Homework Assignment #3, CS U380 - Spring, 09

Casey, adapted from: Isailovic

Due date: Turn in solutions to the homework exercises listed below by Wednesday, Apr. 8.

0.1 Background reading
P&H 2.5

0.2 Short exercises
0.2.1 P&H

0.2.2 Rotating the bits in a word

Place the code for this exercise in a file named RightRotate.c. In the /course/cs380jc/./www/hw3 directory, there is a file testrot.c, which contains a main() that will test and exercise your function. Compile both files with the command

gcc -g -Wall -o testrot testrot.c RightRotate.c

Write and test a C function called RightRotate whose prototype is

unsigned int RightRotate(unsigned int wordValue, int n);

This function returns the (unsigned int) result of rotating wordValue to the right by n bit positions. Assume n is between 0 and 31, inclusive. Examples:

<table>
<thead>
<tr>
<th>Call</th>
<th>unsigned int wordValue</th>
<th>int n</th>
<th>returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RightRotate( 0x00000011 , 4 )</td>
<td>0x10000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RightRotate( 85, 3 )</td>
<td>0xa000000a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/* 1010101 is 85 in binary */</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RightRotate( -2, 1 )</td>
<td>2^31 - 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0.3 Longer exercise: MIPS disassembler

Your code to this exercise will go into a file named dis.c. When everything works, show us in a script named “disscript.”

Write in C a disassembler for MIPS machine language. That is, your program should read a file containing binary MIPS machine instructions and should produce a listing in MIPS assembly language, looking somewhat like what xspim shows you on the screen when you step through the instructions in your program. It’s probably a good idea to copy all the files in the directory /course/cs380jc/./www/hw3 To compile the program file once you’ve copied them, use the command “gcc -g -Wall -o dis dis.c”
Output format: The format of your program’s output should be one line of text per instruction word of input. Each line has four fields.

ADDRESS  RAW-INSN  OP  A1, A2, A3

The OP is the MIPS assembly-language instruction this machine-language word corresponds to, in all lowercase letters. Please use only the standard names from the inside back cover of H&P.

The other fields- A1, A2, and A3 - vary, depending on which instruction you are disassembling.

The current version of sim.c produces the first two fields, followed by the word “unimplemented”, and an output is in file hw3/studentdis.txt. Your version should produce something that looks like hw3/solvedis.txt

Getting started

Make a directory that you will use for doing this homework, and copy everything in the hw3 directory into it.