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***

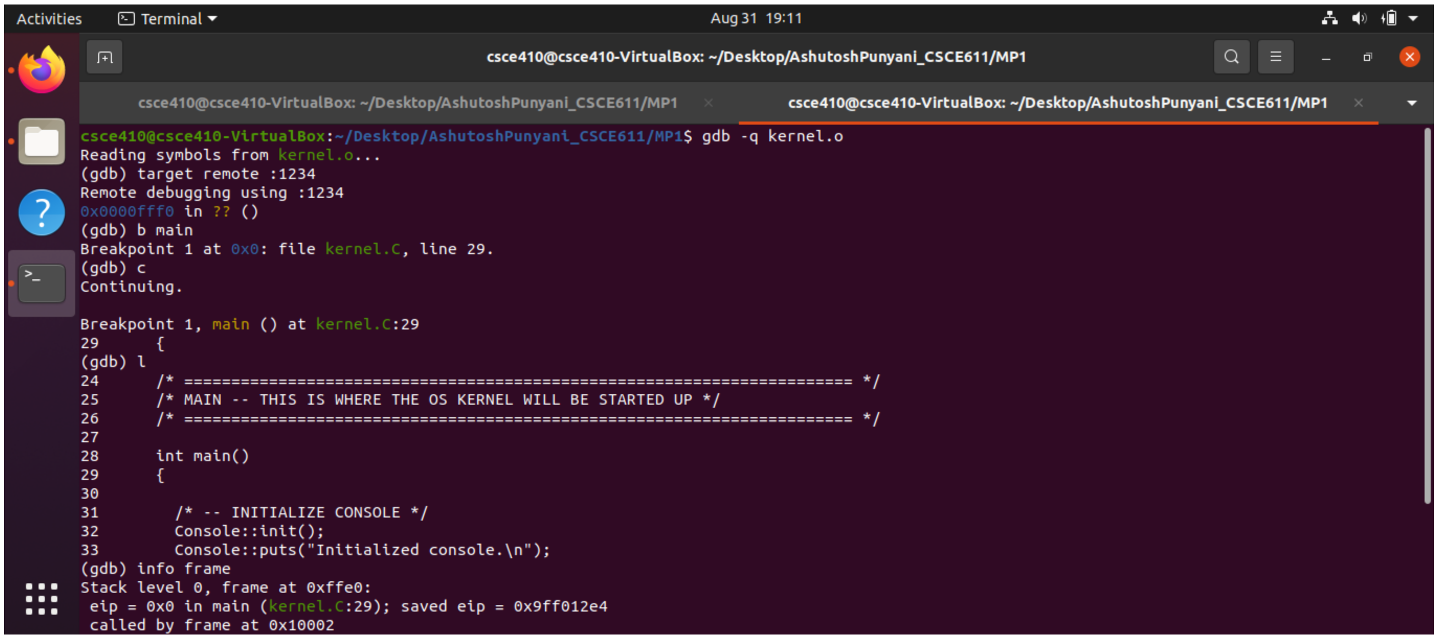
I added the '***-g***' and '***-O0***' options to the end of '***GCC_OPTIONS***' in the makefile and configured GDB in the Bochs configuration file 'bochsrc.bxrc' by adding this instruction:
'***gdbstub: enabled=1, port=1234, text_base=0, data_base=0, bss_base=0***'

Then, I ran these commands in the project folder:
NOTE: If files have already been created using the ***make*** command, we can use ***make clean*** to remove the previously generated files, and then use the ***make*** command again to generate the files.

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

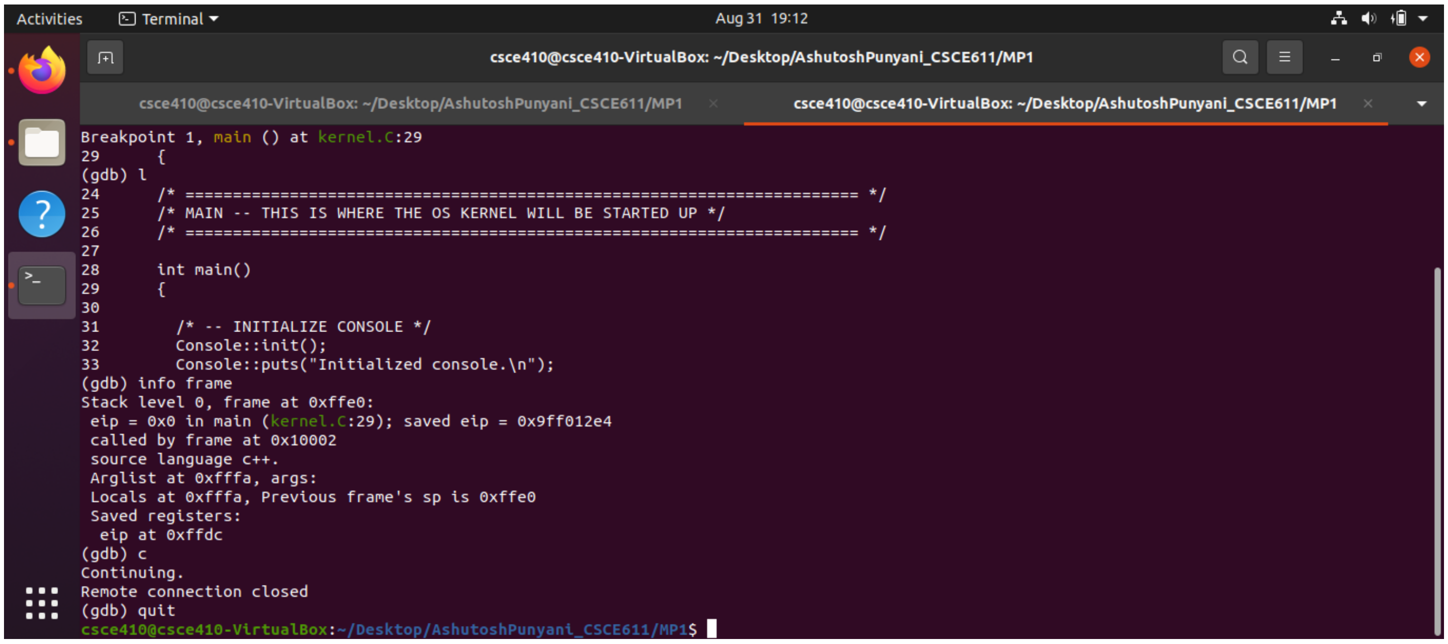
Now, on the screen, I was presented with the following options:
1. Restore factory default configuration
2. Read options from...
3. Edit options
4. Save options to...
5. Restore the Bochs state from...
6. Begin simulation
7. Quit now

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 '***l***' 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."

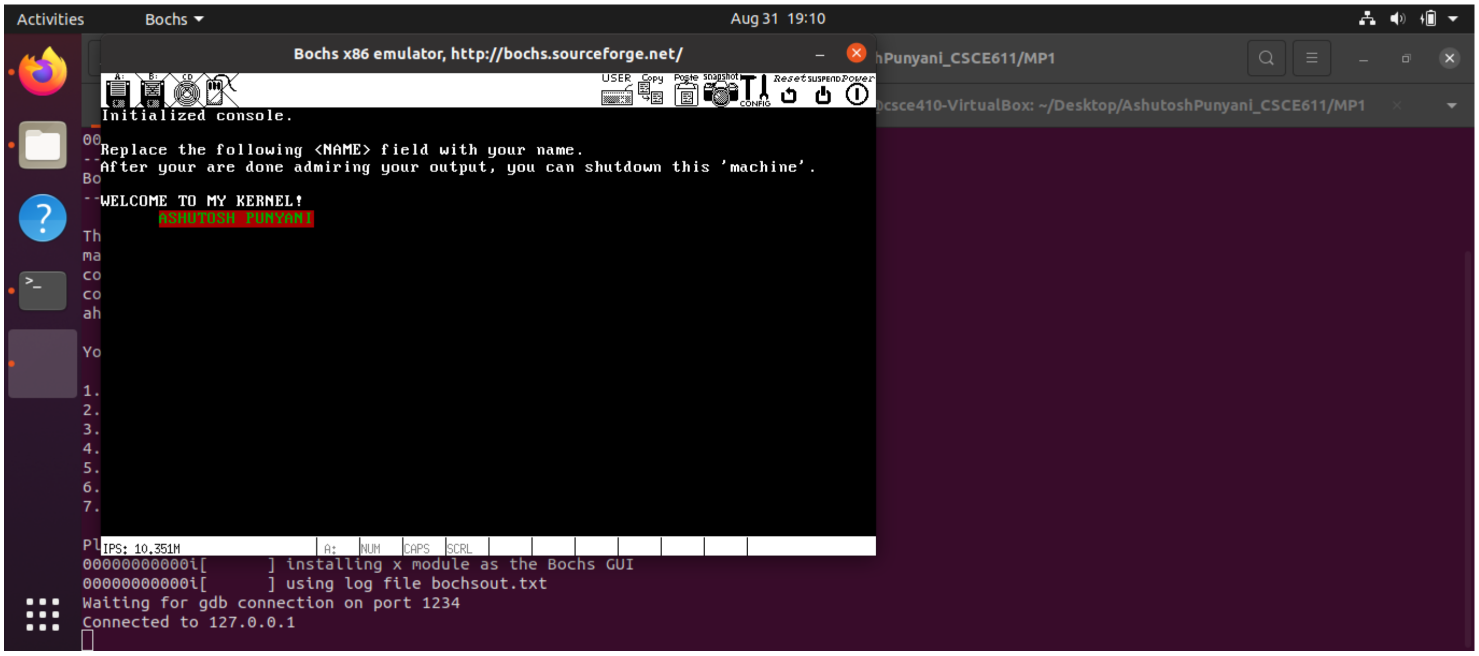


```
csce410@csce410-VirtualBox:~/Desktop/AshutoshPunyani_CSCE611/MP1$ gdb -q kernel.o
Reading symbols from kernel.o...
(gdb) target remote :1234
Remote debugging using :1234
0x0000fff0 in ?? ()
(gdb) b main
Breakpoint 1 at 0x0: file kernel.C, line 29.
(gdb) c
Continuing.

Breakpoint 1, main () at kernel.C:29
29      {
(gdb) l
24      /* ======================================================================= */
25      /* MAIN -- THIS IS WHERE THE OS KERNEL WILL BE STARTED UP */
26      /* ======================================================================= */
27
28      int main()
29      {
30
31        /* -- INITIALIZE CONSOLE */
32        Console::init();
33        Console::puts("Initialized console.\n");
(gdb) info frame
Stack level 0, frame at 0xffe0:
 eip = 0x0 in main (kernel.C:29); saved eip = 0x9ff012e4
 called by frame at 0x10002
```



```
Breakpoint 1, main () at kernel.C:29
29      {
(gdb) l
24      /* ======================================================================= */
25      /* MAIN -- THIS IS WHERE THE OS KERNEL WILL BE STARTED UP */
26      /* ======================================================================= */
27
28      int main()
29      {
30
31        /* -- INITIALIZE CONSOLE */
32        Console::init();
33        Console::puts("Initialized console.\n");
(gdb) info frame
Stack level 0, frame at 0xffe0:
 eip = 0x0 in main (kernel.C:29); saved eip = 0x9ff012e4
 called by frame at 0x10002
 source language c++.
 Arglist at 0xfffa, args:
 Locals at 0xfffa, Previous frame's sp is 0xffe0
 Saved registers:
  eip at 0xffdc
(gdb) c
Continuing.
Remote connection closed
(gdb) quit
csce410@csce410-VirtualBox:~/Desktop/AshutoshPunyani_CSCE611/MP1$ 
```



```
Initialized console.

Replace the following <NAME> field with your name.
After your are done admiring your output, you can shutdown this 'machine'.

WELCOME TO MY KERNEL!
      ASHUTOSH PUNYANI
```

```
00000000000i[      ] installing x module as the Bochs GUI
00000000000i[      ] using log file bochsout.txt
Waiting for gdb connection on port 1234
Connected to 127.0.0.1
```

References:

1. https://people.engr.tamu.edu/bettati/Courses/OSProjects/project_overview.html
2. https://bochs.sourceforge.io/doc/docbook/user/debugging-with-gdb.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/gdb/onlinedocs/gdb/Frame-Info.html
6. https://sourceware.org/gdb/onlinedocs/gdb/List.html