Assignment 3 – XD Report Template

Nithin Duvvuru

CSE 13S – Winter 24

Purpose

Audience for this section: Pretend that you are working in industry, and write this paragraph for your boss. You are answering the basic question, "What does this thing do?". This section can be short. A single paragraph is okay.

Do not just copy the assignment PDF to complete this section, use your own words.

This program is designed to replicate the basic functionality of the xxd program. xxd is a tool used to display binary files in hexadecimal representation, making them easier to read. The program takes files as input and prints their hexadecimal representation along with ASCII characters, showing the contents of the file in a readable format. It operates with either no arguments, reading from stdin and printing to stdout, or with one argument, treating it as a filename to read from. The output format includes hexadecimal byte indices, hex values of the bytes passed in, and their ASCII representation. The program must handle errors such as invalid filenames or multiple arguments by returning a non-zero error code, but must exit cleanly for all other error circumstances, including memory cleanup before exit.

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? **A buffer is a temporary storage area used to hold data while its being transferred between two locations.**
- What is the return value of `read()`? What are the inputs? **The return value is the number of bytes read and the input is the file descriptor.**
- What is a file no. ? What are the file numbers of `stdin`, `stdout`, and `stderr`? **A file number is an integer assosciated with an open file in the OS. stdin: 0, stdout: 1, stderr: 2.**
- What are the cases in which `read(0,16)` will return 16? When will it *not* return 16? **It will return 16 when there are 16 bytes available to be read from stdin but will not return 16 when there are less than 16 bytes available or there is an error when reading.**
- Give at least 2 (very differnt) cases in which a file can not be read all at once **1. The file is too large. 2. The file is actively being written to.**

Testing

List what you will do to test your code. Make sure this is comprehensive. [1] Be sure to test inputs with delays.

[1] This question is a whole lot more vague than it has been the last few assignments. Continue to answer it with the same level of detail and thought.

I would test the code to replicate the functionality we are trying to mimic xxd by first validating all the individual functions and then making sure they all work properly together and output the exact same xxd.

How to Use the Program

Audience: Write this section for the user of your program. You are answering the basic question, "How do I use this thing?". Don't copy the assignment exactly; explain this in your own words. This section will be longer for a more complicated program and shorter for a less complicated program. You should show how to compile and run your program. You should also describe any optional flags or inputs that your program uses, and what they do.

To show "code font" text within a paragraph, you can use `\lstinline{}`, which will look like this: `text`.

For a code block, use `\begin{lstlisting}` and `\end{lstlisting}`, which will look like this:

```
Here is some code in lstlisting.
```

And if you want a box around the code text, then use `\begin{lstlisting}[frame=single]` and `\end{lstlisting}`

which will look like this:

```
Here is some framed code (lstlisting) text.
```

Want to make a footnote? Here's how.[2]

Do you need to cite a reference? You do that by putting the reference in the file `bibtex.bib`, and then you cite your reference like this[1][2][3].

To use this program, if no arguments are provided, the program will read from standard input and print the hexadecimal representation to standard output. Alternatively, you can specify a filename as an argument, and the program will read from that file instead, displaying its contents in a hexadecimal representation.

Program Design

Audience: Write this section for someone who will maintain your program. In industry you maintain your own programs, and so your audience could be future you! List the main data structures and the main algorithms. You are answering the basic question, "How is this thing organized so that I can have a chance of fixing it?". This section will be longer for a more complicated program and shorter for a less complicated program.

The program will use file descriptors for input/output operations and buffers for file reading. We use open() and read() to read data from files and will format the output according to xxd. We will handle 2 errors, one for when there is more than 1 argument, and another for when the filename is invalid.

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.

The code will begin with validating the users input. Then we will use open() and read() to parse through the data we need, using a while loop to make sure we have 16 bytes of data or we are at the end of the file before converting to hexadecimal.

[2] This is my footnote.

Function Descriptions

For each function in your program, you will need to explain your thought process. This means doing the following

- The inputs of every function (even if it's not a parameter)
- The outputs of every function (even if it's not the return value)
- The purpose of each function, a brief description about a sentence long.
- For more complicated functions, include pseudocode that describes how the function works
- For more complicated functions, also include a description of your decision making process; why you chose to use any data structures or control flows that you did.

  main function: takes argc and argv[] as inputs. Outputs exit code 0 for success or 1 for error. Will handle the file and arguments from the user.

  format output function: takes in the data that needs to be formatted into hexadecimal and will print it out in the xxd format.

Do not simply use your code to describe this. This section should be readable to a person with little to no code knowledge. **DO NOT JUST PUT THE FUNCTION SIGNATURES HERE. MORE EXPLANATION IS REQUIRED.**

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

[1] Wikipedia contributors. C (programming language) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/C_(programming_language), 2023. [Online; accessed 20-April-2023].

[2] Robert Mecklenburg. *Managing Projects with GNU Make, 3rd ed.* O'Reilly, Cambridge, Mass., 2005.

[3] Walter R. Tschinkel. Just scoring points. *The Chronicle of Higher Education*, 53(32):B13, 2007.