dup

Eric Missimer

dup

- `int dup(int oldfd)`
  - Uses the lowest-numbered unused descriptor for the new descriptor
  - On success returns the new file descriptor, on error returns -1

dup2

- `int dup2(int oldfd, int newfd)`
  - Makes `newfd` be the copy of `oldfd`, closing `newfd` first if necessary
  - If `oldfd` is not a valid file descriptor, then the call fails, and `newfd` is not closed
  - If `oldfd` is a valid file descriptor, and `newfd` has the same value as `oldfd`, then dup2 does nothing, and returns `newfd`

Pipe in Shells

- For things like `ls > foo.txt` we open up `foo.txt` and set stdout to point to he file

Pipe in Shells

- For things like `ls > foo.txt` we open up `foo.txt` and set stdout to point to he file
- What about `ls | grep foo.txt`?
- What do we open up and set as the output for `ls` and the input for `grep`?

Pipe in Shells

- For things like `ls > foo.txt` we open up `foo.txt` and set stdout to point to he file
- What about `ls | grep foo.txt`?
- What do we open up and set as the output for `ls` and the input for `grep`?
- Answer: an unnamed pipe

pipe

- `int pipe(int pipefd[2])`
- creates a unidirectional data channel that can be used for interprocess communication
- pipefd[0] refers to the read end of the pipe.
- pipefd[1] refers to the write end of the pipe.
- Data written to the write end of the pipe is buffered by the kernel until it is read from the read end of the pipe.