TASK 4: Raspberry PI Installation and ARM Assembly Programming

1. **PI**

First we made a repository in github, gitter\_geeks is the name of our team and tat is the repository which we use.

A screenshot of a computer screen

Description automatically generated

Using the steps given to us we generated a key and pasted it in our github account under ssh key. We also made a directory named test.git to save the following files.

A screenshot of a computer

Description automatically generated

Here is how it looks when we check the repository, we see that three new files were created. In this case, it looks like:

A screenshot of a cell phone

Description automatically generated

1. **ARM Assembly Programming**

Part 1 – After making the repository in GitHub for our code and connecting our Raspberry Pi to the repository at the GitHub using Git we started the following program.

A screenshot of a computer screen

Description automatically generated

With the help of the nano text editor by way of the terminal window, we saved the following code in the file.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

After creating the file we assembled the file and then linked the file to get an executable. After executing the file we ran the program. The result was truly expected, we did not see anything in the output because everything in the program takes place in the registry.

After that we begin utilizing the GDB to see the how the program is being executed. We used -g flag to reassemble the program to run GDB.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

We then set breakpoint at line 11 to stop the program preceding to the execution of the respective line. Then we gave GDB the command b 11, and then we ran the program. After that we gave the command run to look at the breakpoint.

A screenshot of a cell phone

Description automatically generated

We used the command “info registers” to see the generated values in the registers before executing line 11, Register 1 has a value of 8 because the register was first set to 5, then it was 4 because 1 was subtracted from it, and then 4 was added to that value, resulting it to 8.

Part 2 - Write a program that calculates the following expression: A = (A + B) − (C \* D), where A=10, B=11, C=7, and D=2

As we had to refer to part 1, we started in the same way, following is the work for this problem;

A screenshot of a computer screen

Description automatically generated

As in Part 1, With the help of the nano text editor by way of the terminal window, we saved the following code in the file and the program looks as shown below. After creating the file, we assembled the file and then linked the file to get an executable. After executing the file, we ran the program. The result was truly expected, we did not see anything in the output because entire program takes place in the registry. Therefore, we will not receive any output. The same as we did earlier.

A screenshot of a computer screen

Description automatically generated

After that we begin utilizing the GDB to see the how our First Program was being executed. We used -g flag to reassemble the program to run GDB, as we did it before.

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

Here we set the breakpoint to line 14, so to debug it through every step before we end the code.

A screenshot of a computer screen

Description automatically generated

So the above registers states that - Registers 2 is 11, 3 is 7, 4 is 2 which is completely expected because we had the assigned values. Register 5 stores the value of 22 - the result of our initial A added to B (10+11). Register 6 stores the value as 14 which is the result of C\*D (7\*2). Register 1, stores the final value of A which is 7 which is the expected result of the given equation.