REDIRECTION

Monday, 10 May 2021 6:47 PM

Redirection and Pipes

- Processes normally have three files open:

```
0 => stdin,  1 => stdout,  2 => stderr
```

- `command > stdout-here 2> stderr-here < stdin-from-here`

- `command &> file`

```
# file gets stdout and stderr from
command, file is created or
overwritten
```

Command > stdout here (the file gets std output from the command)
Command < stdin here (command getsthe stdin from there)

PIPE

Redirection and Pipes

- `command | command2`

```
# command2's stdin comes from
# command's stdout
```

The left command becomes the stdin for the command2

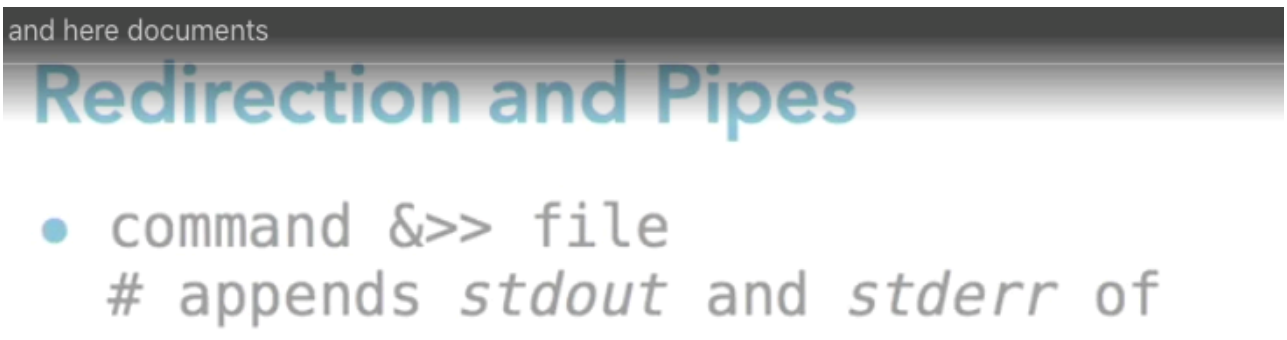
- command 2>&1 | command2
  # gets *stdout* and *stderr* from *command*

APPEND TO A FILE:

Redirection and Pipes

- command |& command2
  # alternative way for *command2* to
  # get *command's stdout* and *stderr*
  # as *its stdin*

- command >> file
  # appends *stdout* of *command* to
  # end of file



```
# command to end of file
```

INPUTS:

Here Documents: <<

- Here documents are a way to embed input for standard input inside of a script.
- They avoid having to create a new file just to hold some input values.

OPEN AND CLOSE FILE DESCRIPTORS:

Open and Close File Descriptors

- ```
  exec N< myfile
  # opens file descriptor N for
  # reading from file myfile
  ```
- ```
  exec N> myfile
  ```

```
# opens file descriptor N for
# writing to myfile
```

Open and Close File Descriptors

- ```
  exec N<> myfile
  # opens file descriptor N for
  # reading & writing with myfile
  ```

- ```
  exec N>&- or exec N<&-
  # closes file descriptor N
  ```

LSOF- LIST open files that the process has opened:

Open and Close File Descriptors

- Use `lsof` to see what file descriptors for a process are open

- ```
  exec 7>/tmp/myfile7
  lsof -p $$
  # $$ is shell's PID
  ```
  ```