Hierarchical File System

Definitions

Hierarchical File System (HFS) A collection of files and directories organized in an upside-down tree structure. In this structure, the collection begins at a single directory. On Unix, this starting directory is called the ***root***, and is indicated by **/** Each file and directory on the system may be accessed by starting at the root and opening a series of directories.
When you are logged into Unix, you always are 'at' a particular 'location' in the filesystem. This is your ***current directory***, or your *working directory*.

Path: Directions to find a particular file or directory in a hierarchical file system. A path specifies the series of directories that must be opened to reach the target directory or file. Each directory in the path is called a ***component of the path*** and is separated from the next component by a **/** The final component of the path is the target directory or file.

Absolute Path: A set of directions from the root (**/**) to a target file or folder. An absolute path is independent of where you currently are (your current directory). An absolute path always starts with the root (**/**) or with a shorthand notation that infers the root, such as **~**
The absolute path to your current (or *working*) directory is displayed by the **pwd** command (**p**rint **w**orking **d**irectory)

Examples:

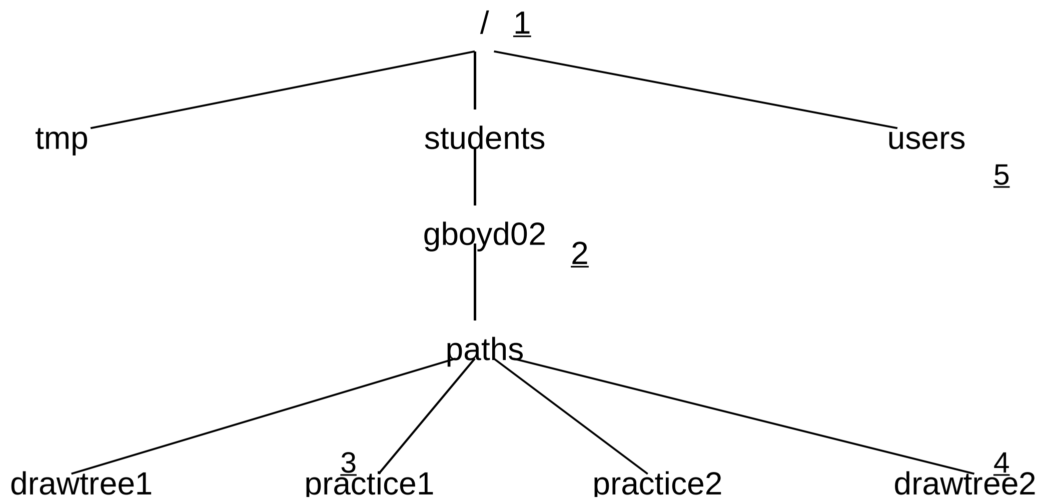
```
/students/joe01
/tmp/foo
/users/gboyd/167examples
/pub/cis/gboyd/cis167/
~/asmt01
```

Relative Path: Directions to proceed from your current directory to another file or directory in the file system. It is totally dependent on your current directory. A relative path never starts with **/** or anything that infers **/**

Examples:

```
asmt01
asmt01/typescript
../gboyd02
./typescript
```

Directory Tree: A map of a directory structure indicating how files and directories are organized hierarchically. A directory tree may be rooted at any directory. If it is rooted at **/** the tree may describe the entire HFS. Otherwise it describes a section (or **subtree**) of it. In both cases, the root will be labeled with a directory path, called the **root of the subtree**.
Directory trees usually are drawn so that files can be distinguished from directories. On our directory trees for class, files have a circle drawn around their names to distinguish them from directories, which do not.



Below is an exercise set for practicing paths. The answers are at the bottom. Cover the answers and fill in the commands referring to the directory tree above. (Note: only some of the directories that exist are represented in the directory tree above.)

Moving Around Using Absolute Paths

To go to	Use the command
1	**cd** ______________________
2	**cd** ______________________
3	**cd** ______________________
4	**cd** ______________________
5	**cd** ______________________

Moving Around Using Relative Paths

To go	Use the command
to 1	**cd** ______________________**(absolute path)**
from 1 to 2	**cd** ______________________
from 2 to 3	**cd** ______________________
from 3 to 4	**cd** ______________________
from 4 to 5	**cd** ______________________

Using Absolute paths:	Using Relative paths:
cd /	**cd /** (this is an absolute path)
cd /students/gboyd02	**cd students/gboyd02**
cd /students/gboyd02/paths/practice1	**cd paths/practice1**
cd /students/gboyd02/paths/drawtree2	**cd ../drawtree2**
cd /users	**cd ../../../../users**

Though relative paths are generally shorter than absolute ones, this is not always true.

One last note: some things are redundant in a path. Examples of this are directory segments that refer to the current directory **.** as well as redundant forward slashes. The following refer to the same location:

students/gboyd02/paths	**./students/gboyd02/paths/practice1/..**
./students/gboyd02/./paths///	**students////gboyd02/./../gboyd02/paths/**

Note that trailing slashes are considered redundant as well. **cat file1** and **cat file1/** both try to display **file1** in the current directory.