Assignment 3 – XXD Report Template

Nickolas Tran

CSE 13S – Winter 24

Purpose

In this assignment, you have to build your own version of the XXD program, which is used to display hexadecimals of binary files. Your program will replicate the key functionalities of XXD, such as reading a file from stdin and printing it out to stdout if no arguments are provided. You will implement buffered reading using system calls to interact with your system. The output will consist of three columns: the index of the first byte in hexadecimal, the hex values of the bytes grouped in 16-byte segments, and the ASCII representation of the bytes. Your program will handle errors by returning a non-zero error code for invalid filenames or multiple arguments.

Questions

- What is a buffer? Why use one?
  A buffer is a temporary storage space for data. It creates temporary storage that allows for a more efficient performance as it allows programs to work in small, manageable chunks.

- What is the return value of `read()`? What are the inputs?
  The return value of `read()` is -1. The inputs are the file descriptor, the pointer to the buffer to store, and the number of bytes to read into the buffer.

- What is a file no. ? What are the file numbers of `stdin`, `stdout`, and `stderr`?
  A file number is a file descriptor associated with an open file. Which is used to keep track of the file via an assoicated number and can do stuff like such as reading, writing, or closing.
  1. stdin: File number - 0
  2. stdout: File number - 1
  3. stderr: File number - 2

- What are the cases in which `read(0,16)` will return 16? When will it *not* return 16?
  Cases returning 16
  1. Data available in stdin
  2. Files containing at least 16 bytes

  Cases returning not 16
  1. Not enough data in stdin
  2. EOF in a file.

- Give at least 2 (very different) cases in which a file can not be read all at once.
  1. Large file size
  2. File on a remote network location

Testing

- Functionality: Tests with one or zero arguments.
- Existence: Testing with valid filename.
- Hexadecimal Representation: Verifies that the representation is correct for the input files.
- ASCII Representation: Verifies that the representation is correct for the input files.
- Error testing: Making sure that the invalid inputs of the user is recorded and handles it. And checks when error messages are informative.
- Input Delays: Create a test script to simulate delays to ensure thhe program reads and processes data correctly.
- Boundary Tests: Testing with large files, exactly 16 bytes, and with an empty file.
- Clang Formatting: Making sure that the code is formatted to the clang-format file before running.

How to Use the Program

The user uses "make" in the command terminal to recompile the set of rules created by the program.

```
make
```

The user then uses "./xd" to run the program. (Having no file name will read from stdin)

```
./xd
```

The user then uses "./xd file name" to run the program. (Having a file name will read fom the file)

```
./xd <file name>
```

Program Design

- Buffer Array: Holds the binary data read from the file or stdin. Passed to the print hex ascii function for formatting and printing.
- Print Hex ASCII Function: Iterates through the buffer, printing the hexadecimal representation of each byte. Prints a space and then the ASCII representation of each character, replacing non-printable characters with dots.
- Main Function: Checks the number of command-line arguments to ensure correct usage. Executes the program by handling command line arguments.

Pseudocode

Give the reader a top down description of your code! How will you break it down? What features will your code have? How will you implement each function.

```
FUNCTION print_hex_ascii(buffer: char[], length: size_t)
    FOR i FROM 0 TO length - 1
        PRINT hexadecimal representation of buffer[i] followed by a space
    END FOR

    FOR i FROM length TO BUFFER_SIZE - 1
        PRINT three spaces
    END FOR

    PRINT a space

    FOR i FROM 0 TO length - 1
        IF buffer[i] is between ASCII 32 and 126 (inclusive)
            PRINT buffer[i]
        ELSE
            PRINT "."
        END IF
    END FOR

    PRINT newline
END FUNCTION
```

Function Descriptions

For function print hex ascii():

- Input: buffer and length
- Output: none (prints the formatted output)
- Purpose: This function takes a buffer containing binary data and prints its hexadecimal representation along with the corresponding ASCII characters. It ensures proper formatting according to the xxd output format.