Setting up and using Bochs with support for GDB

I uninstalled the existing Bochs in my Virtual Linux Machine using this command: 'sudo apt-get remove bochs'. Next, I downloaded the latest version of Bochs (V2.7) from http://sourceforge.net/projects/bochs/files/bochs/2.7/ in a .tar.gz format. I installed these two packages with these commands: 'sudo apt-get install libx11-dev' and 'sudo apt-get install libxrandr-dev'. Then, I extracted the .tar.gz file using this command: 'tar xvzf bochs-2.7.tar.gz'. I opened the bochs-2.7 directory and executed the following commands in the same folder:

- 1. sudo ./configure --enable-gdb-stub --with-x11
- 2. sudo make
- 3. sudo make install

I added the '-g' and '-O0' options to the end of 'GCC_OPTIONS' in the makefile and configured GDB in the Bochs configuration file 'bochsrc.bxrc' by adding this instruction:

'gdbstub: enabled=1, port=1234, text_base=0, data_base=0, bss_base=0'

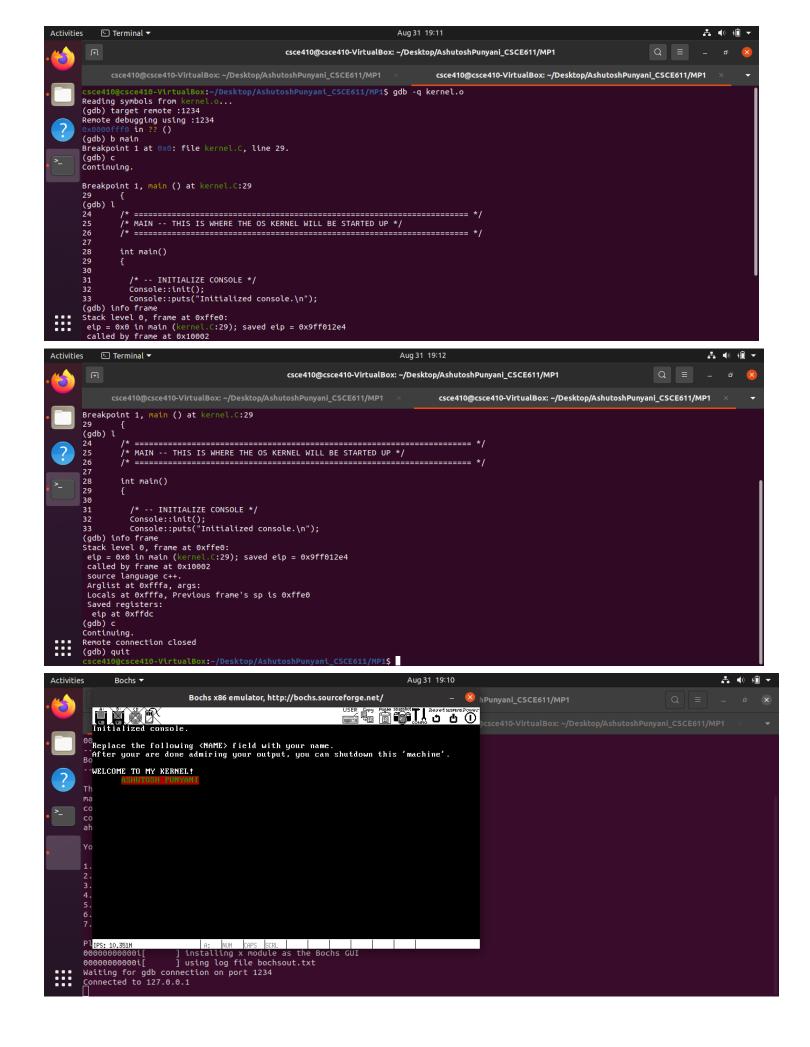
Then, I ran these commands in the project folder:

- 1. sudo make
- 2. sudo ./copykernel.sh
- 3. bochs -f bochsrc.bxrc

Now, on the screen, I was presented with the following options:

- 1. Restore factory default configuration
- 2. Read options from...
- 3. Edit options
- 4. Save options to...
- 5. Restore the Bochs state from...
- 6. Begin simulation
- 7. Quit now

I selected option 6 to start the simulation, which launched the Bochs emulator with a blank screen. In a terminal, I opened the same project folder where I had executed these commands to open GDB 'gdb -q kernel.o'. After GDB started, I entered this command 'target remote:1234'. This command connected to the Bochs terminal. I added a breakpoint using 'b main'. This set a breakpoint in 'kernel.c' at the 'main' function. I continued the function with the 'c' command which reached the breakpoint. Then, I used the 'I' command to print the lines at the breakpoint and 'info frame' to print a verbose description of the selected stack frame. I continued the function again with 'c' which started the Bochs emulator. In the Bochs emulator, I chose 'My Kernel,' which printed the welcome message and my name. Finally, I clicked on the Power icon to close the emulator. In the GDB terminal after clicking on power icon, I saw the message "Remote connection closed."



References:

- 1. https://people.engr.tamu.edu/bettati/Courses/OSProjects/project_overview.html
- 2. https://bochs.sourceforge.io/doc/docbook/user/debugging-with-qdb.html
- 3. https://amelon.org/2018/02/25/integration-bochs.html
- 4. https://people.engr.tamu.edu/bettati/Courses/OSProjects/to-use-gdb-tools.pdf
- 5. https://sourceware.org/qdb/onlinedocs/qdb/Frame-Info.html
- 6. https://sourceware.org/gdb/onlinedocs/gdb/List.html