CMPE 310 Assembly Language Project IV Apr. 2, 2004

Assembly Project for CMPE 310

Assigned: Fri., Apr. 2nd.
Due: Mon., Apr 19th.

Project Description:
Write an 80x86 assembly program using nasm that performs the following functions:

- Converts the time domain values in a floating point array similar to the one you created in Project III into a frequency domain representation using a discrete fourier transform (DFT). The C code version of two routines, a DFT routine and one that converts from rectangular coordinates to polar coordinates is given on the webpage. You need to write the nasm code that is equivalent to this code. The theoretical aspects of the DFT will be discussed in class on Friday.
- The input to your program (to be read as parameters from the command line) is the file name of the time domain data (as you've done for Project III) plus the following in this order: the fundamental frequency to be used by the transform (given as a floating point number) and the number of harmonics (given as an integer). So a valid call to your assembled code looks like:

  ```
  calc_dft y_values_file_name dft_fund_freq dft_num_freqs
  ```
- Example input and output files are provided on my webpage. Here is the example call for the input file given on the webpage

  ```
  calc_dft proj4_input.txt 5.0E-02 25
  ```
- Your program will compute the DFT, convert to polar form and print out the frequency, magnitude and phase in three columns (see the output file given) as floating point numbers for each frequency (starting with the DC component). You may use printf to print this table if you wish.
- Extra credit (5 points) will be given for code that does not use any C library functions except atan2 and printf.

You must use the submit program to submit your code. The class name is cmpe310 and the project name is proj4. Submit the project (project4.asm) file, any code that you use from our examples should be in (common_code.asm). You are also required to turn in a hardcopy and a write-up as described in project 1. The breakdown of the points are as follows:

- Correctness 80%
- Modularity 10%
- Documentation (description, etc.) and code comments 10%

You can construct your own data files for this in the format described above. We will test your code on our own examples. The submitted program is before you come to class on Monday. You must turn in the hardcopy during class on Monday and it must be identical to the code that you submitted.

THE LABS ARE INDIVIDUAL EFFORTS: INSTANCES OF CHEATING WILL RESULT IN YOU FAILING THE COURSE.