Assignment 2: IA32 assembly programming

Due: 11:59PM Thursday, November 3, 2022

Grader (TA): Ruxin Wang <rwang31@lsu.edu>

In this assignment, you will write a basic IA32 assembly routine using gnu assembly. First, create the directory **prog2**, which will be used for submission. Change to this directory, and issue the following command to get the files you need (Note: Do not ignore the period at the end of the command):

```
cp ~cs3501_lee/cs3501_F22/p2/* .
```

View and understand the provided `Makefile` and `main.c` so that you can write your own in the future.

In this assignment, you will write an **assembly program** in the provided file `prog2.s`. The file will implement a function with the prototype:

```
int prog2(int i, int j, int *k, int a[5], int *l);
```

This function will do three things:

1. Return `j - i + 2`
2. Set `*k = 6 * (*k);`
    - It is NOT allowed to use any multiplication or division instructions.
3. Set `*l = a[0] + a[1] + a[2] + a[3] + a[4];`
    - You are not required to use conditional jumping for this task, but using it is strongly recommended.
    - Note that, when you modify any callee-save register (`%ebx, %esi, or %edi`), you need to save and restore its old value.

Note: You are required to write a **comment** for **each instruction** to explain its purpose. You can write a comment by beginning with the "**#**" sign in your assembly code. In addition, you should include your **full name** and **LSU ID** number as a comment in your code.

I suggest you get this working in three distinct steps:

1. First, write a function taking the above arguments, but the only thing it does is return `j - i + 2`. You can test this function by running the provided `xtest` program. (Hint: the return value should be stored in `%eax`.)
2. Second, try to use the address passed in as the `k` parameter to change the caller's variable. You need to get the integer stored at the address, multiply it by 6 (without using any multiplication or division instruction), and then store the computed value at the address.

3. Third, try to use the address passed in as the `a` parameter to read integers stored in the array. Note that the address is the starting address of the array, and you need to calculate an appropriate offset to read each array element.

⇨ You can hand in with '**`make submit`**'.

Here is an example run of my completed tester:

```
>make
gcc -Wall -g -m32 -c main.c
gcc -Wall -g -m32 -c prog2.s
gcc -Wall -g -m32 -o xtest main.o prog2.o

> ./xtest
j-i+2= 5
k*6= 36
array sum= 18

j-i+2= 8
k*6*6= 216
array sum 2= 36
```

When you are ready to submit, you can do so with "**`make submit`**". Note that you can submit multiple times, but don't do so after the due date! Note that you will be graded using the provided `Makefile`, not your copy of it, so do not count on any changes you make to any file except `prog2.s`.

Important notes:

- You should **NOT** send any assignment-related emails to the instructor. The TA has full control over the assignment grading process. The instructor will NOT reply to assignment-related emails and will NOT answer to assignment-related questions during his office hours. You can still ask high-level concepts.
- You should **NOT** send the TA (or the instructor) any emails enclosing your source code. In fairness to other students, the TA will only check your submitted file(s) after the assignment deadline. If you send any email enclosing your source code, there may be a penalty for your assignment grade. You can still ask high-level concepts.
- If there is no comment in your code, a grade of "zero" will be recorded.
- Your program will be compiled and tested only on the *classes* server for grading. You need to make sure that your code is running well on the server. In other words, even though your program is running well on another machine, that will not be considered during grading.
- Since this assignment is posted about two weeks before its deadline, last-minute requests for extension will not be granted. Being busy can never be a valid reason to request an extension when you have about two weeks for the assignment. The instructor/TA will not reply to such requests.

- It is very likely that last-minute questions sent on the due date cannot be answered by TA before the deadline.

Only for Honors Option

Write an assembly program for the C function below and send your code **to the TA via email**. You need to write a comment for each instruction to explain its purpose. In addition, you need to write your own tester program to show the working of your assembly program.

```
int loop(int n, int *a) {
  int i;
  int x = 0;

  for(i = 0; i < n; i++) {
    if (!(a[i] <= x))
      x = a[i];
    x++;
  }

  return x;
}
```