - Commands– What Am I ? & Get Help!
- File System
  - Navigate File System
  - Interact with Files
  - Text Editors
- In class Activity – Linux CTF
- Users & Groups
- File Permission
- Others

Let’s talk (file) permissions

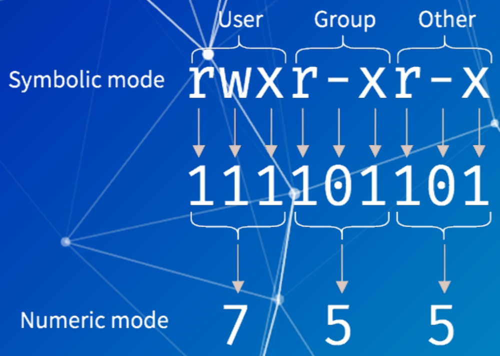
Permission Types

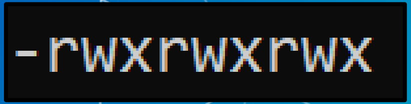
Each file/folder has 3 types of permission types

- **Read** – The Read permission refers to a user’s capability to read the contents of the file.
- **Write** – The Write permissions refer to a user’s capability to write or modify a file or directory.
- **Execute** – The Execute permission affects a user’s capability to execute a file or view the contents of a directory.

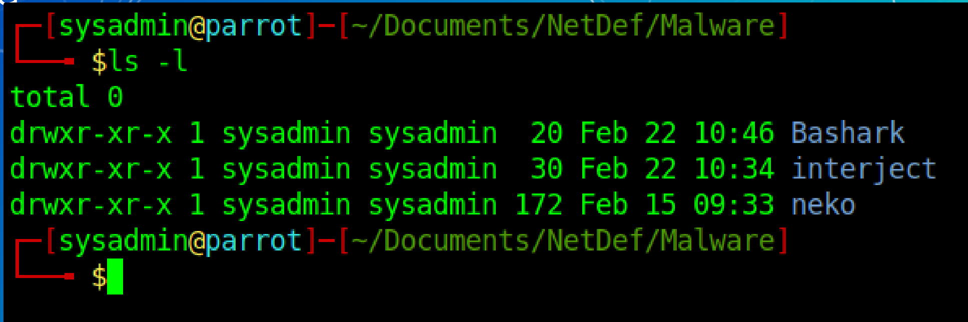
File permissions

- Files owned by user and group
- File modes are read/write/execute
- Mode permissions granted to
  - owner, owning group, everyone
- Modifying
  - See permissions with `ls -al` command
  - Set modes with `chmod` command
  - Set owners with `chown` command



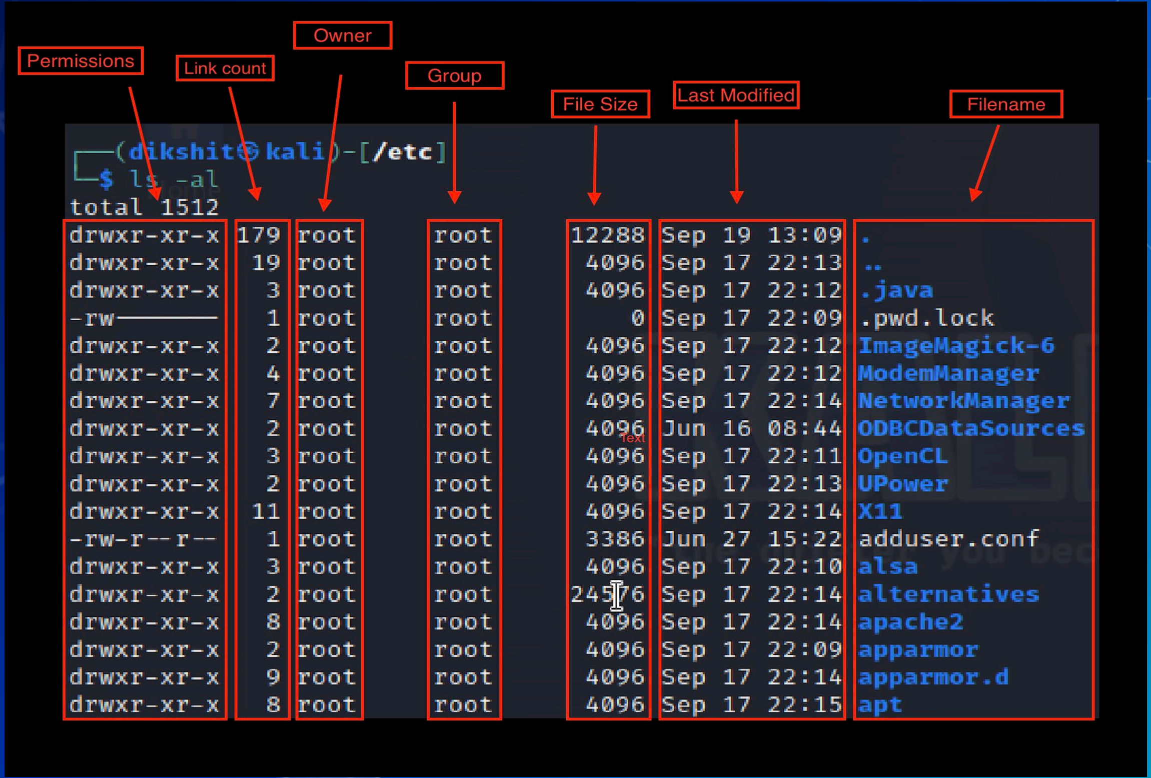


```
-rwxrwxrwx
```

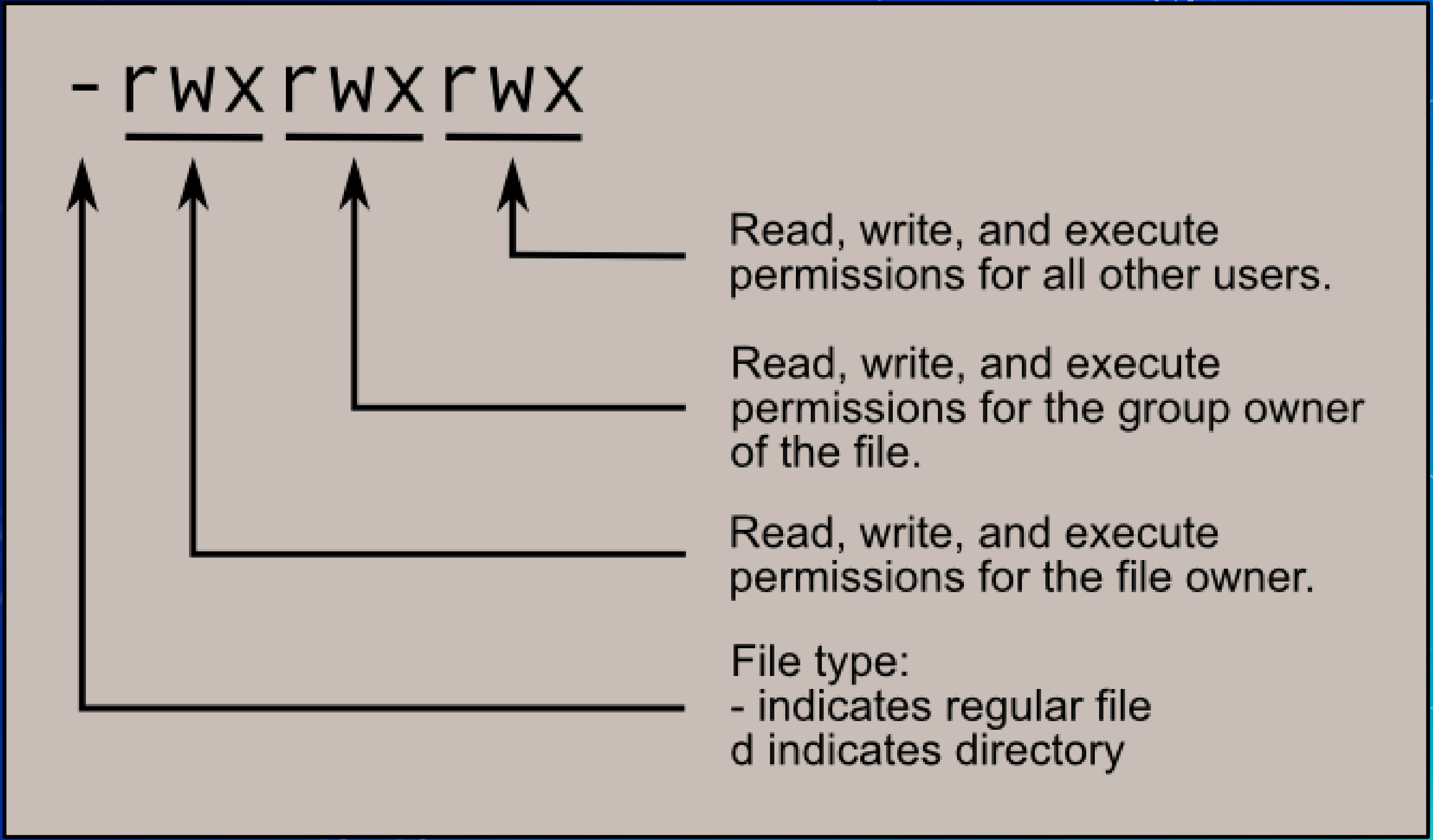


```
┌─[sysadmin@parrot]─[~/Documents/NetDef/Malware]
└──╼ $ls -l
total 0
drwxr-xr-x 1 sysadmin sysadmin  20 Feb 22 10:46 Bashark
drwxr-xr-x 1 sysadmin sysadmin  30 Feb 22 10:34 interject
drwxr-xr-x 1 sysadmin sysadmin 172 Feb 15 09:33 neko
┌─[sysadmin@parrot]─[~/Documents/NetDef/Malware]
└──╼ $
```

See permissions using **ls -al**



```
┌──(dikshit㉿kali)-[/etc]
└─$ ls -al
total 1512
drwxr-xr-x 179 root      root       12288 Sep 19 13:09 .
drwxr-xr-x  19 root      root        4096 Sep 17 22:13 ..
drwxr-xr-x   3 root      root        4096 Sep 17 22:12 .java
-rw-------   1 root      root           0 Sep 17 22:09 .pwd.lock
drwxr-xr-x   2 root      root        4096 Sep 17 22:12 ImageMagick-6
drwxr-xr-x   4 root      root        4096 Sep 17 22:12 ModemManager
drwxr-xr-x   7 root      root        4096 Sep 17 22:14 NetworkManager
drwxr-xr-x   2 root      root        4096 Jun 16 08:44 ODBCDataSources
drwxr-xr-x   3 root      root        4096 Sep 17 22:11 OpenCL
drwxr-xr-x   2 root      root        4096 Sep 17 22:13 UPower
drwxr-xr-x  11 root      root        4096 Sep 17 22:14 X11
-rw-r--r--   1 root      root        3386 Jun 27 15:22 adduser.conf
drwxr-xr-x   3 root      root        4096 Sep 17 22:10 alsa
drwxr-xr-x   2 root      root       24576 Sep 17 22:14 alternatives
drwxr-xr-x   8 root      root        4096 Sep 17 22:14 apache2
drwxr-xr-x   2 root      root        4096 Sep 17 22:09 apparmor
drwxr-xr-x   9 root      root        4096 Sep 17 22:14 apparmor.d
drwxr-xr-x   8 root      root        4096 Sep 17 22:15 apt
```



-rwxrwxrwx

Read, write, and execute permissions for all other users.

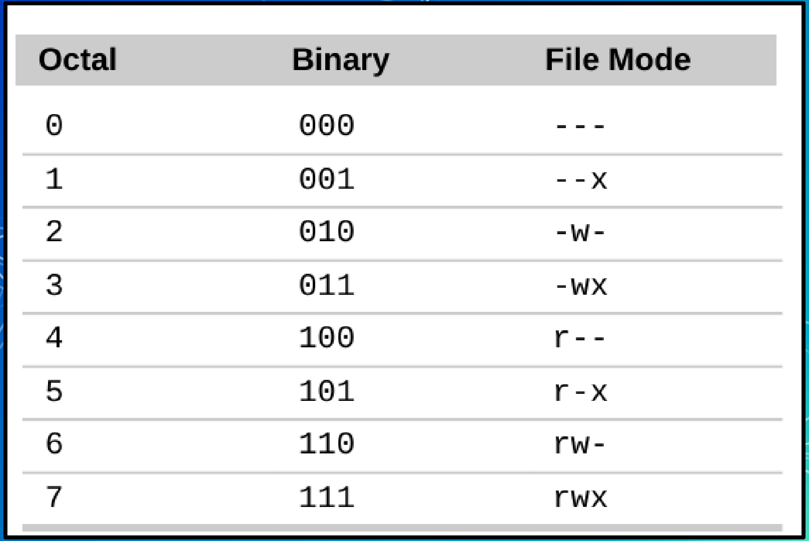
Read, write, and execute permissions for the group owner of the file.

Read, write, and execute permissions for the file owner.

File type:
- indicates regular file
d indicates directory

Reading a Permission Entry

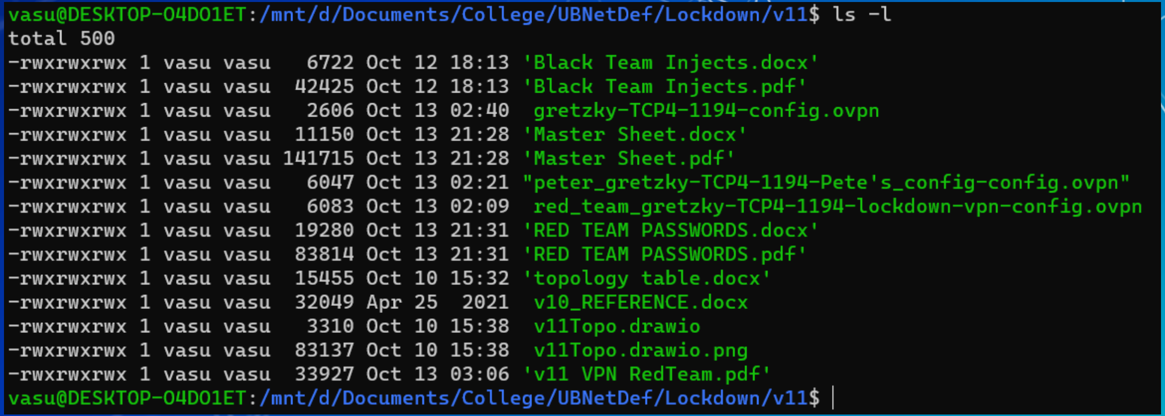
- <type flag> <owner permissions> <group permissions> <world permissions>
- Default permissions = 644
  - ○ Read and write for owner
  - ○ Read for group and the world.
- What is 755?
- What about 245?



Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

chmod

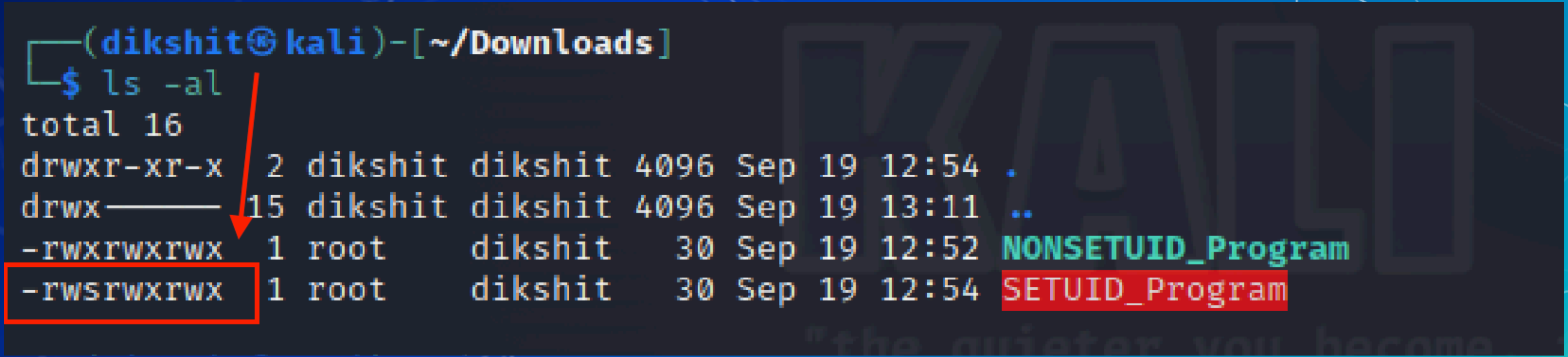
- chmod = change file mode bits
- change file permissions
- `chmod <permission> <filename>`
  - Allow a file to be executable: `chmod +x myFile`
  - Grant all permissions to a file: `chmod 777 myFile`



```
vasu@DESKTOP-O4DO1ET:/mnt/d/Documents/College/UBNetDef/Lockdown/v11$ ls -l
total 500
-rwxrwxrwx 1 vasu vasu   6722 Oct 12 18:13 'Black Team Injects.docx'
-rwxrwxrwx 1 vasu vasu  42425 Oct 12 18:13 'Black Team Injects.pdf'
-rwxrwxrwx 1 vasu vasu   2606 Oct 13 02:40  gretzky-TCP4-1194-config.ovpn
-rwxrwxrwx 1 vasu vasu  11150 Oct 13 21:28 'Master Sheet.docx'
-rwxrwxrwx 1 vasu vasu 141715 Oct 13 21:28 'Master Sheet.pdf'
-rwxrwxrwx 1 vasu vasu   6047 Oct 13 02:21 "peter_gretzky-TCP4-1194-Pete's_config-config.ovpn"
-rwxrwxrwx 1 vasu vasu   6083 Oct 13 02:09  red_team_gretzky-TCP4-1194-lockdown-vpn-config.ovpn
-rwxrwxrwx 1 vasu vasu  19280 Oct 13 21:31 'RED TEAM PASSWORDS.docx'
-rwxrwxrwx 1 vasu vasu  83814 Oct 13 21:31 'RED TEAM PASSWORDS.pdf'
-rwxrwxrwx 1 vasu vasu  15455 Oct 10 15:32 'topology table.docx'
-rwxrwxrwx 1 vasu vasu  32049 Apr 25  2021  v10_REFERENCE.docx
-rwxrwxrwx 1 vasu vasu   3310 Oct 10 15:38  v11Topo.drawio
-rwxrwxrwx 1 vasu vasu  83137 Oct 10 15:38  v11Topo.drawio.png
-rwxrwxrwx 1 vasu vasu  33927 Oct 13 03:06 'v11 VPN RedTeam.pdf'
vasu@DESKTOP-O4DO1ET:/mnt/d/Documents/College/UBNetDef/Lockdown/v11$ |
```

Set-UID Program

- A bit that makes an executable run with the privileges of the owner of the file



```
┌──(dikshit㉿kali)-[~/Downloads]
└─$ ls -al
total 16
drwxr-xr-x  2 dikshit dikshit 4096 Sep 19 12:54 .
drwx------ 15 dikshit dikshit 4096 Sep 19 13:11 ..
-rwxrwxrwx  1 root    dikshit   30 Sep 19 12:52 NONSETUID_Program
-rwsrwxrwx  1 root    dikshit   30 Sep 19 12:54 SETUID_Program
```

User IDs (UIDS) in Linux

Each Linux/Unix process has 3 UIDs associated with it.

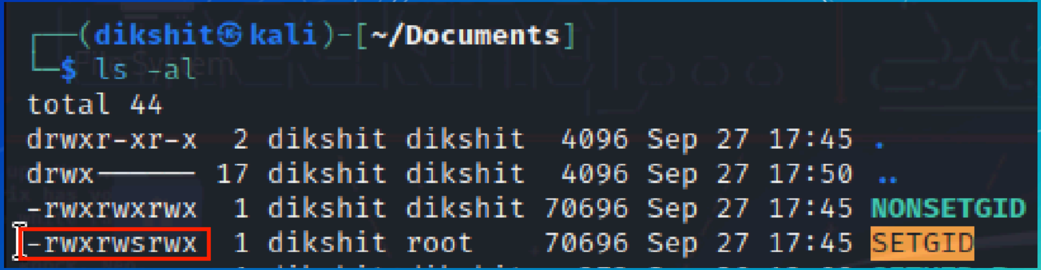
- **Real UID (RUID)**: This is the UID of the user/process that created the process.
- **Effective UID (EUID)**: This UID is used to evaluate privileges of the process to perform a particular action.
  - ○ EUID can be changed either to RUID, or SUID
  - ○ For Set-UID, EUID will equal to RUID
  - ○ For NonSet-UID, EUID will be equal to user ID of root
- Saved UID (SUID): If the binary image file, that was launched has a Set-UID bit on, SUID will be the UID of the owner of the file. Otherwise, SUID will be the RUID.

Note: Set-UID is not equal to Saved-UID

Set-GID

- Similar to SETUID, but instead of taking on the user ID of the owner, the executing program assumes the group ID of the file.

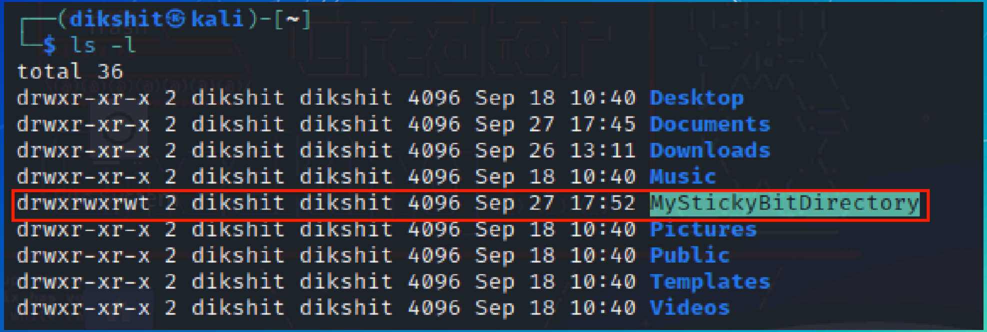
- This is often used on directories. When SETGID is set on a directory, files created within that directory inherit the group ownership of the directory, not the primary group of the creating user.



```
┌──(dikshit㉿kali)-[~/Documents]
└─$ ls -al
total 44
drwxr-xr-x  2 dikshit dikshit  4096 Sep 27 17:45 .
drwx------ 17 dikshit dikshit  4096 Sep 27 17:50 ..
-rwxrwxrwx  1 dikshit dikshit 70696 Sep 27 17:45 NONSETGID
-rwxrwsrwx  1 dikshit root    70696 Sep 27 17:45 SETGID
```

Sticky Bit

- The sticky bit is a permission bit that protects the files within a directory
- When the sticky bit is set on a directory, only the file's owner, the directory's owner, or the root user can delete or rename files in that directory, regardless of the file's permissions.



```
┌──(dikshit㉿kali)-[~]
└─$ ls -l
total 36
drwxr-xr-x 2 dikshit dikshit 4096 Sep 18 10:40 Desktop
drwxr-xr-x 2 dikshit dikshit 4096 Sep 27 17:45 Documents
drwxr-xr-x 2 dikshit dikshit 4096 Sep 26 13:11 Downloads
drwxr-xr-x 2 dikshit dikshit 4096 Sep 18 10:40 Music
drwxrwxrwt 2 dikshit dikshit 4096 Sep 27 17:52 MyStickyBitDirectory
drwxr-xr-x 2 dikshit dikshit 4096 Sep 18 10:40 Pictures
drwxr-xr-x 2 dikshit dikshit 4096 Sep 18 10:40 Public
drwxr-xr-x 2 dikshit dikshit 4096 Sep 18 10:40 Templates
drwxr-xr-x 2 dikshit dikshit 4096 Sep 18 10:40 Videos
```

Questions?

Agenda

- Linux Basics
  - What is Linux? What is Kernal? What is Linux Distribution
  - Terminal
  - Commands
- File System
  - Navigate File System
  - Interact with Files
- In class Activity – Linux CTF
- File Permission
- Users & Groups
- Others

Package managers

- Used to install, uninstall, update and upgrade packages.
- Each distro has its own version
  - `apt` - Ubuntu, and Debian based
  - `yum` - CentOS and other Red Hat Enterprise
- To install a new package:
  - `sudo <package manager> install <package name>`

Update != Upgrade

- Update does not update your system!
  - It updates sources which keep track of new packages
- Upgrades actually downloads the new stuff
- Run update before upgrade

Remote connections (ssh)

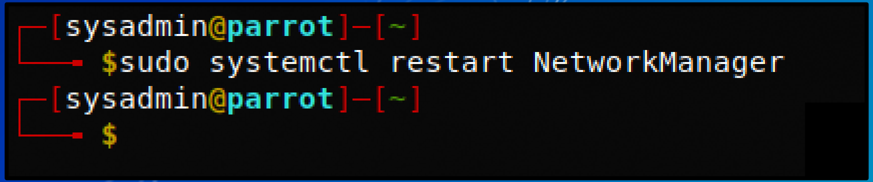
- SSH is the most popular way of accessing and managing Linux systems remotely
- Usage: `ssh username@remote-host`
    - E.g., `ssh vasu@133.76.94.20`

- SSH can use public/private keys instead of/in conjunction with password based authentication
- Check out ssh-keygen and the man pages/google

Secure Shell Protocol

Services

- Services on Linux on are managed by the `systemd` service
  - Not all distros use systemd, but the major ones do
- `systemctl <command> <service name>`
  - status
  - enable
  - start/stop
- When have you used `systemctl` before?



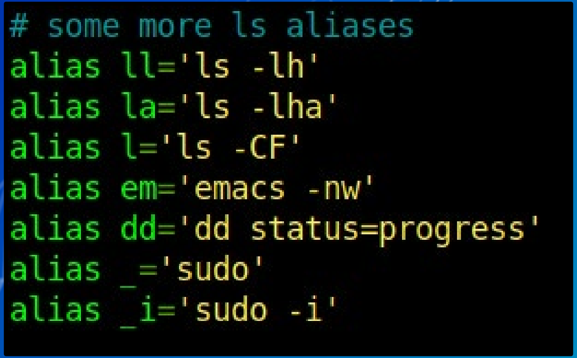
```
┌─[sysadmin@parrot]─[~]
└──╼ $sudo systemctl restart NetworkManager
┌─[sysadmin@parrot]─[~]
└──╼ $
```

Environment variables

- Environment variables are a way to store information in a shell
- They can be set for the duration of a shell session with the `export` command
  - Syntax: `NEW_ENV=something`
  - Syntax: `export NEW_ENV=something`
- Environment variables can be put in shell configs and run every time a shell starts
- You can check the value of an environment variable with the echo command
  - `echo $NEW_ENV` would return “something”

Aliases

- Aliases are a great way to reduce repetitive and/or long commands
  - Because who doesn’t like being lazy?
- The syntax is easy: `alias word=’long command’`
  - Example: `alias errorlog=’cat /var/log/system.log | grep error’`
- To see a list of all currently set aliases, just type `alias`
- To unset an alias, type `unalias <X>` where `<X>` is the alias you want to unset



```
# some more ls aliases
alias ll='ls -lh'
alias la='ls -lha'
alias l='ls -CF'
alias em='emacs -nw'
alias dd='dd status=progress'
alias _='sudo'
alias _i='sudo -i'
```

Pipes and redirecting things

- Redirect output to files
  - `command > outputfile.txt` (This will overwrite the file)
  - `command >> outfile.txt` (This will append to the file)
- Input file contents
  - `command < inputfile.txt`
- Pipe
  - `command | command2`
    - `cat log.txt | grep “success” | less`

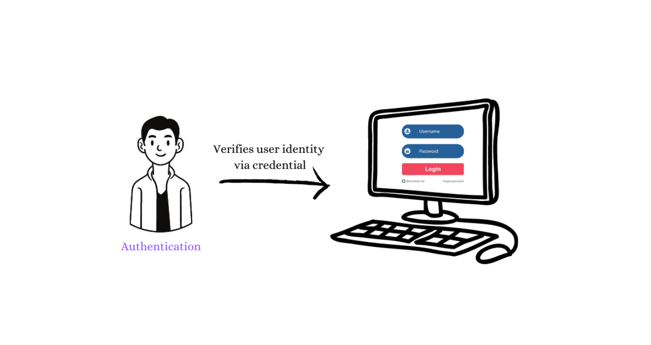
Previous Commands

- `history` : Show your history on shells that keep track
  - `history -c` to clear your history
- `Ctrl + R` : Search command history
- `!!` : Rerun previous command
- `sudo !!` : Rerun as superuser (you will do this a lot)
- <Up Arrow> : Cycle through previous commands

What is Authentication?

- Authentication is the process of verifying the identity of a user, system, or application.

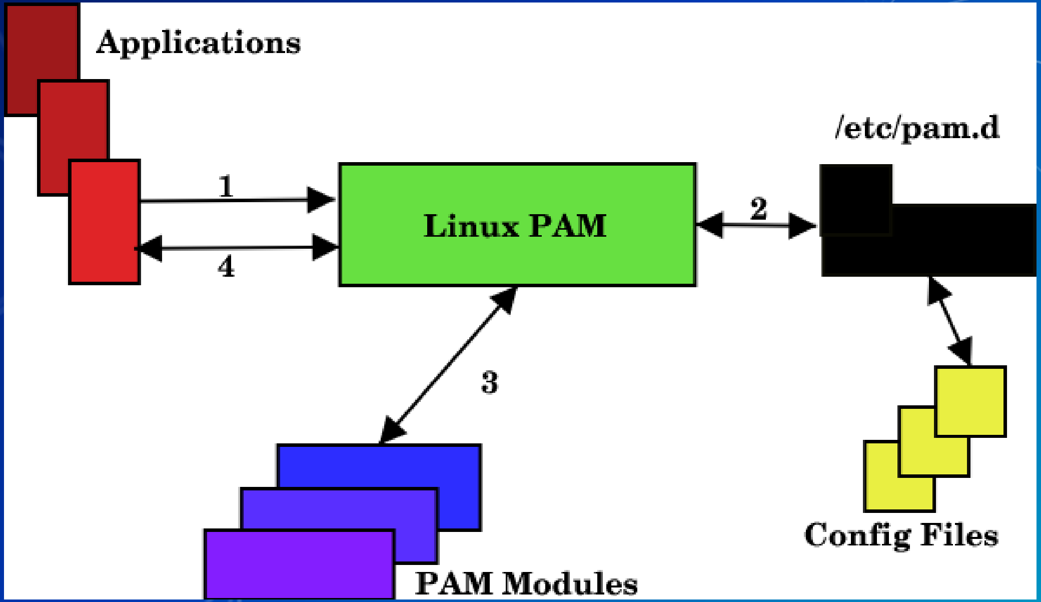
- It's essentially answering the question: "Are you who you say you are?"



PAM (Pluggable Authentication Modules)

- PAM, or Pluggable Authentication Modules, is a framework used on Unix-based systems, including Linux, to manage authentication.
- There are four primary management groups, commonly referred to as "module types".
  - auth:
    - It deals with user credentials and can be responsible for setting up user authentication tokens. Essentially, it's about proving and verifying who you claim to be.
  - account:
    - This checks if the user is allowed to get access at this specific time, from this specific terminal, to this specific service, etc.
  - password:
    - This module type is concerned with password management. It deals with updating passwords
  - session:
    - This is about setting up and tearing down sessions. It can handle tasks that need to be done at the beginning or end of sessions, like logging, mounting directories, or setting quotas.

Understanding PAM using Diagram



In Class Activity

Linux CTF 2

Summary

Today we:

- Learned about the Linux filesystem.
- Reviewed several commands for Linux administration.
- Used tools like `man` pages to understand command syntax.

OverTheWire: Bandit

Another Linux CTF centered around basic to advanced command usage.

https://overthewire.org/wargames/bandit/

If you want to talk more about Linux, just message me, or swing by my OH

That’s all folks