

## Looking over previous tasks...

I might want to have another look at inter-process communication via pipes because...:

- It's confusing as to when and where we make particular function calls
- It's also confusing to work out which file descriptors act as standard in/out for which processes

I think that a few more examples/practice questions in this area would be appreciated.

## Notes on pthread\_cond\_\*\*\*:

What are condition variables?

- Condition variables are variables of type pthread\_cond\_t.
- They allow a thread to relinquish control of the processor, and wait until a condition is reached.

Some of the functions:

- pthread\_cond\_init or use PTHREAD\_COND\_INITIALIZER to create a new control variable.
- pthread\_cond\_wait and pthread\_cond\_timedwait block a thread until a signal is generated from another thread, with the second function unblocking after a specified period of time (returning ETIMEDOUT)
- pthread\_cond\_signal is used to signal to one thread that is blocked waiting on a condition that it can proceed once again
- pthread\_cond\_broadcast is like signal but to all threads blocked on the condition at once
- pthread\_cond\_destroy is used to destroy a condition variable after it has no more use

Mutex considerations:

- Because a thread can begin waiting for a condition, but another thread can signal before it can complete that operation, you may get deadlocks – it never receives the signal
- So you should use a mutex alongside condition variables – and obtain the lock for that mutex before calling pthread\_cond\_wait
- Logic considerations: if you call wait, you should make sure that signal is called when the condition that makes it call wait becomes true

References:

- pthread\_cond\_\*\*\* man page
- <http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>