Assignment 3 – XD Report Template

Your Name

CSE 13S – Winter 24

Purpose

ANS

Our purpose in this assignment is to explore the xxd program. We will be displaying binary files in their hex representations. We will work to build our own functional xxd programs

Questions

Please answer the following questions before you start coding. They will help guide you through the assignment. To make the grader's life easier, please do not remove the questions, and simply put your answers below the text of each question.

- What is a buffer? Why use one?

  ANS

  A buffer is a temporary storage of data while it is being transferred from one place to another. We can use the buffer to read in and modify files.

- What is the return value of `read()`? What are the inputs?

  ANS

  The file descriptor from which to read, buffer where the read data will be stored, and the number of bytes to read are the inputs. The output will be the number of bytes successfully read.

- What is a file no. ? What are the file numbers of `stdin`, `stdout`, and `stderr`?

  ANS

  A file no. is an integer that represents a file. stdin is '0', stdout is '1' and stderr is '2'.

- What are the cases in which `read(0,16)` will return 16? When will it *not* return 16?

  ANS

  The function will return 16 if 16 bytes were read, however it will return less if there is an error or it is at the end of the file and unable to read the full 16 bytes.

- Give at least 2 (very differnt) cases in which a file can not be read all at once

ANS

A very large file size or a buffer size that is too small will prevent a file being read all at once.

Testing

ANS

To test my code I will compare expected outputs for a variety of inputs, from positive and negative numbers, to special characters, the keyboard, and a standard basic input. All inputs are tested for expected output in a delayed input setup.

How to Use the Program

The xd program is ran from the command line. The user will use "make" to creat the executable, and type out "./xd test.txt" for example, and the output will be printed to the console.

```
cadenroberts@cse13svm:~/cawrober/asgn3$ make
clang -Werror -Wall -Wextra -Wconversion -Wdouble-promotion -Wstrict-prototypes -pedantic -c -o
xd.o xd.c clang xd.o -o xd
cadenroberts@cse13svm:~/cawrober/asgn3$ ./xd tests/test.txt
00000000: 3031 3233 3435 3637 3839 6162 6364 6566 0123456789abcdef
00000010: 0a                                       .
```

Program Design

The program uses basic open(), read(), and close() functions to manage the file input. An array is used to read the buffer, and integers keep track of other important information.

Pseudocode

ANS

```
Check argc to be 1, setting it to STDIN_FILENO, or 2, set to open(argv[1], O_RDONLY)
Initialize structures needed
While bytes are left to read:
    print index of first byte in hexadecimal padded to 8 digits, followed by ": "
    print hex values of bytes passed in, padded to 2 digits, printing 8 pairs of 2 digits with a
        single space between each set of 4 digits, using a double space if there is not a byte left
         to read
    print double space and the char respresentations of the bytes on that line, with '.' being
        printed for any ASCII not between 32 and 126
    print newline and offset where we are reading from in the file
```

Function Descriptions

ANS

open(argv[1], O_RDONLY) will return an integer serving as the file descriptor. read(fd, buffer, 16) will return an integer equal to the number of bytes successfully read. close(fd) will close the specified file.

Optimizations

This section is optional, but is required if you do the extra credit. It due `only` on your final design. You do not need it on your initial.

In what way did you make your code shorter. List everything you did!

References