Manoj Reddy Donthireddy,
2020375.

Assignment -1

Combining c and Assembly Language Program

I have used makefile for organizing code for easy compilation. With makefile, I was able to create multiple targets for compilation which made my task easier and there is no need to recompile once again when we make some changes at each step. We need to make a c program file, asm file, and a makefile to show the steps of compilation. In function A I used scanf to call 64-bit integer value and called C function into it. I have compiled each one of the file which explains the pause and the compilation.
As given the A should call B passing a 64-bit integer as an argument and B should interpret that as a 8-byte ASCII string and print individual characters on screen which gives an idea the assembly language is integrated with the function B that is called by A.
The asm code in B.asm created a section.bss for declaring variables(Here used for bit strings), section.data is used in the code for declaring initialized data or constants exactly here data elements are stored in the memory and the section.text is used for keeping the actual code that started with a global variable which gives an idea that where the program execution begins in kernel. Where I used the extern c in B.asm to tell the compiler that this variable we are declaring was defined elsewhere and extend its functions().

In the asm code 'rdi' we specified to start from array of stored bit integers and the register rax 64 bit address values are returned from the given functions in this register which start from 1 and writes to the whole register.the rsi declared the bit of strings. Then finally caled syscall which fundamental interface between an application and the linux kernel. And called ret to pop up the return address off the stack which is pointed by the stack pointer register and then continues execution the given function.
Here when B executes the ret instruction it jumps to a third function C where any explicit call to function not happened and here A ret from B passed the control to function C instead of accordingly which finally the function c terminated the program by using the exit system call.

Overally the c program inputs the64 bit integers using scanf and c program calls the asm routine B.asm where moves the value from rax to rdi. Each instruction consists of an operation code and each executable instruction generates one machine language instruction . The asm file is created which is of hexadecimal form and has directive global B and section as B.