Arlan Prado

CS 118-02

Lab 3

Purpose:

The purpose of this lab is to get comfortable with arithmetic, registries, and variables in assembly. The code calculates the total amount of expenses and costs for one year using variables and registries.

Process:

The first thing to do is initialize input variables for costs of housing, food and transportation, tuition, and costs of “incidentals”. The only output variable initialized is the calculated total of all the costs. Since there are only number values entering in variables, the datatype I used is .long.

```
#DATA SECTION
#
.data

houseMon: .long 650    #put in 12 times
foodandtrans: .long 150  #put in 52 times
eduperterm: .long 400    #put in 2 times
incid: .long 500       #put in 1 time

#DATA OUTPUTS
total: .long 0
```

To calculate the annual total, each value has to be multiplied for the number of times the costs will have to be paid for the year. The housing costs are multiplied by 12 for each month, food and transportation are paid weekly so it’s multiplied by 52, there are usually 2 education terms so the estimated cost would be multiplied by 2, and total incidentals per year is only entered in once. The annual values of each types of costs

are moved into registries then add all the registries into one main registry then moved into total. I moved each value into separated registries but compiled it all into eax then copied into total. Once that's complete, end the program.

```
.text
.globl _start
_start:

imul $12, houseMon, %eax
imul $52, foodandtrans, %ebx
imul $2, eduperterm, %ecx
movl incid, %edx
add %ebx, %eax
add %ecx, %eax
add %edx, %eax

mov %eax, total

mov $1, %eax
mov $0, %ebx
int $0x80
```

With the comments and code, the total should be $16,900.

```
test.s   apradliving.s
30  movl incid, %edx
31  add %ebx, %eax
32  add %ecx, %eax
33  add %edx, %eax
34
35  mov %eax, total
36
37  mov $1, %eax
38  mov $0, %ebx
39  int $0x80
40
```

```
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from apradliving...done.
(gdb) b 36
Breakpoint 1 at 0x804809a: file apradliving.s, line 36.
(gdb) r
Starting program: /home/debian/lab3/apradliving

Breakpoint 1, _start () at apradliving.s:37
37      mov $1, %eax
(gdb) print total
$1 = 16900
(gdb)
```

Pitfalls:

The most difficult part was wondering how to multiply variables and registries which was solved by a simple google search. The lab was fairly simple but helped get me more comfortable.

Possible Improvements:

An improvement to the completion of this assignment could be a better understanding of the registries so maybe I could make the code more compact somehow. But overall I feel that I did well for what I knew about assembly language/GAS.