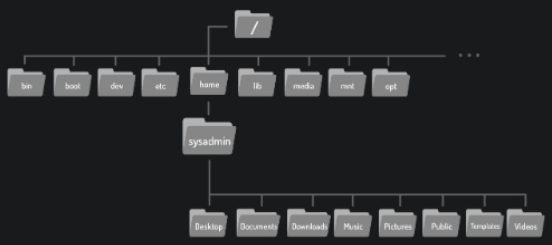
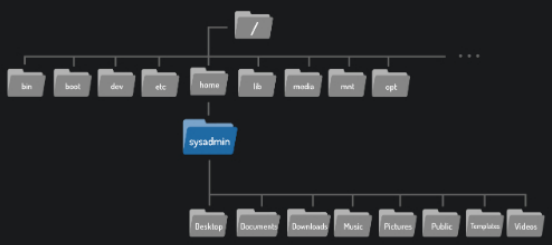
Changing Directories

Files are used to store data such as text, graphics and programs. Directories are a type of file used to store other files–they provide a hierarchical organizational structure. The image below shows an abbreviated version of the filesystem structure on the virtual machines.



When you start a fresh virtual machine, either by opening the course or after using the reset button, you are logged in as the `sysadmin` user in your home directory, highlighted below:



To navigate the filesystem structure, use the `cd` (change directory) command to change directories.

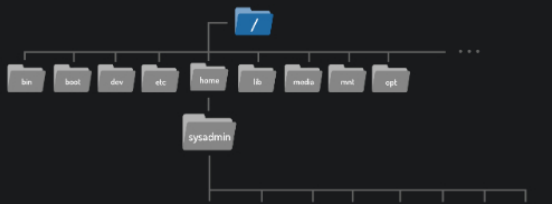
```
cd [options] [path]
```

If you look back at the graphic above, you will see the `Documents` directory is located within the `home` directory, where you are currently located. To move to the `Documents` directory, use it as argument to the `cd` command:

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

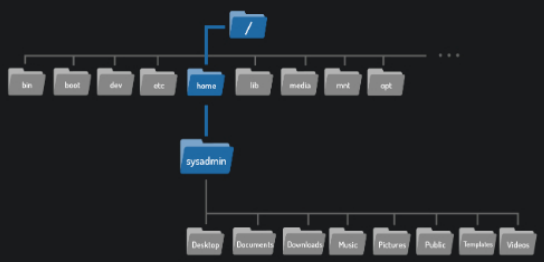
Directories are equivalent to folders on Windows and Mac OS. Like these more popular operating systems, a Linux directory structure has a top level. It is not called "My Computer", but rather the *root* directory and it is represented by the `/` character. To move to the root directory, use the `/` character as the argument to the `cd` command.

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



Toggle onscreen keyboard

The argument to the `cd` command is more than just the name of a directory, it is actually a *path*. A path is a list of directories separated by the `/` character. For example, `/home/sysadmin` is the path to your home directory:



If you think of the filesystem as a map, paths are the step-by-step directions; they can be used to indicate the location of any file within the filesystem. There are two types of paths: absolute and relative. Absolute paths start at the root of the filesystem, relative paths start from your current location.

Absolute Paths

An absolute path allows you to specify the exact location of a directory. It always starts at the root directory, therefore it always begins with the `/` character. The path to the home directory `/home/sysadmin` is an absolute path. The path begins at the root `/` directory, moves into the `home` directory, and then into the `sysadmin` directory. Following this path on a graphical user interface (GUI) like your home computer would look something like this:

Use this path as an argument to the `cd` command to move back into the home directory for the `sysadmin` user.

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

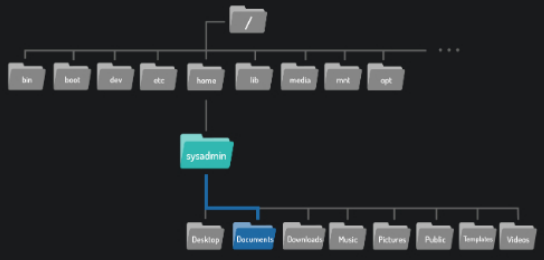
No output means the command succeeded. Go ahead and confirm this using the `pwd` command:

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

Relative Paths

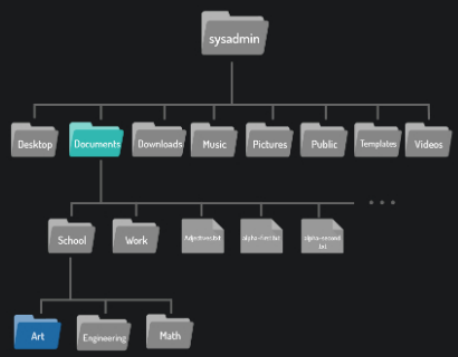
A relative path gives directions to a file relative to your current location in the filesystem. Relative paths do not start with the `/` character, they start with the name of a directory. Take another look at the first `cd` command example. The argument is an example of the simplest relative path: the name of a directory in your current location.

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

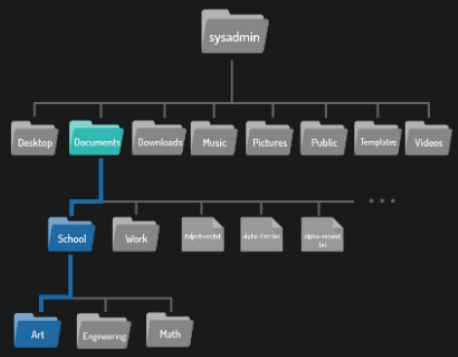


The image below shows a map of the files contained within the `sysadmin` directory

The image below shows a map of the files contained within the `sysadmin` directory. You are currently in the `Documents` directory and want to move to the `Art` directory:



A relative path begins in from with the current directory, however you don't include it in the path. The first step would be to move into the `School` directory, and then move into the `Art` directory. Use the `/` character to separate the directory names and the result `School/Art` is a relative path from the `Documents` directory to the `Art` directory:



Use the relative path as an argument to the `cd` command to move into the `Art` directory.

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

Use the `pwd` command to confirm the change:

```
sysadmin@localhost:~/Documents/School/Art$ pwd
/home/sysadmin/Documents/School/Art
```

Consider This

The output of the `pwd` command is the absolute path to the `Art` directory.

Consider This

In the example above the `cd` command followed the `School/Art` path:

```
cd School/Art
```

A path can also be broken down into multiple `cd` commands. The following set of commands would achieve the same results:

```
cd School
cd Art
```

Shortcuts

The `..` Characters

Regardless of which directory you are in, `..` always represents one directory higher relative to the current directory, sometimes referred to as the parent directory. To move from the `Art` directory back to the `School` directory:

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

The `.` Character

Regardless of which directory you are in, the `.` character always represents your current directory. For the `cd` this shortcut is not very useful, but it will come in handy for commands covered in subsequent sections.

The `~` Character

The home directory of the current user is represented by the `~` character. As stated above, you always begin as the `sysadmin` user, whose home is located at `/home/sysadmin`. To return to your home directory at any time execute the following command:

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