NDG Linux Unhatched

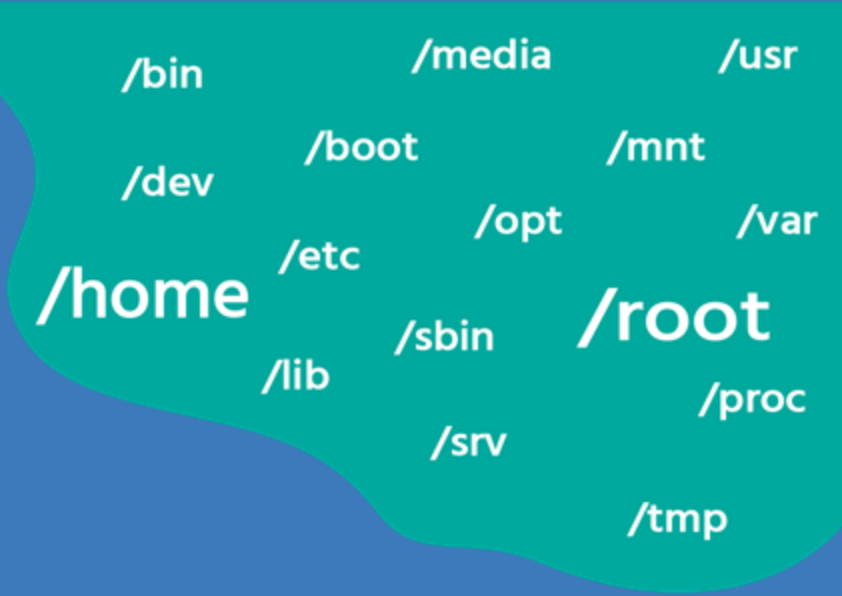
Why Learn Linux?

- Linux is everywhere!
    - Linux is used by web servers, personal computers, mobile devices, and more.
- Linux is operating system software that is found in many areas of IT (i.e, cloud technology, virtualization, cybersecurity, networking…).
- Learning Linux results in acquiring in-demand technical job skills to analyze, process, protect, and transmit data.

Why Learn Linux?

- Linux emphasizes the use of a powerful tool called the *command line*.
- Linux powers the Internet. How? A high percentage of Internet servers and services use Linux.
- Linux is an open source product. Open source technology skills are in high-demand in today's job market.

Navigating Basic Command Syntax



Learn how to type commands

NDG

Basic Linux Command Syntax

- What is a command?
    - A command is a software program that can be executed on the command line and performs an action on the computer.
- To execute a command, type the command at the prompt:

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

(In the example above, the command `ls` was typed at the prompt `sysadmin@localhost:~$`.)

- Most commands follow a simple pattern of syntax:

```
command [options…] [arguments…]
```

Basic Linux Command Syntax

- What are *arguments*?

```
command [options…] [arguments…]
```

- An argument can be used to specify something for the command to act upon.

- For example, the `ls` command can be given the *name of a directory* as an argument and it will list the contents of that directory:

```
sysadmin@localhost:~$ ls Documents
School          alpha-second.txt  food.txt     linux.txt     os.csv
Work            alpha-third.txt   hello.sh     longfile.txt  people.csv
...
```

Basic Linux Command Syntax

- What are *options*?

```
command [options…] [arguments…]
```

- Options can be used to alter the behavior of a command.

- For example, the `-l` option can be used with the `ls` command to result in a "long display" output that provides more information about the directories:

```
sysadmin@localhost:~$ ls -l
total 32
drwx------ 2 sysadmin sysadmin 4096 Dec 20  2017 Desktop
drwx------ 4 sysadmin sysadmin 4096 Dec 20  2017 Documents
drwx------ 2 sysadmin sysadmin 4096 Dec 20  2017 Downloads
...
```

Printing The Working Directory

How to locate where you are in the filesystem

NDG

Printing the Working Directory

- To find out where you are in the Linux filesystem, use the `pwd` (print working directory) command.

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

- According to the output of the `pwd` command above, the user is in their `home` folder in the filesystem.

Note

In the first prompt above, the blue ~ is equivalent to `/home/sysadmin`, representing the user's home directory. After changing directories, the new location can also be confirmed in the new prompt, again shown in blue.

```
sysadmin@localhost:/Documents$
```

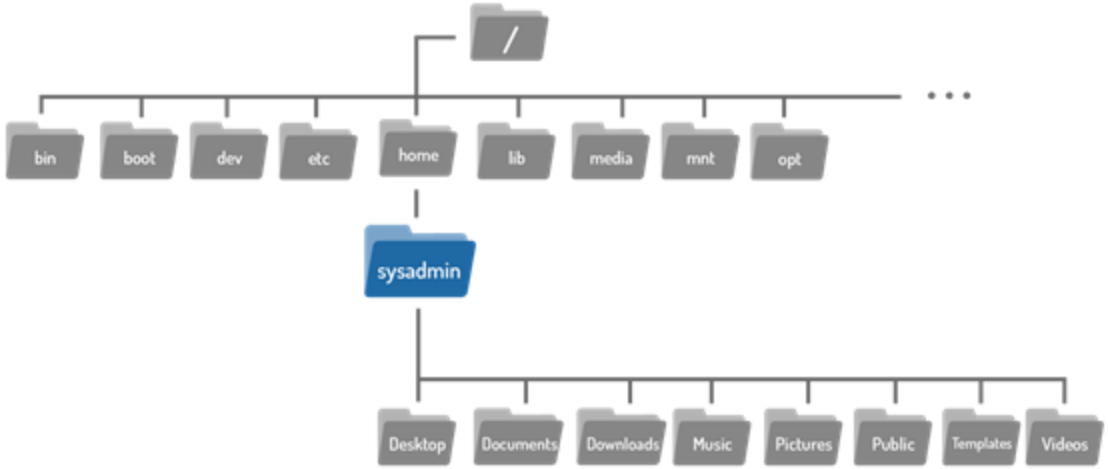
Changing Directories

How to move to another directory in the filesystem

NDG

Changing Directories

- Directories are a type of file used to store other files.
- They provide a hierarchical structure:



Changing Directories

- To navigate the filesystem structure, use the `cd` command. Using a directory as an argument will change to that directory:

```
sysadmin@localhost:~$ cd Documents
sysadmin@localhost:/Documents$
```

- A *path* is a list of directories separated by the `/` character (i.e., `/home/sysadmin`).
- *Absolute paths* specify the exact location of a directory beginning at the root (`/`) directory: `/home/sysadmin`
- *Relative paths* give directions to a file relative to your current location (instead of the root directory): `School/Art`

Changing Directories

- Shortcuts can also be used to change directories.
    - The `..` characters always represents one (1) directory higher relative to your current directory, also known as the *parent* directory.

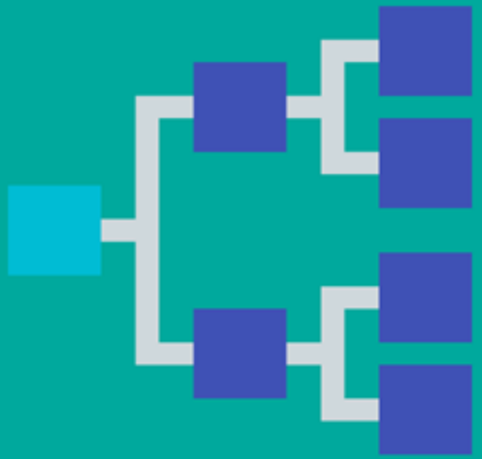
```
sysadmin@localhost:/Documents/School/Art$ cd ..
sysadmin@localhost:/Documents/School$
```

    - The `.` character represents your current directory.
    - The `~` character represents the home directory of the current user.

```
sysadmin@localhost:/Documents/School/Art$ cd ~
sysadmin@localhost:~$
```

Get Started

Listing Files



Learn how to list files and file information

Listing Files

- The `ls` command is used to list the contents of a directory.
- Remember that when used with the `-l` option, the `ls` command will display more details about files:

```
sysadmin@localhost:~$ ls -l /var/log
total 844
-rw-r--r-- 1 root   root  18047 Dec 20  2017 alternatives.log
drwxr-x--- 2 root   adm    4096 Dec 20  2017 apache2
drwxr-xr-x 1 root   root   4096 Dec 20  2017 apt
```

- The output includes information about file type, file permissions, file ownership, timestamps and more.

Listing Files

- The information displayed in the `ls -l` output can be broken down into different fields:
  - **File Type**:

```
-rw-r--r-- 1 root   root  18047 Dec 20  2017 alternatives.log
drwxr-x--- 2 root   adm    4096 Dec 20  2017 apache2
```

    - The first field has ten characters. The first character indicates the file type.
  - **Permissions**:

```
drwxr-x--- 2 root   adm    4096 Dec 20  2017 apache2
```

    - Permissions indicate how users can access a file.

Listing Files

- **File Size**:

```
-rw-r--r-- 1 root   root  18047 Dec 20  2017 alternatives.log
```

  - Size of files in bytes.

- **Timestamp**:

```
drwxr-x--- 2 root   adm    4096 Dec 20  2017 apache2
```

  - Indicates when a file's contents were last modified.

- **File Name**:

```
drwxr-x--- 2 root   adm    4096 Dec 20  2017 apache2
```

  - The name of the file or directory.

Administrative Access

How to gain administrative access using the Linux command line

Administrative Access

- Some commands provide sensitive information such as passwords and system information.

- Preventing regular users from executing these commands helps protect the system.

- Logging in as the `root` user provides administrative access to execute these commands.

Administrative Access

- The `su` command allows you to temporarily login as a different user by opening a new shell.

- `su` can be used to login as the `root` user, providing administrative privileges.

- To login as the `root` user, execute the `su` command without specifying a user and type in the *root* password:

```
sysadmin@localhost:~$ su -
Password:
root@localhost:~$
```

- To return to the regular user, use the `exit` command

Administrative Access

- The `sudo` command allows a user to execute a command as another user without creating a new shell.

- To execute a command as `root`, type `sudo` before the command and provide the *root* password:

```
sysadmin@localhost:~$ sudo sl
Password:
sysadmin@localhost:~$
```

Permissions

Learn about user permissions for files and directories

Permissions

- Permissions determine the ways users can interact a file or directory.

- When listing a file with the `ls -l` command, the output includes permissions.

- Recall that the permissions are in the first field after the file type character:

```
-rw-rw-r-- 1 sysadmin   sysadmin    21 Aug 1  02:35 hello.sh
```

Permissions

- Permissions are broken down into three sets of three characters.
  - **Owner**:

```
-rw-rw-r-- 1 sysadmin   sysadmin    21 Aug 1  02:35 hello.sh
```

  - **Group**:

```
-rw-rw-r-- 1 sysadmin   sysadmin    21 Aug 1  02:35 hello.sh
```

  - **Others**:

```
-rw-rw-r-- 1 sysadmin   sysadmin    21 Aug 1  02:35 hello.sh
```

Permissions

- There are three permission types:

Permission	Effect on File	Effect on Directory
Read (`r`)	Can read or copy file contents	Allows for listing of files
Write (`w`)	Can modify or overwrite file contents	Can modify directory contents (only with execute)
Execute (`x`)	Can run the file as a process	Can change (`cd`) to the directory if parent directory has execute also

Changing Permissions

How to change file and directory permissions

Changing Permissions

- The `chmod` command is used to change the permissions of a file or directory.
- There are two techniques to change permissions using `chmod` ; the *symbolic method* and the *octal method* (not covered in this course).
- Symbolic Method:

```
chmod [ <SET><ACTION><PERMISSIONS>]... FILE
```

  - First indicate the `<SET>` using the following symbols:
    - `u` - User permissions
    - `g` - Group permissions
    - `o` - Other permissions
    - `a` - All permissions

Changing Permissions

- Then specify the `<ACTION>` symbol:
  - `(+)` - Add the permission
  - `(=)` - Specify exact permission
  - `(-)` - Remove the permission
- Specify one or more `<PERMISSIONS>` to be acted upon:
  - `r` - Read
  - `w` - Write
  - `x` - Execute
- Lastly, specify the `FILE` or pathname to assign the permissions to. The action should resemble the following example:

```
sysadmin@localhost:/Documents$ chmod u+x hello.sh
```

How to change file and directory ownership

Changing Ownership

- The `chown` command is used to change the ownership of a file or directory.
- To change the user owner of a file, use the following syntax:

```
chown [OPTIONS] [OWNER] [FILE]
```

- For example, to change the owner of the hello.sh file from `sysadmin` to `root`:

```
sysadmin@localhost:/Documents$ ls -l hello.sh
-rw-rw-r-- 1 sysadmin sysadmin 112 Aug  1 02:35 hello.sh
sysadmin@localhost:/Documents$ sudo chown root hello.sh
sysadmin@localhost:/Documents$ ls -l hello.sh
-rw-rw-r-- 1 root sysadmin 112 Aug  1 02:35 hello.sh
```

Don't forget to use the `sudo` command in order to gain the necessary administrative privileges.

On Moving Files

How to move files around the filesystem

Moving Files

- The `mv` command is used to move a file from one location in the filesystem to another.
- The `mv` command requires at least two arguments; a source (file to be moved) and a destination (where the file will be moved):

```
mv SOURCE DESTINATION
```

- For example, to move the `people.csv` file to the `Work` directory:

```
sysadmin@localhost:/Documents$ mv people.csv Work
sysadmin@localhost:/Documents$ ls Work
people.csv
```

Copying Files

How to create copies of files in the filesystem

Copying Files

- The `cp` command is used to copy files.
- Similar to the `mv` command, `cp` requires at least two arguments; a source (file to be copied) and a destination (where the file will be copied):

```
cp SOURCE DESTINATION
```

- For example, to copy the `/etc/passwd` file to the current directory:

```
sysadmin@localhost:/Documents$ cp /etc/passwd .
```

Recall that `.` is a shortcut which represents the current directory.

Copying Files

- The `dd` command is a utility used for copying files or partitions.
- This command can be useful for a few reasons:
    - To clone or delete disk partitions
    - To copy raw data to removable devices (USB, CDROM...)
    - Backup your computer's MBR (Master Boot Record)
    - Create files with binary zeros used for virtual memory
- For example, using `dd` the following creates a file named `/tmp/swapex` with 50 blocks of zeros that are one megabyte in size:

```
sysadmin@localhost:~$ dd if=/dev/zero of=/tmp/swapex bs=1M count=50
```

    - `if` = input file; `of` = output file; `bs` = byte size

The process of Removing Files

How to remove files from the filesystem

Removing Files

- The `rm` command is used to delete files and directories.
- The `rm` command without any options is used to remove files:

```
sysadmin@localhost:/Documents$ rm linux.txt
sysadmin@localhost:/Documents$ ls linux.txt
ls: cannot access linux.txt: no such file or directory
```

- To delete a directory, use the `-r` or `-R` option with the `rm` command:

```
sysadmin@localhost:/Documents$ rm Work
rm: cannot remove 'Work': Is a directory
sysadmin@localhost:/Documents$ rm -r Work
```

Viewing Files

How to view file contents

Viewing Files

- The `cat` command can be used to view the entire contents of a file:

```
sysadmin@localhost:/Documents$ cat food.txt
Food is good.
```

- The `head` command displays the *first* ten lines of a file.

```
sysadmin@localhost:/Documents$ head alpha.txt
```

- The `tail` command displays the *last* ten lines of a file.

```
sysadmin@localhost:/Documents$ tail alpha.txt
```

- The `-n` option can be used with `head` and `tail` to adjust the number of line displayed.

Filtering Input, Regular Expressions And Basic Patterns

How to filter file contents using regular expressions and basic patterns

NDG

Filtering Input

- The `grep` command is a text filter that will search input and return matches of a pattern:

```
grep [OPTIONS] PATTERN [FILE]
```

- For example, to filter information about the `sysadmin` user in the `passwd` file, use `sysadmin` as the *`PATTERN`* argument and `passwd` as the *`[FILE]`* argument:

```
sysadmin@localhost:/Documents$ grep sysadmin passwd
sysadmin:x:1001:1001:System Administrator,,,,:/home/sysadmin:/bin/bash
```

Filtering Input

- Regular expressions (regex) are patterns that only certain commands can interpret.
- Regular expressions can also be used with the `grep` command to filter file contents.
    - Basic regular expressions include:

`.`	Any one single character	`*`	Zero or more of the previous character
`[ ]`	Any one specified character	`^`	Pattern must be at beginning, or literal ^
`[^ ]`	Not the specified character	`$`	Pattern must be at end, or literal $

    - Extended regular expressions include:

`+`	One or more of the previous pattern	`( )`	Used to create groups
`{ }`	Specify minimum, maximum, or exact matches of previous pattern	`|`	Alternation (logical “or”)

Filtering Input

- Regular expressions can be used with `grep` to match basic patterns in text.
- Anchor Characters:
    - First anchor character ^ is used to ensure that the pattern appears at the *beginning*.

```
sysadmin@localhost:~/Documents$ grep '^root' /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

    - Second anchor character $ is used to ensure that the pattern appears at the *end*.

```
sysadmin@localhost:~/Documents$ grep 'r$' alpha-first.txt
B is for Bear
F is for Flower
```

Filtering Input

- The `.` character will match *any* character (except newline)
    - For example, the pattern `r..f` will return the following:

```
sysadmin@localhost:~/Documents$ grep 'r..f' red.txt
reef
roof
```

- The `[ ]` brackets match a *single* character from a list or range contained in the brackets:

```
sysadmin@localhost:~/Documents$ grep '[0-9]' profile.txt
I am 37 years old.
3121991
I have 2 dogs.
123456789101112
```

Filtering Input

- The `*` character will match *zero or more* of a character preceding it.
    - For example, the pattern `e*` will match zero or more of the letter `e` and will result in the following:

```
sysadmin@localhost:~/Documents$  grep 're*d' red.txt
red
reeed
rd
reed
```

The Art of Shutting Down

How to shutdown the Linux system

NDG

Shutting Down

- The `shutdown` command brings down the system safely and notifies all users the system is going down.

```
shutdown [OPTIONS] TIME [MESSAGE]
```

- Formats of the `TIME` argument can be the word `now`, a time of day in the format `hh:mm` or the number of minutes to delay in the format `+minutes`.

```
root@localhost:~# shutdown +1 "Goodbye World!"

Broadcast message from sysadmin@localhost
        (/dev/console) at 2:23 ...

The system is going down for maintenance in 1 minute!
Goodbye World!
```

Getting Started with
Network Configuration

Learn about your computer on the network

Network Configuration

- Systems are connected to a network in order to communicate to other systems (like the internet).
- The `ifconfig` command is used to display network configuration information.
- `ifconfig` can also used to modify network settings.

```
root@localhost:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:01:02
          inet addr:192.168.1.2  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::42:c0ff:fea8:102/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1128 (1.1 KB)  TX bytes:648 (648.0 B)
```

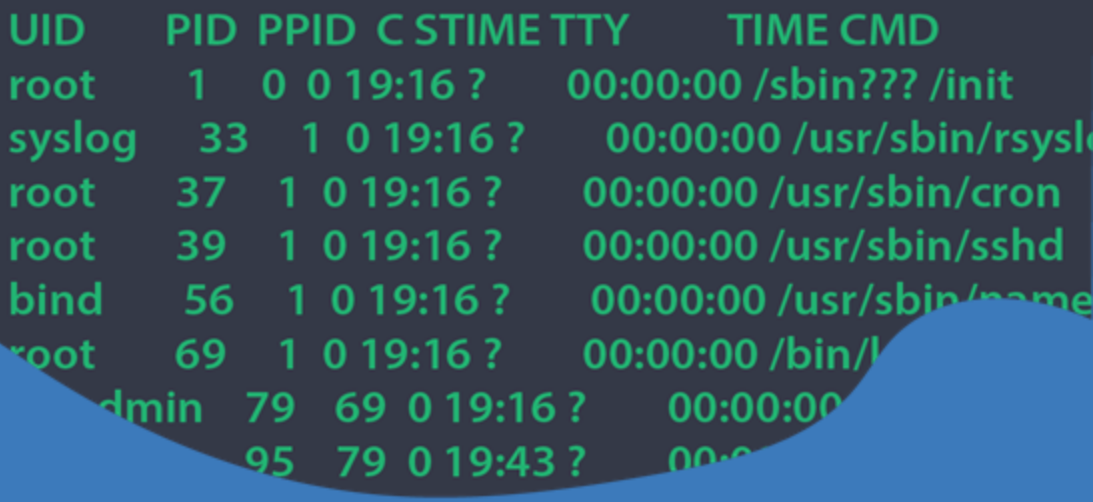
Network Configuration

- The `ping` command is used to verify connectivity between two computers on a network.
- The `ping` command sends “packets” to another machine by using an IP address.
- If the `ping` command is successful, the output will look like the following:

```
root@localhost:~# ping -c 2 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.027 ms

--- 192.168.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1329ms
rtt min/avg/max/mdev = 0.027/0.031/0.035/0.004 ms
```

Viewing Processes



How to view processes on the Linux system

Viewing Processes

- Running a command results in a *process*.

- In the Linux operating system, processes are executed with the privileges of the user who executes the command.

- The `ps` command can be used to list processes:

```
sysadmin@localhost:~$ ps
  PID TTY          TIME CMD
   87 ?        00:00:00 bash
  128 ?        00:00:00 ps
```

Viewing Processes

- The output of the `ps` command includes the following columns of information:
    - `PID`: The process identifier, which is unique to the process. This information is useful to control the process by its ID number.
    - `TTY`: The name of the terminal where the process is running.
    - `TIME`: The total amount of processor time used by the process.
    - `CMD`: The command that started the process.

Package Management

Learn about how software is installed on a Linux system

Package Management

- Package management is a system by which software can be installed, updated, queried or removed from a filesystem.

- In Linux, there are many different software package management systems.

- The Debian Advanced Package Tool `apt-get` is a front-end program that makes package management easy for Linux beginners.

Package Management

- Package files are commonly installed by downloading them directly from repositories located on Internet servers.

- Before installing a package, refresh the list of available packages using the `apt-get update` command.

- To search for packages, you can use the `apt-cache search` command:

```
apt-cache search [keyword]
```

- Once you've found the package that you want to install, you can install it with the `apt-get install` command:

```
sudo apt-get install [package]
```

Package Management

- To update all packages
    - First update the cache of all packages available with `apt-get update`.
    - Second, execute the `apt-get upgrade` command.

- The `apt-get` command is able to either remove or purge a package.
    - Purging deletes all package files, while removing deletes all but the configuration files.
    - To remove a package, use `apt-get remove` as an administrator
    - To purge a package, use `apt-get purge` as an administrator

Updating User Passwords

How to update a user's password on a Linux system

Updating User Passwords

- The `passwd` command is used to update user passwords.
- Users can only change their own passwords, whereas the root user can update the password for any user.
- To execute the `passwd` command use the following syntax:

```
passwd [OPTIONS] [USER]
```

- You will be prompted to enter the existing password once and the new password twice.

Updating User Passwords

- To view status information about your password use the `-S` option with the `passwd` command:

```
sysadmin@localhost:~$ passwd -S sysadmin

sysadmin P 03/01/2015 0 99999 7 -1
```

  - `sysadmin`: User Name
  - `P`: Password Status (`P` indicates usable status, `L` indicates locked, `NP` means no password)
  - `03/01/2015`: Change Date (date when password was last changed)
  - `0`: Minimum (minimum number of days that must pass before the current password can be changed by the user)
  - `99999`: Maximum (the maximum number of days remaining for the password to expire)
  - `7`: Warn (the number of days prior to password expiry that the user is warned)
  - `-1`: Inactive (the number of days after password expiry that the user account remains active)

Types of Redirection

Learn about input/output redirection on the command line

NDG

Redirection

- Input/Output redirection allows for information in the command line to be sent to files, devices, and other commands.

- There are three type of input/output:
    - **Standard Input** (STDIN): Information the command receives and processes when it is executed.

```
sysadmin@localhost:~$ ls
```

    - **Standard Output** (STDOUT): The output of the command.

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

    - **Standard Error** (STDERR): Error messages generated by commands that are not correctly executed.

```
sysadmin@localhost:~$ ls fakefile
ls: cannot access fakefile: No such file or directory
```

Redirection

- STDOUT can be used to write to files by redirecting command output to a file.

```
[COMMAND] > [FILE]
```

- For example, the output of the `cat` command can be redirected to a new file called `newfile1.txt`:

```
sysadmin@localhost:~/Documents$ cat food.txt
Food is good.
sysadmin@localhost:~/Documents$ cat food.txt > newfile1.txt
sysadmin@localhost:~/Documents$ cat newfile1.txt
Food is good.
```

- The single > character will overwrite any contents of a file, so to append content to a file use the double-greater than characters >>.

Using the Text Editor

Learn about Linux text editors

Text Editors

- There are numerous text editors in Linux (`nano`, `emacs`, `vi`)
- The most advanced text editor for Linux is called `vi`. Benefits of `vi` include:
    - Available on every Linux distribution in the world.
    - Can be executed both in a CLI (command line interface) and a GUI (graphical user interface).
    - Core functions have been around for decades.
- To get started using `vi`, type the command followed by the pathname to the file to edit or create:

```
sysadmin@localhost:~$ vi newfile.txt
```

Text Editors

- There are three modes used to navigate and edit in `vi`:
  - command mode
  - insert mode
  - ex mode

- Command Mode
  - Used to type commands that move around a document, manipulate text, and to access the other two modes.
  - Command Mode Movement
    - Movement commands in `vi` use a motion and an optional number prefix.
    - For example, `5h` would move the cursor five characters to the left.
    - Some other movement commands include:

      `j` - down one line
      `k` - up one line
      `$` - end of the line
      `l` - right one character
      `w` - one word forward
      `b` - one word back
      `^` - beginning of the line `$` - end of the line

Text Editors

- Command Mode Actions
  - `vi` uses the following three action commands:
    - `d` - Delete (cut)
    - `y` - Yank (copy)
    - `p` - Put (paste)
- The following syntax is used for command mode:

```
action [count] motion
```

  - For example:
    - `d3w` will delete the next three words
    - `yw` will yank the current word

Text Editors

- Insert Mode
  - Insert mode is used to add text to the document.
  - There a few ways to enter insert mode from command mode

`a` - Enter insert mode after cursor

`i` - Enter insert mode before cursor

`A` - Enter insert mode at end of line

`I` - Enter insert mode at beginning of line

`o` - Enter insert mode on blank line after cursor

`O` - Enter insert mode on blank line before cursor

- Ex Mode
  - Used to view or change settings, as well as carry out file-related commands like opening, saving or aborting changes to a file.
  - To get to ex mode, type a `:` character in command mode.
    - For example, `:w` will write (save) the file to the filesystem; `:q!` will quit without saving.

We hope you've enjoyed this brief introduction into the world of Linux!