Darren Dawson
dwdawson@ucsc.edu
CSE 130

Lab 0 Writeup

Testing

In order to test dog, I compared it to the cat function using 'diff" using the following command.

- diff -B <(cat filename) <(./dog filename)

Passed Tests:

- diff -B <(cat README.md) <(./dog README.md)
- diff -B <(cat dog.c) <(./dog README.md)
- diff -B <(cat noFile) <(./dog noFile)
- diff -B <(cat dog.c) <(./dog noFile)
- diff -B <(cat dog.c README.md) <(./dog dog.c README.md)
- diff -B <(cat DESIGN.pdf) <(./dog DESIGN.pdf)

In each of these cases, the expected output was achieved. For tests where the input files were the same, the outputs matched identically (and so nothing was returned). This includes the tests for pdfs and those with multiple file arguments. For situations where the input files are not the same, it spotted the differences between them.

The only case that was not explicitly tested using diff was the case where dog reads from stdin. However, I did a manual side by side and these results seemed to line up.

Questions

Q: How does the code for handling a file differs from that for handling standard input? What concept is this an example of?

A: In both cases, you use the read(2) function to fill the character buffer with bytes to write to stdout. However, you call read(pointer to file, …) and read(standard input, …) for reading from a file vs standard input respectively.

Because you are able to use the same function calls with different arguments (and different sources - file vs stdin), this is an example of function overloading.