CS3843 Programming Assignment #3: What does determineGrade.s do? (50 points)

This program is our first assignment to aid in learning IA32 assembly language. Primarily, you will determine what determineGrade.s does. Additionally, you will use hexDump and dumpRegs functions (that I am providing). **Please do all your work on an hen server**. You would get different results on the fox server or possibly other machines. Also, my cs3843p3Driver.o, dumpRegs.o and hexDump.o will probably not work on the fox servers since they use 4-byte addressing.

You will invoke the program using:

./p3 <p3Input.txt

**I have provided the following files in /usr/local/courses/clark/cs3843/Pgm3Hen:**

* cs3843p3Driver.o - the driver which reads an input file from stdin, prints the student Id and grades, invokes determineGrade, and prints total grade.
* determineGrade.s - an assembler function which determines the grade. You will be modifying and documenting this function. After you copy this to your directory, you might need to   
   chmod +w determineGrade.s  
  Note that the locals are

iqTot

iTot

* dumpRegs.o - dumpRegs prints the contents of the 8 major registers to stdout. It does not receive any parameters. You should use dumpRegs to see what is happening at different points in the IA32 code.
* hexDump.o - a hexDump function (as described in assignment #1). It also prints the address being dumped. If you use %esp as the address, you can print the stack. If you use %ebp, you can see what is above the current activation record. Note that anything uninitialized will show garbage values.
* cs3843p3Input.txt - input file for the program containing some student data.

**In this program, you will do the following:**

1. Determine what each instruction of an assembly function, determineGrade.s, is doing and document it. Modify the function header comment and add detailed comments within the assembler code explaining what each instruction is doing. Assume it is passed 10 parameters named **iG1** thru **iG10.** You are not required to document the dot assembler directives.
2. Immediately after the following line of code, **call dumpRegs**:  
    pushl %ebx # Save the caller's %ebx since we are using %ebx
3. Also, immediately after the following line of code**, call dumpRegs**:

subl $20, %esp

1. Immediately after that second call of dumpRegs, you need to set up the parameters for hexDump and **call hexDump**. The call of hexDump requires parameters as described in assignment #1. Pass it 64 for the buffer length and 12 for the bytes per line. At this point, use %ebp for the buffer address. Note: you may need to change how much memory was allocated/deallocated on the stack since hexDump is passed parameters.



1. Answer the **questions from below** (worth 20 of the 50 points)



**Assembling and linking**

1. When you want to assemble a file, use:

gcc -c *filename*.s

1. When you want to link your code to produce the p3 executable, use:

gcc -o p3 cs3843p3Driver.o determineGrade.o dumpRegs.o hexDump.o

1. To execute it, use:

./p3 < p3Input.txt

**What to submit in BlackBoard (in *abc123.zip):***

1. Your fully documented determineGrade.s. Please refer to the Assembler documentation standards which is on my website under Programming Assignments.
2. Your output which also invokes dumpRegs (twice) and hexDump as described above.
3. Your answers to the questions below.

**Questions to answer**

In addition to what is described above, also turn in answers to the following questions:

(2 pts) 1. How do you get the current address of the stack?

It will be placed in %ebp

(2 pts) 2. What affects the amount of memory that is needed from the runtime memory stack for a function?

Parameters, automatic locals

(2 pts) 3. When **passing parameters** to a function from the **caller**, how are they passed? (This is asking for the caller's perspective.)

Hint: the first parameter is passed by storing it at 0(%esp).

Save the values for registers %eax %edx and $ecx if it's needed after the call

Put each parameter onto stack in order from right to left

Call function using call

(1 pts) 4. Suppose an address is in %ebx. How do we pass that address as the first parameter to a function?

movl 8(%ebx),%eax

movl %eax,(%esp)

(1 pts) 5. Suppose an address is in %ebx. How do we pass the value at that address as the first parameter to a function?

movl 8(%ebp),%ebx

(12 pts) 6. What is the algorithm used by determineGrade to determine the grade? Your description should either use pseudo code or C.

iQtot = 0

itot = iG1 + iG2 + iG3

if(g4 <= g5) jump to .L2

else keep going down and jump to .L3

if(g6 <= g7) jump to .L4 and compute

else keep going down and jump to .L5

if(g8 != 0) jump to .L6

else keep going down and set value of iG8 = iG10

.L6

If(g9 !=0) jump to .L7

else keep going down and set value of iG9 = iG10

.L7

arthmetic shift value of G9 by 1 bits

multiply value of G10 by 3

iTOT = iTOT + iQtot + iG8 + result of arithmetic shift of G9 + G10 \* 3

return iTOT