

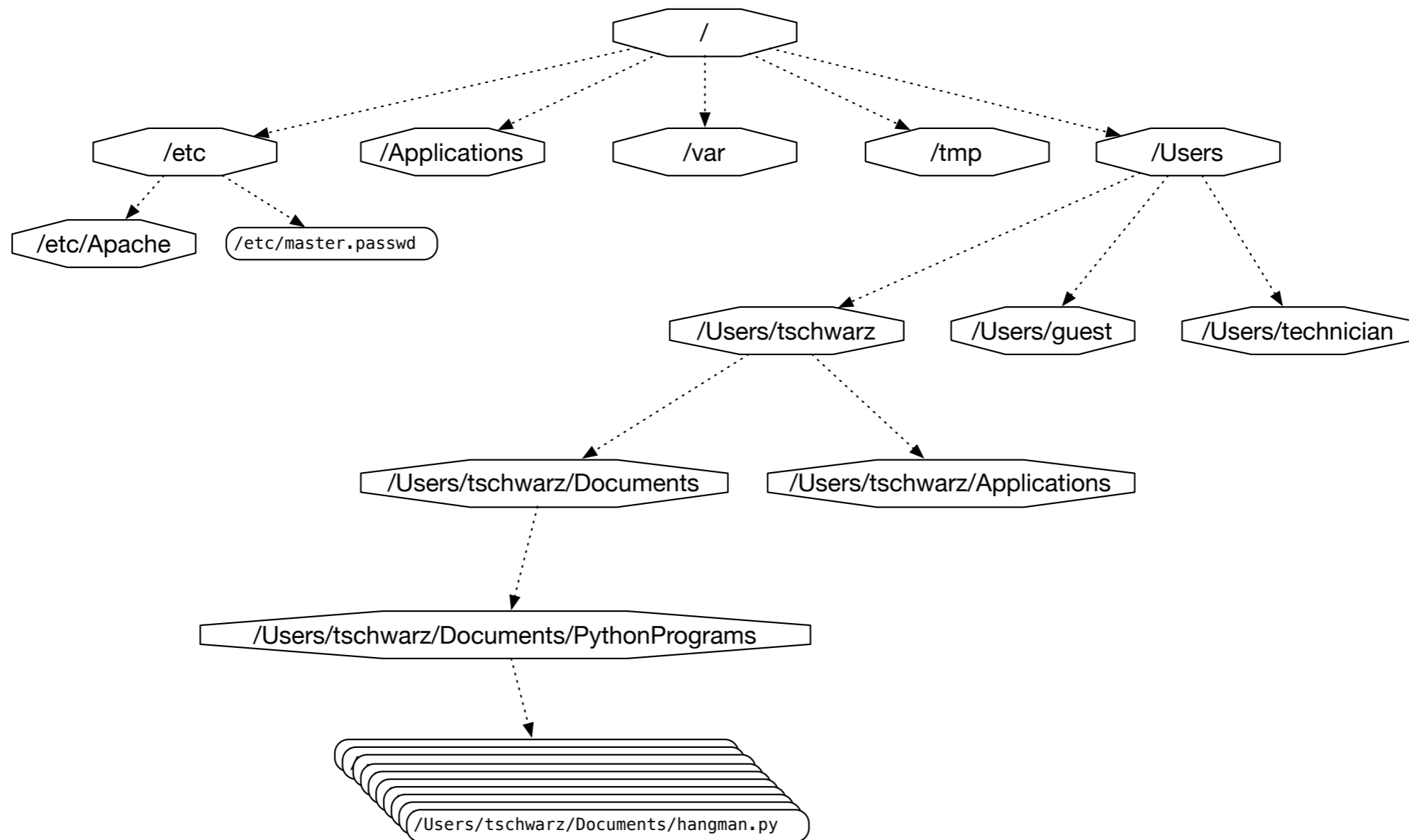
Files in Python

Thomas Schwarz, SJ

Files

- Files
 - Basic container of data in modern computing system
 - Organized into a hierarchy of directories

Files



A small subset of directories and files on a system

Files in Python

- Files accessed in
 - text mode
 - Contents interpreted according to encoding
 - binary mode
 - Contents not interpreted

Files in Python

- Python interacts by files through
 - reading
 - writing / appending
 - both

Files in Python

- Files need to be opened
 - File given by name
 - Relative path: Navigation from directory of the file
 - Absolute path: Navigation from the root of the file system

Files in Python

- File Name Examples:

- Absolute path on a Mac / Unix

`/Users/tjschwarzs/Google Drive/AATeaching/Python/Programs/pr.py`

- Relative path on a Mac / Unix

- “../” means move up on directory

`pr.py`

`../Slides/week7.key`

Files in Python

- Windows uses backward slashes to separate directories in a file name
 - Sometimes need to be escaped: \\
 - Absolute paths need to include drive name:
 - `c:\\users\\tschwarz\\My Documents\\Teaching\\temp.py`
- *We will typically read and create files in the same directory as the python program is located*

Files in Python

- Before files are used, program needs to open them
- After they are being used, program should close them
 - Will automatically closed when program terminates
 - Long-running programs could hog resources

Opening Files in Python

- File objects have normal variable names

```
inFile = open("data.txt", "w")
```

- opens a file “data.txt” in write mode

- open takes :
 - file name — absolute / relative path
 - mode — r (read), w (write), a (appending)
 - mode — b (binary), “” or t (text mode)

Closing Files in Python

- We close file by invoking close
 - `inFile.close()`

Why we need to close files

- Files are automatically closed when the program terminates
- When one application has opened a file for writing it acquires a write lock on the file and no other application can access the file.
- When one application has opened a file for reading, it acquires a read lock on the file and no other application can write to it.
- If you write programs that last more than a few seconds, you do not want to hog files when you do not need them.

With-clauses

- Python 3 allows us to open and close files in a single block (context)

```
with open("twoft8.11.txt") as inFile, open("twoftres8.11.txt",  
"w") as outFile:
```

```
    #Here you work with the file
```