Distros:

Starters:

1. **Ubuntu** *(And all it's derivatives)*
2. **Mint** *(And all it's derivatives)*

Others:

1. **CentOS** *(Great for servers apparently*
2. **Debian**
3. **OpenSUSE**
4. **Manjaro**
5. **Elementary OS**
6. **Fedora**
7. **ZorinOS**
8. **Gentoo**
9. **Ubermix** *(Great for kids)*

10. **Tails** *(That one that Snowden used and praised)*

There are many more distros and probably more being developed, but this is a pretty good starting list, if I say so myself.

Common Directories:

/ : "Root" top of the system hierarchy.

/bin : Binaries and other executable programs.

/etc : System configuration files.

/home : Home directories, for each user account.

/opt : Optional or third party software.

/tmp : Temporary space, typically cleared on reboot.

/usr : User related programs.

/var : Variable data, most notably log files.

Comprehensive Directory Listing:

/boot : Files needed to boot the operating system.

/cdrom : Mount point for CD-ROMs.

/cgroup : Controls Groups hierarchy.

/dev : Device files, typically controlled by the operating system and the system administrators.

Basic Linux Commands:

ls - List directory contents.

cd - Changes the current directory.

pwd - Displays the present working directory.

cat - Concatenates and displays files.

echo - Display arguments to the screen.

man - Displays the online manual.

exit - Exits the shell or your current session.

clear - Clears the screen.

Navigating Man Pages:

Enter - Move down one line.

Space - Move down one page.

g - Move to the top of the page.

G - Move to the bottom of the page.

q - Quit.

man -k search_term - Allows you to search the man pages.

Directory Shortcuts:

. : This directory.

.. : The parent directory.

cd : Change to the previous directory.

To execute a command/file or script in your current directory use " **cd ./name**"

Creating and Removing Directories:

mkdir [-p] directory - Create a directory

rmdir [-p] directory - Remove a directory.

rm -rf directory - Recursively removes directory.

• [**-p]** is not needed, but it's the parent directory command.

rmdir only removes empty directories, to remove a full directory and it's contents use "**rm -rf**".

!!! *Note: After removing something using the above commands, you'll no longer be able to retrieve them, use this at your own risk*. **!!!**

Decoding ls -l Output:

$ ls -l

Example: **-rw-rw-r-- 1 rick users 10400 Sep 27 09:30 funnydata.data**

Permission: **-rw-rw-r--**

Number of links: **1**

Owner name: **rick**

Group name: **users**

Number of bytes in the file: **10400**

Last modification time: **Sep 27 09:30**

File name: **funnydata.data**

Permissions:

$ ls -l

Example: **-rw-rw-r-- 1 rick users 10400 Sep 27 09:30 funnydata.data**

Symbol	Type
-	Regular file
d	Directory
l	Symbolic link

Symbol	Permission
r	Read
w	Write
x	Execute

Permission	File	Directory
Read (r)	Allows a file to be read.	Allows file names in the directory to be read.
Write (w)	Allows a file to be modified.	Allows entries to be modified within the directory.
Execute (x)	Allows the execution of a file.	Allows access to contents and metadata for entries.

Permission Symbol:	Permission Categories:
Symbol	**Category**
u	User
g	Group
o	Other
a	All

Groups:

- Every user is in at least one group.
- Users can belong to many groups.
- Groups are used to organize users.
- The ***"groups"*** command displays a user's groups.
- You can also used ***"id -Gn"***.

Secret Decoder Ring:

```
 __Type
|
▼   ▼--Group
-rw-rw-r-- 1 rick users 10400 Sep 27 09:30 funnydata.data
 ▲     ▲--Other
 |__User
```

Changing Permissions:

Item	Meaning
chmod	Change mode command
ugoa	User category user, group, other, all
+ - =	Add,subtract or set permissions
rwx	Read, Write, Execute

```
chmod g+w funnydata.data #Adds a permission

chmod g-w funnydata.data #Removes a permission

chmod g+wx funnydata.data #Adds the write and execute permissions

chmod u+rwx,g-x funnydata.data #Adds the read, write, execute permissions to "user" but
removes the execute permission from "group"

chmod a=r funnydata.data #Sets the permission read to "all"

chmod o= #Leaving the "=" assignment operator empty removes all the permissions, in this
case from "others"
```

Numeric Based Permissions:

r	w	x	
0	**0**	**0**	Value off
1	**1**	**1**	Binary value for on
4	**2**	**1**	Base 10 value for on

It's always Read then Write then Execute, in that specific order.

Ocat	Binary	String	Description
0	**0**	**---**	No permissions
1	**1**	**--x**	Execute only
2	**10**	**-w-**	Write only
3	**11**	**-wb**	Write and execute (2+1)
4	**100**	**r--**	Read only
5	**101**	**r-x**	Read and execute (4+1)
6	**110**	**rw-**	Read and write (4+2)
7	**111**	**rwx**	Read, write, and execute (4+2+1)

Order Has Meaning:

	U	G	O
Symbolic	**rwx**	**r-x**	**r--**
Binary	**111**	**101**	**100**
Decimal	**7**	**5**	**4**

Commonly Used Permissions:

Symbolic	Octal
-rwx------	**700**
-rwxr-xr-x	**755**
-rw-rw-r--	**664**
-rw-rw----	**660**
-rw-r--r--	**644**

When asked or from documentation you get asked to use **"777"** permission code ask yourself if there isn't a better way, since this gives everybody on the system access. It can cause trouble, as an example the injection of malicious code, the same goes for **"666"** permission code.

Working With Groups:

- New files belong to your primary group.
- The **"chgrp"** command changes the group.

```
groups #Will display the groups that you're in/assigned to
ricky sales #As an example

chgrp sales sales.data #Would add the sales group to the "sales.data" file
```

Directory Permission Revisited:

- Permissions on a directory can affect the files in the directory.
- If the file permission look correct, start checking directory permissions.
- Work your way up to the root.

File Creation Mask:

- File creation mask determines default permissions.
- If no mask were used, permissions would be:
  - **777** for directories
  - **666** for files

The umask Command:

```
umask [-S] [mode]
```

- Sets the file creation mask to mode, if given.
- Use **-S** for symbolic notation.

How does umask work:

	Directory	**File**
Base Permission	**777**	**666**
Subtract Umask	**-022**	**-022**
Creations Permission	**755**	**644**

	Directory	File
Base Permission	**777**	**666**
Subtract Umask	**-002**	**-002**
Creations Permissions	**775**	**664**

Octal Substraction Is an Estimation:

	Directory	File
Base Permission	**777**	**666**
Subtract Umask	**-007**	**-007**
Creations Permission	**770**	**660***

Common umask modes:

- **022**
- **002**
- **077**
- **007**

Table of all the resulting permutations of umask:

Octal	Binary	Dir Perms	File Perms
0	**0**	**rwx**	**rw-**
1	**1**	**rw-**	**rw-**
2	**10**	**r-x**	**r--**
3	**11**	**r--**	**r--**
4	**100**	**-wx**	**-w-**
5	**101**	**-w-**	**-w-**
6	**110**	**--x**	**---**
7	**111**	**---**	**---**

Special Modes:

- **umask 0022** is the same as **umask 022**.
- **chmod 0644** is the same as **chmod 644**.
- The special modes are:
  - **setuid**
  - **setgid**

- **sticky**

Take note that the special modes are declared by prepending a character to the octal mode that normally use with **"umask"** or **"chmod"**.