Daniel Cender

Feb 2, 2020

CST-221 O500

Wk 4: Memory Mangement: Decimal Number Conversion Program

**Summary**

This program takes in an integer in decimal (base-10) format from the user, runs some bit transformations on it, and displays it to the user in three different formats (binary, decimal, and hexadecimal).

**Result**

The result appears to match what was requested by the assignment specification. The only part that may be incorrectly interpreted is the number of characters shown in the hexadecimal representation of the user’s input.

A screenshot of a computer

Description automatically generated

**Code**

Even though the entire code solution is below, it may also be found uploaded to my GitHub repository for this course, here: <https://github.com/DanielCender/CST-221/tree/master/Wk4/MemoryManagement>

The below code was compiled from the command line and run with the following commands:

* ‘*gcc -o MenMan MemoryManagement.c,*  then
* ‘./*MenMan`*

/\*\*

*\**

*\* Author: Daniel cender*

*\* Date: 02/02/2020*

*\* Description: This program takes a decimal integer from the user,*

*\* then performs multiple bitwise operations to transform the int,*

*\* before displaying the result back to the user.*

*\**

\*/

#*include* <*stdint.h*>

#*include* <*stdio.h*>

#*include* <*stdlib.h*>

#*include* <*string.h*>

/\*\*

*\* Function utility inspired by people much more experienced than myself:*

*\* https://www.programmingsimplified.com/c/source-code/c-program-convert-decimal-to-binary*

*\* Description: This utility takes an integer as input.*

*\* It then creates iterates through the 31 usable bits of the integer,*

*\* shifting to that bit on the input variable, checking if it's 0 or 1,*

*\* then adding it to the nth place in the string binary representation.*

\*/

*char\** decimal\_to\_binary(*int* *n*)

{

*char\** returning = malloc(32 *+* 1);

*int* count = 0;

*if* (returning == *NULL*)

exit(EXIT\_FAILURE);

*for* (*int* idx = 32; idx >= 0; idx*--*) {

*int* current = (n >> idx) *&* 1; // *looks at nth bit to check it*

*if* (current == 1) {

*\**(returning *+* count) = '*1*';

} *else* {

*\**(returning *+* count) = '*0*';

}

count++;

}

*return* returning;

}

/\* *\**

*\**

*\* A funciton almost implemented and test entirely to return a hexadecimal number*

*\* for a real decimal integer.*

*\**

\*/

*char\** decimal\_to\_hex(*int* *n*)

{

*char\** accum = malloc(*sizeof*(*char*) *\** 10); // *need 0x00000000 maximum*

*if* (accum == *NULL*) {

exit(EXIT\_FAILURE);

}

*\**(accum) = '*0*';

*\**(accum *+* 1) = '*x*';

*for* (*int* idx = n; idx < n; idx*--*) {

*int* current = n;

printf("*Number in current placeholder: %i*", current */* 16);

printf("*Quotient is: %i*", current *|* 16);

// *TODO: This special utility was never finished...*

}

}

*int* main()

{

*int* input;

printf("*Hello, please input a base-10 number between 0 and 4,095, inclusive:\n*");

scanf("*%i*", *&*input);

printf("*\nYou entered: %i\n*", input);

*if* (input < 0 || input > 4096) {

printf("*The number you entered is out of the allowed range! Exiting now!\n*");

*return* 1;

}

printf("*Binary version: %s\n*", decimal\_to\_binary(input));

printf("*Hexadecimal input format: 0x%X\n*", input);

input = input << 16;

printf("*Input, shifted 16 bits to the left: %s\n*", decimal\_to\_binary(input));

// *TODO: then mask out (AND) the bottom 16 bits*

// *int masked = shiftedLeft & 0xffff0000; // will leave top bits untouched and kill all lower*

// *input &= ~0xFFFF;*

// *input &= 0x0F;*

input &= *0x*00001111;

printf("*Input, with last 16 bits masked out: %s\n*", decimal\_to\_binary(input));

input |= *0x*07FF;

printf("*Input, after 'or-ing' with 0x07FF: %i\n*", input);

printf("*Resulting number, decimal: %i\n*", input);

printf("*Resulting number, binary: %s\n*", decimal\_to\_binary(input));

printf("*Resulting number, hexadecimal: %X\n*", input);

*return* 0;

}

Resources

C File Handling. (n.d.). Retrieved January 25, 2020, from <https://www.programiz.com/c-programming/c-file-input-output#opening>

How to C in 2016. (n.d.). Retrieved January 23, 2020, from https://matt.sh/howto-c

Tanenbaum, A. S., & Bos, H. (2017). *Modern operating systems*. Vancouver, B.C.: Langara College.