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 read characters and convert them into hex characters. It buffers in 16 bytes, grouping 4 and 8 characters together. Each group represents 2 of the original characters inputed.

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 temporarily stores an input or output. You could use a buffer when trying to take multiple inputs to store the first input while the second value is being entered.

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

  Read returns the number of bytes read. You input a file name to find the number of bytes.

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

  A file no. is a unique number given to each file to be able to distinguish which file is which. The file no. of 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?

  Read(0,16) will return 16 when there are 16 inputs in the buffer to be read. It will not return 16 when there are less than 16 arguments.

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

If the file size is bigger than the amount of memory available the file can not be read all at once. Also if the files is updated or changed while it is being read it may not be read all at once.

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 make a test that checks if the user puts in more than one argument. I will also test that the user provides a valid filename.

[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.

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 use this program you must first compile by using

```
make
```

then to run the program use

```
./xd
```

to read an input or

```
./xd 'filename'
```

to read a text file

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].

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.

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.

My Buffer reader function will look like this:

```
While reader (read())

if readbytes == buffersize
print buffer and index


change asciis between 27 and 127 to '.'
print changed input

reset readbytes
increase index
```

[2]This is my footnote.

My main function will look like this:

```
Main (arg1, arg2)

create variables:
reader
readbytes
index
buffer[buffersize]


if arg1 == 2
open arg2[c] in read



run buffer reader function

if readbytes > 0
print buffer for readbytes
add spaces once reader is full
change nonAsciis to '.'
putchar for readbytes


close file
return 0
```

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.**

The buffer reader function needs the files, buffer, buffer size, and past read bytes as inputs. It outputs the buffer and the index. The function uses the read function in a while loop that is updated based on past data. It checks if the buffer is full and once it is it prints the buffer and index.

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.