Lab #9: The Right to Assemble! Name_____

All Source Codes and answer sheet must be uploaded in your CMPE 310 BOX I created for you. AND DEMO to TA before leaving lab.

Pay careful attention to discussion. The objective of this project is to walk you through the process of typical maintenance tasks using a simple assembly language program to give you a sense of what we'll be doing in future projects.

We keep this program so simple so that you don't even have to know anything about assembly language to follow what's happening. But in following our steps through this process, you'll get a sense of what's involved in working with assembly programs.

You should complete this lab individually. When you have completed the assignment, raised your hand to get TA's attention, so that you can demonstrate your program to the TA and get your lab signed off before leaving class.

Reading/reference material

- NASM tutorial

Concepts

Illustrate the importance developing skills not just in writing new code, but also in working with existing code.

Problem: Your task is to enter the "Hello World!" application program below into an editor and save it as HelloWorld. And then modify it to output "Hello Universe!" instead.

```
section .data                           ;section declaration

msg     db      "Hello, world!",0xA     ;our string
len     equ     $ - msg                 ;length of our string

section .text                           ;section declaration

global _start ;we must export the entry point to the ELF linker or
loader. They conventionally recognize _start as their entry point. Use
ld -e foo to override the default.

_start:

;write our string to stdout (monitor)

 mov     eax,4   ;system call number (sys_write)
 mov     ebx,1   ;first argument: file handle (stdout)
 mov     ecx,mag ;second argument: pointer to message to write
 mov     edx,len ;third argument: message length
 int     0x80    ;call kernel and exit

;final exit
 mov     eax,1   ;system call number (sys_exit)
 xor     ebx,ebx ;first syscall argument: exit code
 int     0x80    ;call kernel to take over
```

Looking at this code, you can immediately see the message is enclosed in quotes. Your first instinct is to just change the word "World" to "Universe". Even without knowing assembly language you are already programming! In a simple case like this, that strategy is okay; with a larger application, however, such an approach may only lead to the creation of more problems. So what do we need to know?

Answer the following questions:
With any maintenance effort, we must do the following:

1. Observe and record the behavior of the current application by saving and running the code.
    - If it isn't working in the first place, you may think that your changes are the sources of errors and spend many hours in a futile attempt to fix the program by correcting your changes.
    - Run the code. Did it work? If not, what's wrong with it? _____

    - Every programmer gets these kinds of errors at some time. Very few of us are perfect typist!
    - Can you fix it? _____

    - What is the output on the monitor? _____

2. Once you are sure the program works as claimed, then go ahead and make your changes with the knowledge that any new problems are associated with our modifications.

    - How many places was the word "Universe" substituted for the word "World"?

3. Provide a screen shot of the terminal window upon running your executable file for "**Hello Universe**!"

4. Is that it? Could it really be that simple as changing a word? _____

5. Well, actually no. We edited our existing Hello World code file, and when we saved it, we overwrote the existing application, so we no longer have the original code.
    - It is always a good idea to keep the working version of a program separate from a new version as we discussed in class.
    - So to correct this situation, we really should have created a new project called Hello Universe and copied the code from our existing project into the new one and save changes without affecting the original.

Software Maintenance Tips

1. Check that the existing code works as claimed.
2. Make changes to a *copy* of the existing code.
3. A maintenance task isn't necessary complete once the functionality has been achieved. Comment your code neatly for the next programmer who has to do maintenance (for now the TA).
4. Keep backup copies of your code when developing new programs.

******All Source Code and Screenshot must be uploaded in your CMPE 310 BOX I created for you**.***