Assignment 7 – XXD

Johann Lee

CSE 13S – Fall 2023

Purpose and How to use.

The purpose of this project is to read a series of characters and translate them into hex characters. This is done in buffers of 16 bytes with groups of 8, 4 character groupings. Each grouping represents 2 characters from the original input. Compiling the program uses "make" which will create the "xd" binary. Run this binary using either "./xd" or "./xd 'filename.txt'". The former takes in input from standard input and the latter reads from a text file. The output should look something like below.

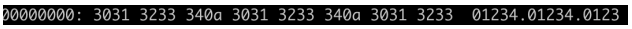

Figure 1: Here is an example output. The first part represents how many bytes into the input the code has read. With next 8 groups are as mentioned before the input characters. Next is the line of characters that were translated for that specific group of 16 bytes.

File read will go through the entire file and exit by itself. Stdin on the other hand will require the user to exit. This is done by using "ctrl d". The program will print whatever is left in the buffer and close itself.

Program Design

My buffered reader works by using the read function within a while loop which is constantly updated based on past read data. This will be further elaborated in the pseudo-code. Within this while loop will be a check for if the buffer is full and if so a print loop will occur. This for the most part is how my buffered reader works.

```
while reader = read(file, buffer + past read bytes, buffer size - past read bytes)
    update readbytes =+ reader

    //check full buffer
    if readbytes == buffersize
        print buffer and index

        print original input and change \n to .

        print newline
        reset readbytes
        increase index
```

My main function does not elaborate that much more on this buffered reader.

```
main(argc, argv)
    int fd
    char buffer[buffersize]
    ssize reader
```

```
        size readbytes
        int index

        if argc == 2
            open from argv[1] in read mode

        buffer reader //shown above in verbatim

        if readBytes > 0
            print out rest of buffer
            print out original input

        close file
        return 0
```

Result

References