Stark Industries – Jarvis Mark I

Assembly Lang Prog – CSCI 4434

Benjamin Placzek 00650430 **Professor Robert Antonetti**
Computer Science
University New Haven
5/11/2021

Problem Statement

Stark Industries have hired me to write the code to implement targeting functions on their Jarvis Mark I Targeting Computer. This code will be used onboard their Battleships allowing for the targeting of six objects simultaneously. For an engagement to be considered successful, the round must be placed within a 30m radius of the targeted Battleship.

Objective

Use ASC II String input data, process the data with the equations of motion correctly, and output a buffer loaded with ASC II String data.

Design Trades (Description of Thought Process)

1. Run through a physics problem by hand, using inputs I can think of
2. Craft assembly modules
    a. Data Parser for input string null terminated
    b. Convert ASCII to float
    c. Corrected bearing aim
    d. Barrel elevation on x axis
    e. Charge needed
    f. Data Loader for output string null terminated
3. Implement modules in a main program, calling as necessary, and debugging procedurally
4. Debug implementation of modules together
5. Compare output with physics problem by hand, fix or cite as open issue
6. Document listings

Design Layout

1. Understand math calculations
2. Create assembly modules in the order listed in **Design Trades**
3. Debug
4. Report on output/flaws

See PhysicsEx.pdf for written out work

Assembly Code Listing

See asm_code file for assembly listing and coding attempts

Descriptions

Detail descriptions of each function:

- stringio:
    - Three attempts are documented in my submitted modules. Neither of them were able to work. I proceeded with this assignment hard-coding the values to be used in performing the calculations.
- asc2float:
    - This function was also not implemented because I could not get any user input for floats in my program.
- main:
    - Calls each other function and returns service calls
- init:
    - Initializes constants and numbers to be used by the program
- in:
    - Gets fake user input from a hardcoded array
- bearing:
    - Attempts calculations for bearing_aim
- elevation:
    - Attempts calculations for elev_aim
- charge:
    - Attempts calculations for M_charge

Test Cases

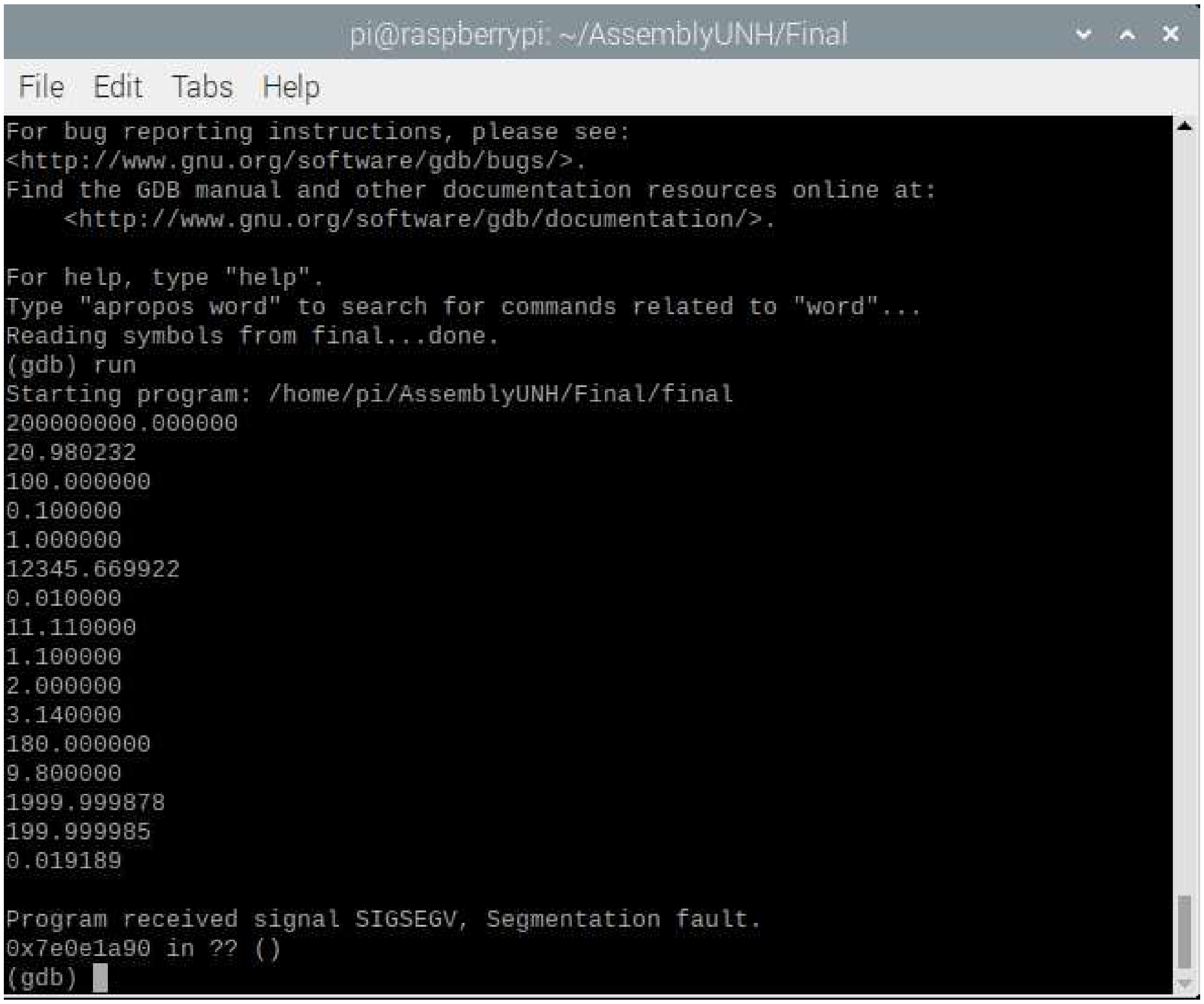
- Upon running my program, several outputs are generated. Each output corresponds to one of the constants or inputs or calculations stored within the program.
- To identify what is what, the list of output is seen in the comments of fprint.s

Open Issues

What is wrong with my program

- Register s1 (lbarrel) is stored as 10.0 but is displayed incorrectly as 20.980
- String input cannot be parsed correctly. Three attempts were had but I could not figure out how to process a string correctly from console input (see stringio 1-3)
- The math is not done. I found that while doing this program that it was extremely difficult for me to calculate trigonometric functions, and I could not find reliable ways to compute them.
- There is no output at the end, because there is no point. I output the values as I work to show my work.

Overall Results



```
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 final...done.
(gdb) run
Starting program: /home/pi/AssemblyUNH/Final/final
200000000.000000
20.980232
100.000000
0.100000
1.000000
12345.669922
0.010000
11.110000
1.100000
2.000000
3.140000
180.000000
9.800000
1999.999878
199.999985
0.019189

Program received signal SIGSEGV, Segmentation fault.
0x7e0e1a90 in ?? ()
(gdb)
```

Results of program run against hand calculated test cases:

- The assembly code written produces accurate results for the data inputted. The values seen above are validated with both a C program and work by hand (PhysicsEx.pdf)

References for Code:

https://www.keil.com/support/man/docs/armasm/armasm_dom1361289988007.htm

https://www.keil.com/support/man/docs/c51/c51_lib_data.htm

https://www.keil.com/support/man/docs/c51/c51_math_h.htm

https://bartaz.github.io/ieee754-visualization/

https://www.h-schmidt.net/FloatConverter/IEEE754.html