

CS530, Spring 2020, Program Assignment #1

Tue, 28 Jan 2020

You shall develop, test, and deliver a program which will print contents of a regular file in hexadecimal or binary format. The program shall be named dbx.

The basic hexdump program will open and read every byte in the file specified by the user when launching the program and write both the ASCII hex value for that byte as well as it's printable (human-readable) character (characters, digits, symbols) to standard output. For bytes forming non-printable characters, print a '.' character (a dot/period character, i.e. hex value 2E).

The output should be formatted:

```
<address>: <B0B1> <B2B3> <B4B5> <B6B7> <B8B9> <BABB> <BCBD> <BEBF> 123456789ABCDEF
-- Where B0B1 means Byte #0, Byte #1, etc; the 123456789ABCDEF is the
char/digit/symbol in human readable form
```

Example hex output:

```
0003540: 0504 0675 6e73 6967 6e65 6420 6368 6172  ...unsigned char
0003550: 0008 0107 0000 0131 0675 6e73 6967 6e65  ....1.unsigne
```

Your program shall also provide an option to print in binary instead of hex. If the binary option is selected, then the output format shall be:

```
<address> bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb 123456
-- where b means bit; each line has 6 octets with a space in between each octet
```

e.g. (output format fragment sample):

```
0003540: 00000101 00000100 00000110 01110101 01101110 01110011  ...uns
0003546: 01101001 01100111 01101110 01100101 01100100 00100000  igned
```

Create directory ~/a1 by hand on edoras in your class account. Then create C/C++ source file(s), an include file, a Makefile, and a README file, put them into ~/a1.

Note, your hexdump program will only open regular file type (binary or text/ASCII content), you are not required to handle other file types or errors encountered if the user attempted to run your hexdump program on other file types (dir, special, link, pipe).

Suggestion - become familiar with the xxd program on edoras; your simple hexdumper program is similar to running xxd with no options or with the 'b' (binary) option.

Running your dbx program: You will build your program such that the xbd program will be launched as follows from the shell's command line:

```
edoras% dbx <filename>
edoras% dbx -b <filename>
```

the first example will cause your dbx program to dump (print) the file specified by <filename> in ASCII/hex format, the second example will cause your dbx program to dump (print) the file specified by <filename> in binary format.

ADDITIONAL REQUIREMENTS:

README file - you shall create a README file; consult the instructions for README file content on the course Blackboard. Also, your source files SHALL CONTAIN sufficient comments for making the source easy to read. Points will be taken off for poorly (or non) commented source or inadequate README file documentation.

Compiler and Make - You shall use C/C++ (gcc/g++) compiler and also make to compile your program for this assignment; you will need to create a Makefile for your project, consult the example Makefile(s) on the course Blackboard. Name the executable, 'dbx' (dumper for basic hex).

Project teams. You are encouraged to work in teams of two. Descriptions of pair programming and other partnering techniques are available on the course Blackboard. If you cannot work in teams of two, you may work individually but no extra credit will be given.

TURNING IN YOUR WORK:

The assignment is due at 1700, Monday, 17 February 2020

When ready to turn this in, make sure your files are ready for testing on edoras and turn-in your README file using Blackboard Assignment turnin feature..upload your README file, source code file(s), and any additional files to be graded for this assignment.