# Process Groups

- Short digression on process groups, because
  - You will need them for part Part 8 of Assignment 4
  - A couple of groups ran into confusing behavior – process groups explain it.

# What we know

- We know that every process has a process id (PID).
- We know that fork creates a new process with a new process ID.
- We know that file descriptors are copied during a fork so that a parent and child have identical file descriptors unless we do something different to change that.

# What we don't yet know

- In addition to a process ID, processes belong to a process group (as well as a session; we'll not be talking about sessions further).
- Process groups allow you to batch a set of processes together into (you guessed it) an abstraction to provide better control over them.
- For example (from A4):
  - "…processes spawn new helper processes. If a user kills a command with Control-C, the helper processes should also die. Unix's solution uses *process groups*, where a process group is a set of processes. The Control-C key kills all members of the current foreground process group, not just the current foreground process.

# What people saw in class Tuesday

- When we added the fork call to cat_wrapper_out, some students started seeing an error message from cat "`cat: -: Input/output error`"
- Parsing that:
  - The `cat` command that was exec is the program with an error.
  - The -: is a filename; UNIX uses "-" to indicate standard in.
  - Typing "`cat –`" is the same as typeing "`cat`"
  - So, the program cat experienced an input/output error while trying to read standard IO.

# What happened?

- The parent exited, orphaning the child process.
- When a child is orphaned, it:
  - Keeps running, but
  - Is forcibly disassociated from its controlling terminal
- *If programs that continue running try to access that terminal, those attempts result in errors, with errno set to EIO.*
- I looked around for a good discussion of this and liked the one I found at: http://www.informit.com/articles/article.aspx?p=397655&seqNum=6.