Assignment 7 – XXD

Vybavnag Kandasamy

CSE 13S – Fall 2023

Purpose

The program is named xd, is designed to display binary files in a human-readable hexadecimal format. This functionality is essential in scenarios where understanding the content of binary files is necessary, and a straightforward, readable format is required. In its simplest form, xd converts the binary data into hexadecimal representation, making it easier for users to interpret the content of these files.

How to Use the Program

To use the xd program, the user should first compile it using the provided makefile. Once compiled, xd can be run from the command line. If no arguments are provided, xd reads from the standard input (stdin) and outputs to the standard output (stdout). If a filename is supplied as an argument, xd reads from that file.

to run use the code provided below after ensuring all required files, xd.c, bad_xd.c and makefile are present in working directory.

```
make

./xd inputfile.txt or ./bad_xd inputfile.txt

make clean
```

Program Design

The xd program is structured around key components for processing binary files. The main data structure used is a buffer that reads and stores 16 bytes at a time. This buffer is crucial for converting binary data into hexadecimal format efficiently. The primary algorithm involves reading binary data, converting it to hexadecimal, and formatting the output appropriately. The code is organized to facilitate maintenance and future modifications, ensuring that someone familiar with the program can easily understand and modify its components.

Algorithms

pusdeocode for xd.c:

```
Include Standard IO Library

Function printHexAndAscii(buffer, bytesRead, offset):
    Print offset in hexadecimal
    Loop over each byte in buffer:
        Print byte in hexadecimal
    Print ASCII representation of bytes
End Function

Main Function:
```

```
    If incorrect number of arguments:
        Print usage instructions and exit

    Open file in binary mode
    If file opening fails, show error and exit

    While there's data to read:
        Read data into buffer
        Call printHexAndAscii with buffer, bytes read, and current offset
        Update offset

    Close file
End Main Function
```

Results

need to impliment program for this part. 1.

```
veenstra@arm128:~/s23/13s-cse/resources/asgn1$ ./pig_arm
Number of players (2 to 10)? 0
Invalid number of players. Using 2 instead.
Random-number seed? 3
Margaret Hamilton
 rolls 15, has 15
 rolls 5, has 20
 rolls 0, has 20
Katherine Johnson
 rolls 0, has 0
Margaret Hamilton
 rolls 5, has 25
 rolls 0, has 25
Katherine Johnson
 rolls 0, has 0
```

Figure 1: Screenshot of the program running.

References