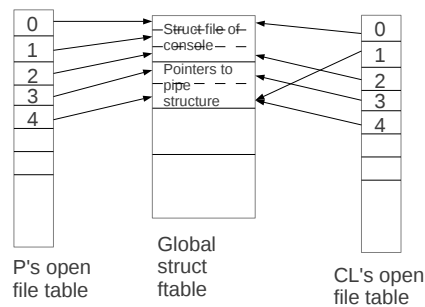


# CS347: Operating Systems

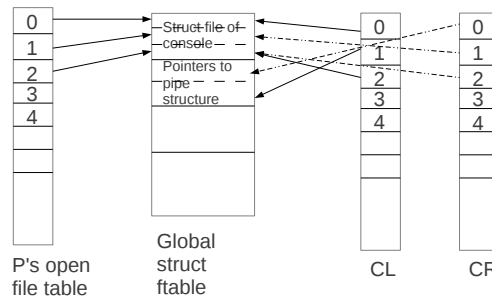
## Problem Set 4: Solutions

1. (a) The fd 1 of the child process has been modified to point to one of the pipe's file descriptors



by the close and dup operations.

- (b) CL and CR are now connected to either ends of the pipe. The parent has no pointers to the



pipe any more.

2. (a)
- A free inode on disk is marked as allocated for this file.
  - An inode from the in-memory inode cache is allocated to hold data for this new inode number.
  - A directory entry is written into the parent directory on disk, to point to the new inode.
  - An entry is created in the in-memory open file table to point to the inode in cache.
  - An entry is added to the in-memory per-process file descriptor table to point to the open file table entry.
- (b) The directory entry should be added after the on-disk inode is marked as free. The memory operations can happen in any order, as the memory data structures will not survive a crash.

3.
  - read  $i(D1)$ , read  $d(D1)$ —this will give us the inode number of F1.
  - read  $i(F1)$ . After reading the inode and bringing it to cache, its link count will be updated. At this point, the inode is only updated in the buffer cache.
  - read  $i(D2)$ , read  $d(D2)$ —we check that the new file name F2 does not exist in the directory. After this, the directory contents are updated, and a note is made in the log.
  - Now, the log starts committing. This transaction has two modified blocks (the inode of F1 and the directory content of D2). So we will see two disk blocks written to the log: write to L+1 and L+2, followed by a write to block L (the log header).
  - Next, the transactions are installed: a write to disk blocks  $i(F1)$  and  $d(D2)$ .
  - Finally, another write to block L to clear the transaction header.
4. A DMA driver frees up CPU cycles that would have been spent copying data from the device to physical memory.
5. A
6. B,C