File Descriptor Management

File Descriptors

The POSIX file access APIs (e.g. read(2) and write(2)) operate on file descriptors. A file descriptor is an integer*. Normally, when a process starts, it has (at least) three already-open file descriptors:

0: standard input (read only)
1: standard output (write only)
2: standard error (write only)

As additional files are opened (e.g. with open(2), creat(2) or pipe(2)) each is assigned the lowest unused file descriptor.

(*If you want a little more truth, these integers are indexes into a table of open files that the Operating System maintains for each process)

Input/Output Redirection

Two other operations can be used to manipulate file descriptors:

- close(2) closes the file on the specified file descriptor, making that file descriptor available for reuse.
- dup(2) allocates the lowest available file descriptor, and creates another reference to the open file instance on the specified file descriptor.

Input/output redirection is accomplished by:

- opening the new input/output file
- closing the file descriptor (0, 1, 2) to be replaced
- duplicate the new input/output file to the (newly vacated) file descriptor to be replaced
- close the (now redundant) file descriptor on to which that file was originally opened

Example: input redirection

```
int ifd = open(newfile, O_RDONLY);
if (ifd >= 0) {
        close(0);
        dup(ifd);
        close(ifd);
}
```

Example: output redirection

```
int ofd = creat(newfile, 0666);
if (ofd >= 0) {
        close(1);
        dup(ofd);
        close(ofd);
}
```