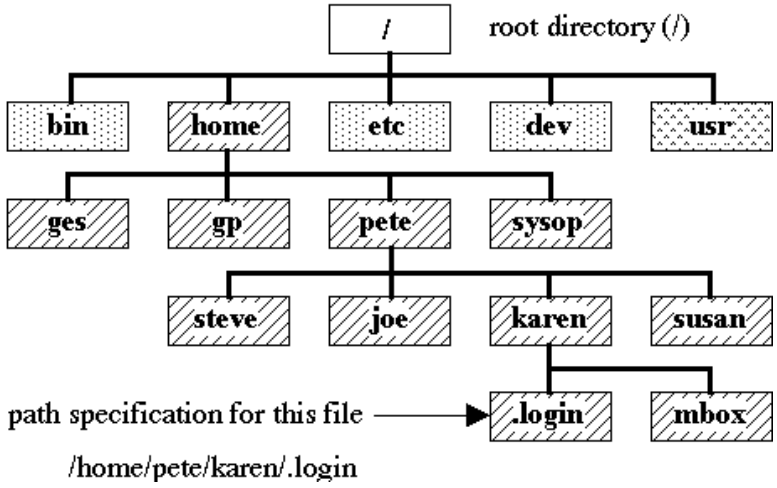
Navigating inside a supercomputer

Background

Once you are connected to the supercomputer, you need to be able to create and access folders and create and access files. **The folder/file structure of a supercomputer is very similar to that of your desktop or laptop**. **The main difference is that you will need navigate through folders and files from the command line.** Again, since supercomputers only understand UNIX language, the commands we will input from the command line are UNIX commands.

A typical file structure

The exact folders (directories from now on) and files in the supercomputer will depend on the user. However, the figure below is a good example of a typical supercomputer. One of the reasons why UNIX is chosen as the operating system for supercomputers is that while all users can typically access all files/directories in the supercomputer, they typically can only modify files/directories within their own **home directory**. Only the so-called **root user** can access and modify all files/directories, and typically only a handful of people in an organization are given the necessary information to connect to the supercomputer as the root user.



So, what is the home directory? Essentially is the directory that you are automatically placed at when you log in to the supercomputer. Typically, the name of this directory is the same as your username. For example, in the tree above, when the user karen logs in, she will automatically start at her home directory which is located at `/home/pete/karen`. The specification of the location of a directory is known as the **path.** The `/home/pete/karen` is an *absolute path* because it gives the position of a directory relative to the root directory. There are also *relative paths* that gives the position of a directory relative to your current directory. For instance, in the tree above, the relative path of the `karen` directory if your current directory is `pete` is `/karen`.

So how to navigate through directories? You will use the command **cd**. This command makes whatever directory you put after it your **c**urrent **d**irectory. For instance, for the tree above, to make `susan` your current directory (regardless of what your current directory is) you could type

```
cd /home/pete/susan
```

If your current directory is `karen` and you want to "go back one directory" to the directory `pete`, you could type

```
cd ..
```

If your current directory is `susan` and you want to go to the directory `steve`, you could type

```
cd ../steve
```

If at some point you get lost and you simply would like to go back to your home directory, you could type

```
cd
```

But how to see the content of a directory? You will use the command **ls**. This command **ls**ts the content of whatever directory you put after it. For instance, for the tree above, to see the content of the directory `karen` (regardless of what your current directory is) you could type

```
ls /home/pete/karen
```

which would display the following in your terminal

```
.login      mbox
```

If your current directory is `karen`, to see the content of the directory `home` you could type

```
ls ../..
```

which would display the following in your terminal

```
ges      gp      pete      sysop
```

But how to open and view a file? You will use the command **vi**. This command opens whatever file you put after it and lets you **v**iew **i**nside the file. For instance, for the tree above, to see inside the file `.login` (regardless of what your current directory is) you could type

```
vi /home/pete/karen/.login
```

If your current directory is `karen` and you want to see inside the file `.login`, you could type

```
vi .login
```

If your current directory is `pete` and you want to see inside the file `.login`, you could type

```
vi /karen/.login
```

But how to execute a program? You will use the command **./**. This command executes whatever program you put after it (provided you have permission to do so). Keep in mind that a program is just a special kind of file. For instance, for the three above, to execute the program `mbox` (regardless of what your current directory is) you could type

```
./home/pete/karen/mbox
```

If your current directory is `karen` and you want to execute the program `mbox`, you could type

```
./mbox
```

If your current directory is `susan` and you want to execute the program `mbox`, you could type

```
./../karen/mbox
```

Hopefully, with the above examples you start to see some common themes or patterns about working within a supercomputer terminal. You can go ahead and try to explore the supercomputer Mio. At this point, your home directory is empty, but you can still navigate through other directories in the supercomputer (do not worry, you cannot break anything!).

Note that alternatively you can do a great deal of navigation in the supercomputer using WinSCP (if your local computer runs on Windows) or Cyber Duck (if your local computer runs on MacOS) in a way that is more familiar to you right now. However, you cannot execute programs (run codes) from WinSCP or Cyber Duck.

More specifically, with either WInSCp or with Cyber Duck, you can see files and folders in the supercomputer very similar to how you see files in your Windows Explorer (in Windows computers) or Finder (in Mac computers). Moreover, you can transfer files and folders from your local computer to the supercomputer (and vice versa), by dragging the files or folders from the Windows Explorer (or Finder) to the WinSCP (or Cyber Duck) window.

- You can download WinSCP here: https://winscp.net/eng/download.php
- You can download Cyber Duck here: https://cyberduck.io/download/

Both downloads are free. Once you open the programs, you will need to “Open a session” or “Open a connection”. Remember that the “Host name” or “Server name” is **mio.mines.edu**. If you are given a choice for connection protocol, pick Secure File Transfer Protocol (SFTP). You will need to provide your username and your password as you were taught to do in the previous tutorial.