Assignment 3 – XD Report Template

Your Name

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.

The purpose of this program is to mimic the basic features of the xxd tool. More specifically, it is able to read in a file and output the contents of the file in hexadecimal as well as ASCII representation.

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 loads in data that is being read and stores it there temporarily and then returns all of it at once when it is requested. A buffer has a few advantages. One of the advantages is that it allows you to manage the pace of data flow, which can prevent congestion in the system.
- What is the return value of `read()`? What are the inputs?
  - The read() function takes in 3 parameters. The file descriptor to read data from, a pointer to the buffer where the data will be stored, and the number of bytes to read from the file descriptor. The function returns the number of bytes read. If an error occurs, it returns -1. If the end of file is reached, it returns 0.
- What is a file no. ? What are the file numbers of `stdin`, `stdout`, and `stderr`?
  - File no. or file descriptor is used identify and open files . The files stdin, stdout, and stderr have file numbers 0,1,2 respectively.
- What are the cases in which `read(0,16)` will return 16? When will it *not* return 16?
  - It will return 16 when it reads 16 bytes. it will not return 16 if there are fewer than 16 bytes for it to read, like if it were to reach the end of the file.
- Give at least 2 (very different) cases in which a file can not be read all at once
  - If a file is larger than the computer has ram, trying to read the file all at once could potentially cause a crash or some other error.
  - If a file is being updated continuously, it is pointless to try to read it in all at once as you won't receive any updates that the file itself receives. You must incrementally read in data to keep it updated.

Testing

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

- I will test to make sure that my code properly handles errors. I will sample invalid inputs and make sure that the code returns a non zero error code when supposed to and that it exits cleanly.
- I will compare outputs with the already built in xxd program to make sure that the outputs match.
- I will give my finished program files of various sizes to read to make sure that it can read all of them without error. I will for sure have files that are smaller and larger than the buffer itself to see how the program deals with those.

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 the program, simply enter xd and then the file name you wish to be read in. If you do not provide a file name, stdin will be read instead. Please do not give any more parameters, as that will lead to an error. Please provide a valid file name.

The program will output the index of each byte, alongside its respective hexadecimal and ASCII 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 organization of this program is very straightforward. The program is a single function stored inside file **xd.c**.

[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.

[2] This is my footnote.

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.

All the functios will be inside **xd.c**.

First all the files will be read and requested using read() and open(). The reader will continuously check to make sure if it has reached end of file. If it has, stop and return the requested data in the buffer. It will also return the requetsed data in the buffer if it is is full.

Next the output will be properly formatted using the data we obtained. There will be a for loop that will iterate through each 8 byte interval and output the index, hexadecimal, and ascii representations of the respective 8 byte interval. Each character will go through an if statement to test if it an ASCII between 32 and 126. If not, it will be printed as a "." in the ASCII representation.

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.

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.**

As the program's function and purpose is relatively simple, all of the code will be stored in one file named **xd.c**. The file will contain a single function which will mimick the behavior of the xxd program. It will take in one argument which is the file name. If no file name is provided, it will take the stdin file as the input instead.

It will use buffered reading to repeatedly read in files until the entire file has been read. The buffer will be 16 bytes at a time.

It will then output each 8th index, alongside its hexadecimal and ascii representation.

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.