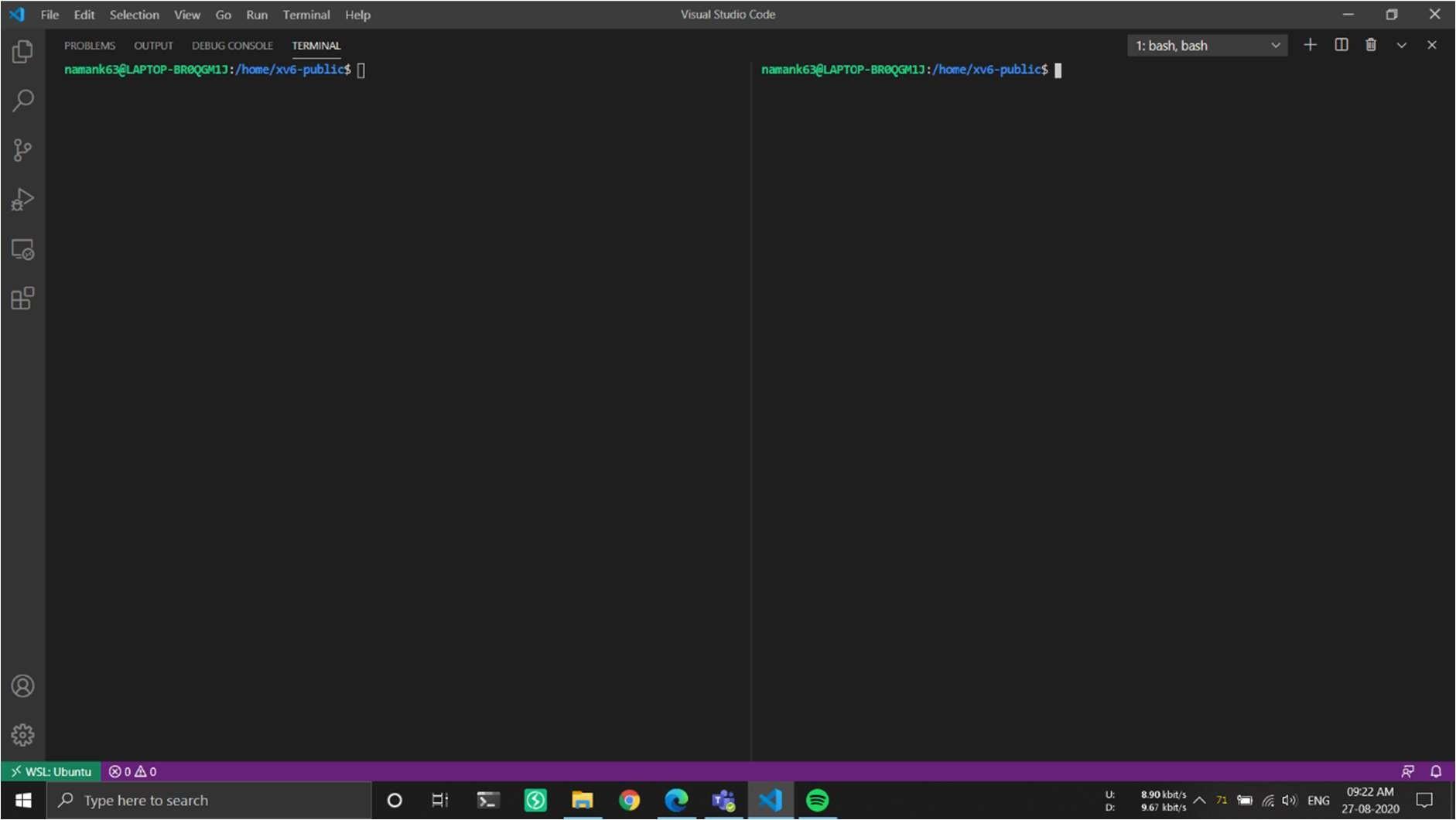
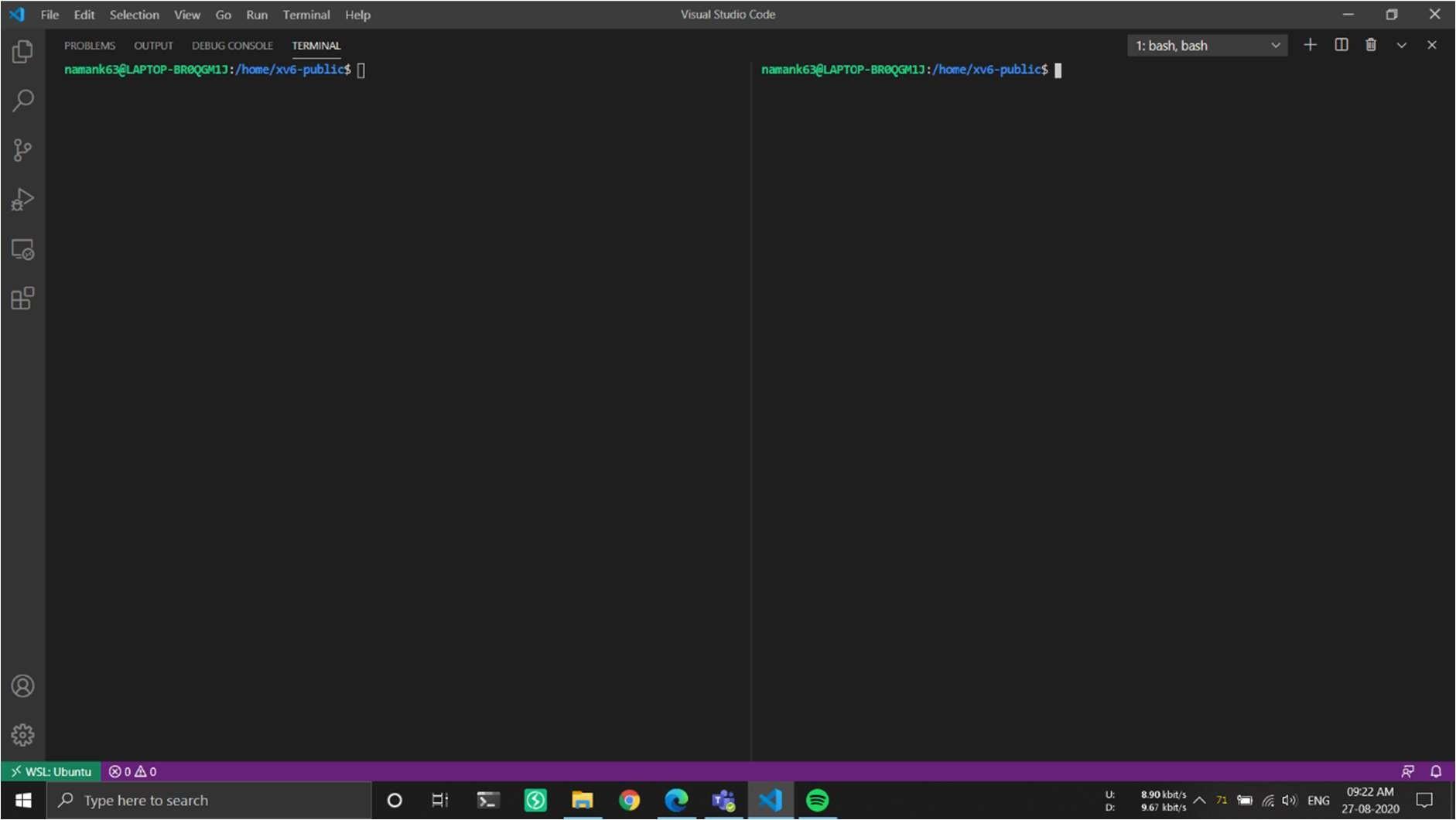
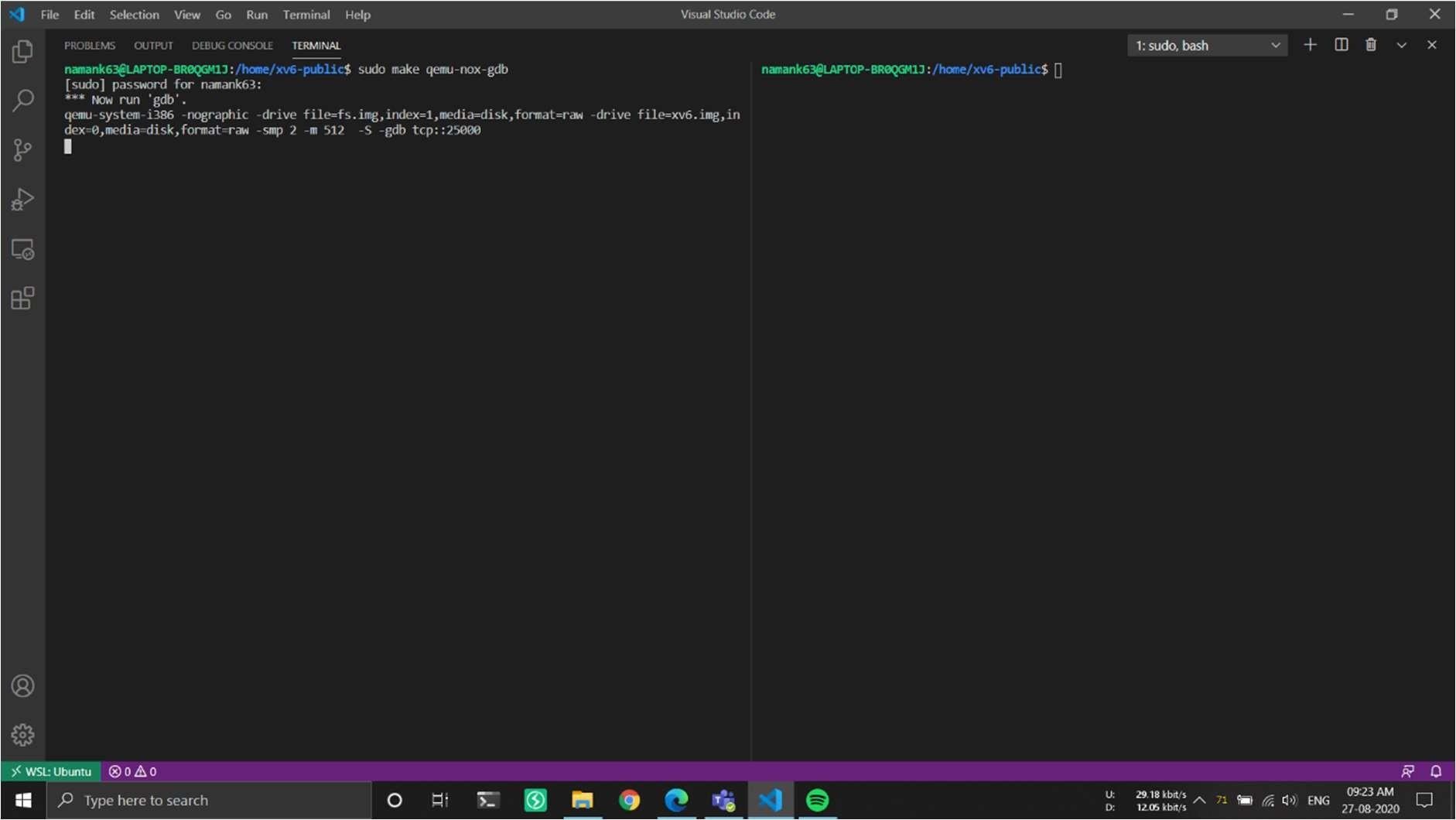
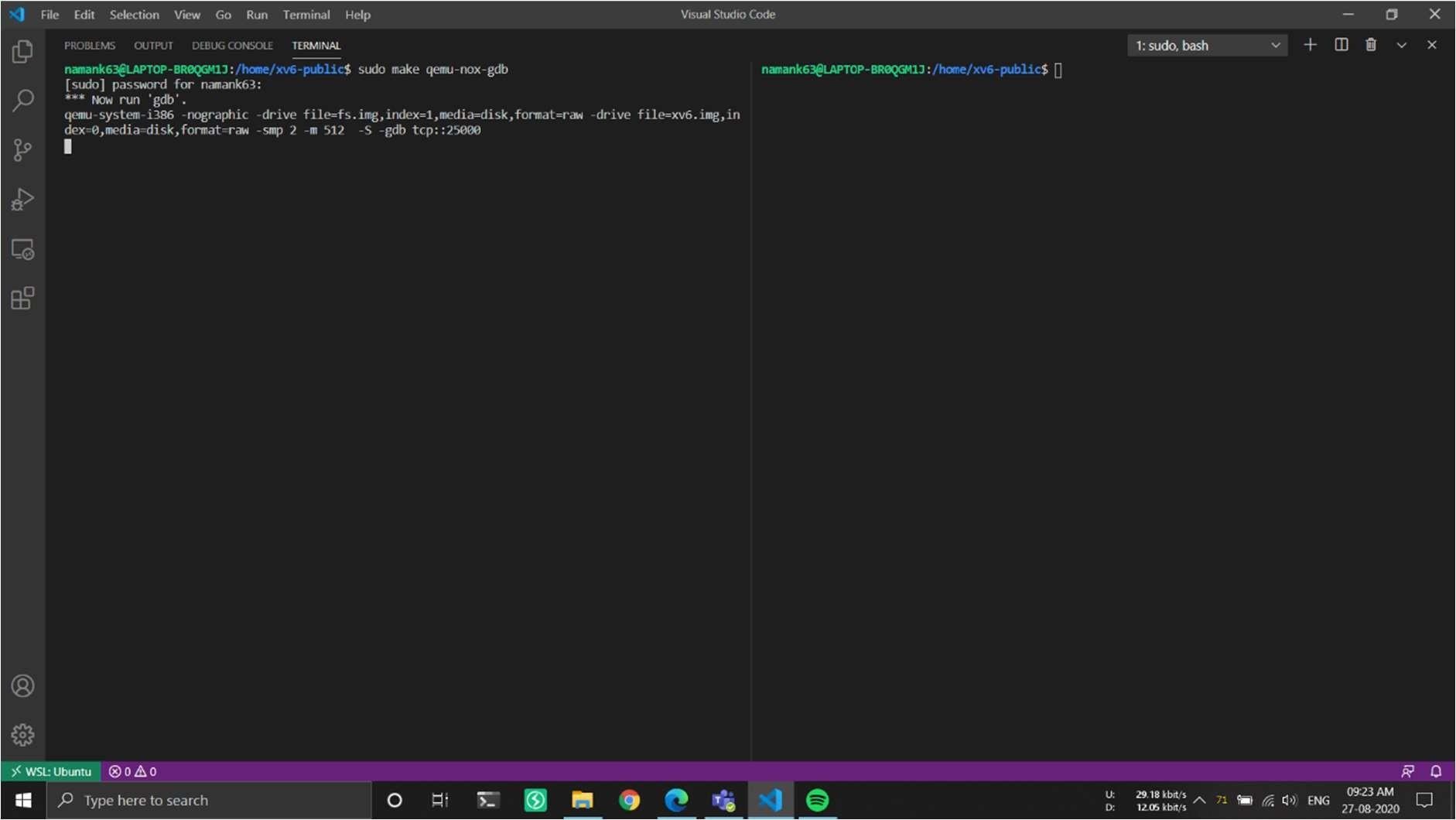
Experiment#1 Naman Kumar (2019CA51)

Step 1: I installed QEMU and xv6 OS.

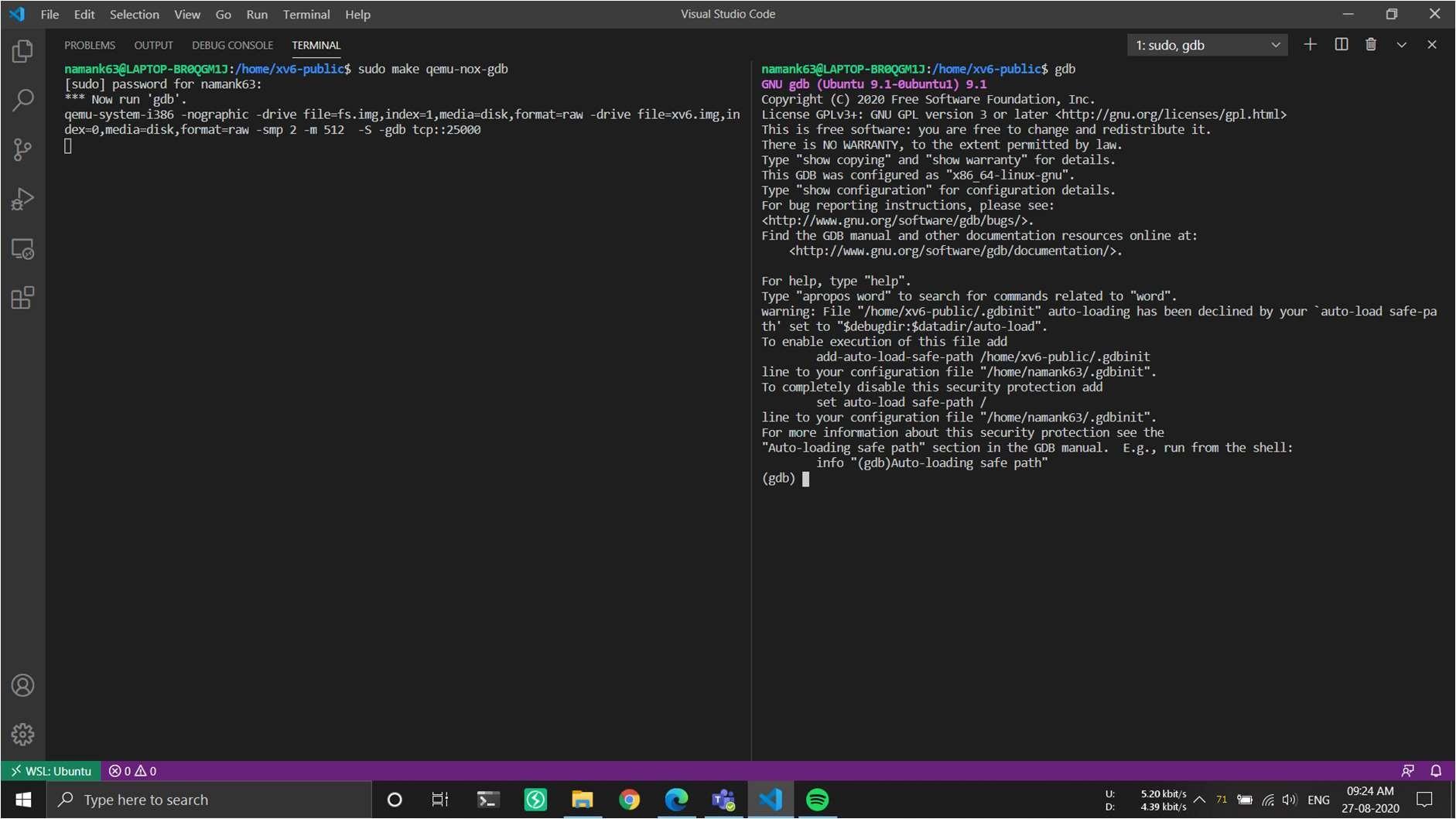
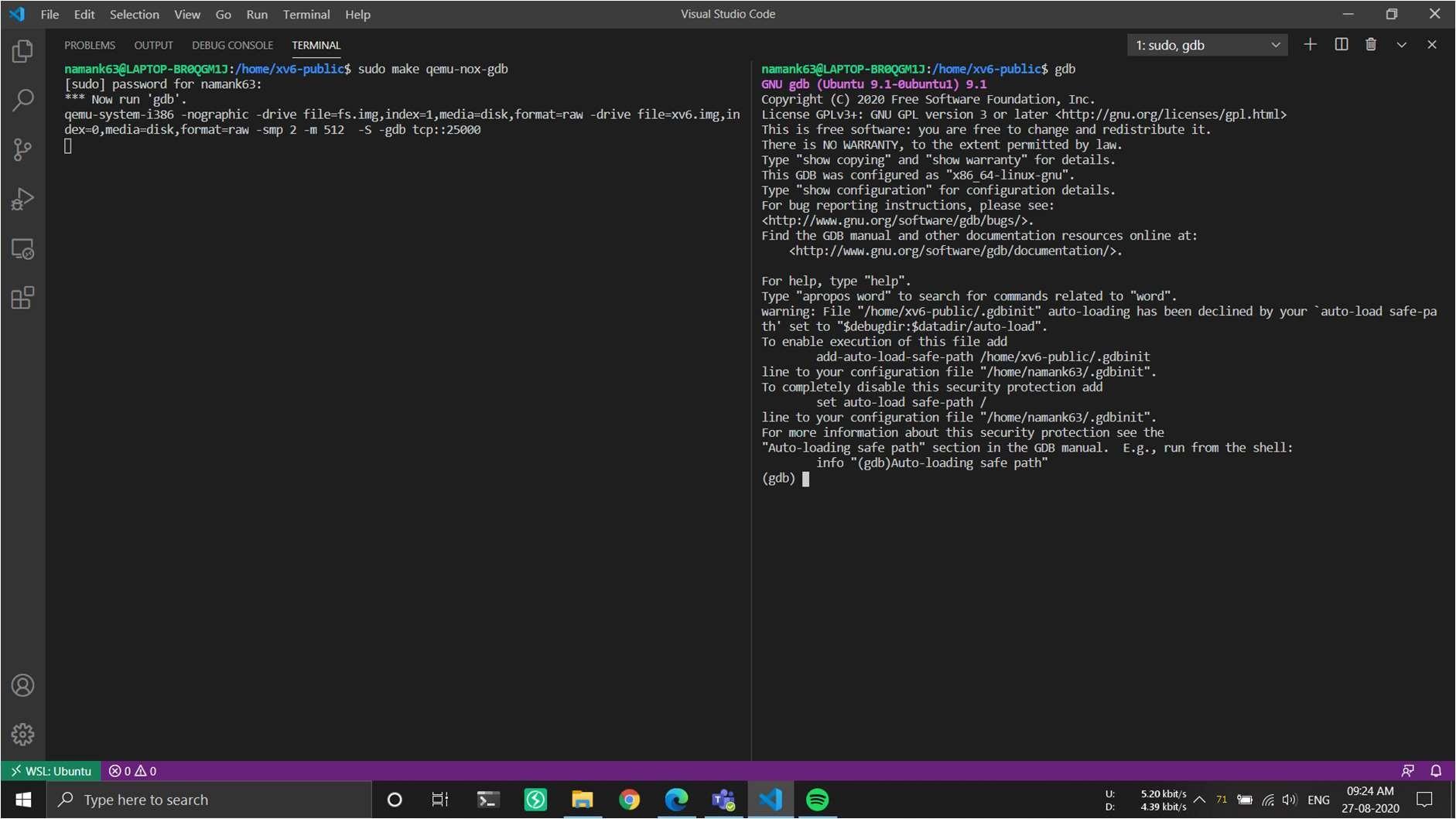
Step 2: Then I open 2 terminals in xv6 OS directory.



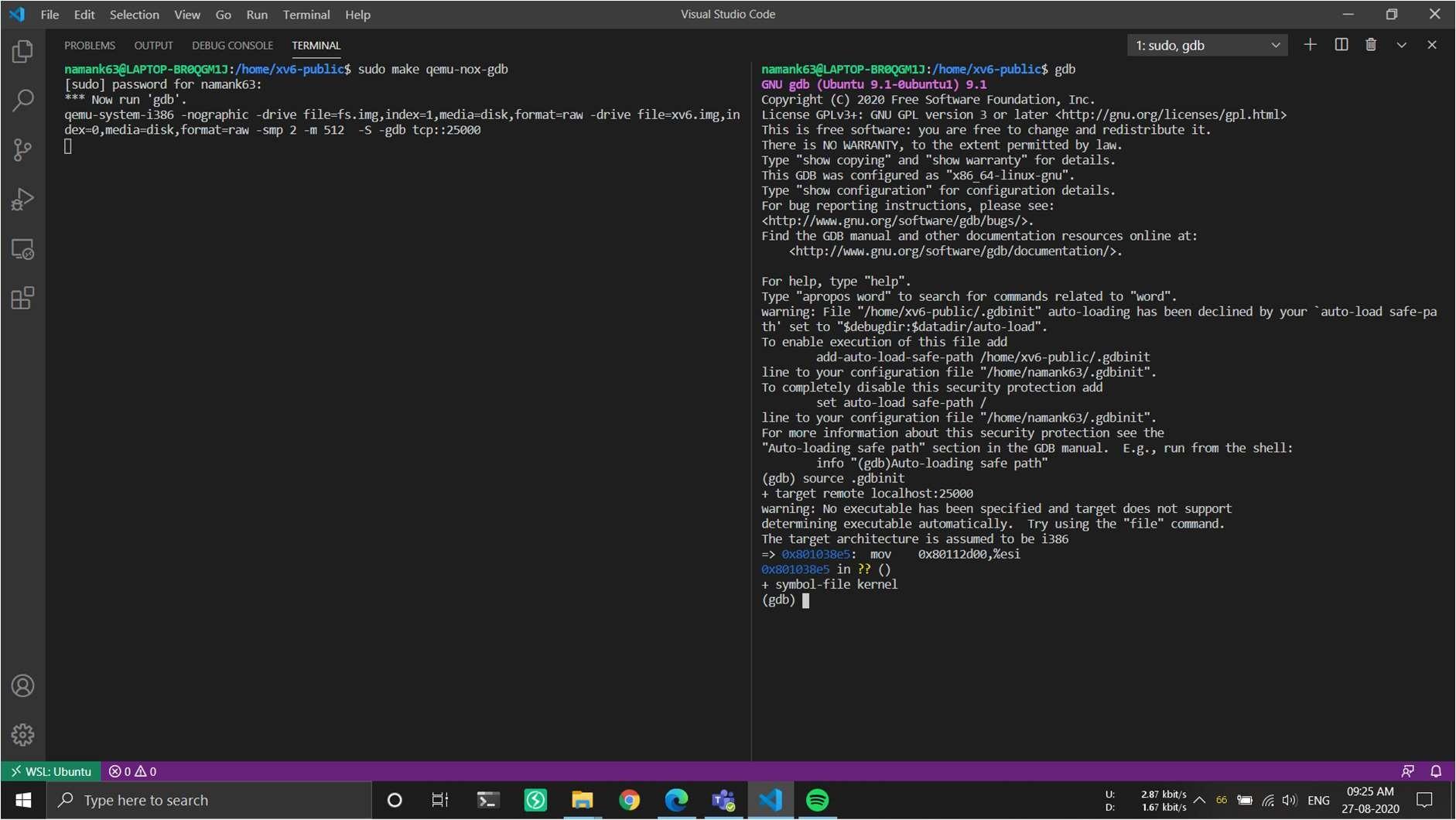
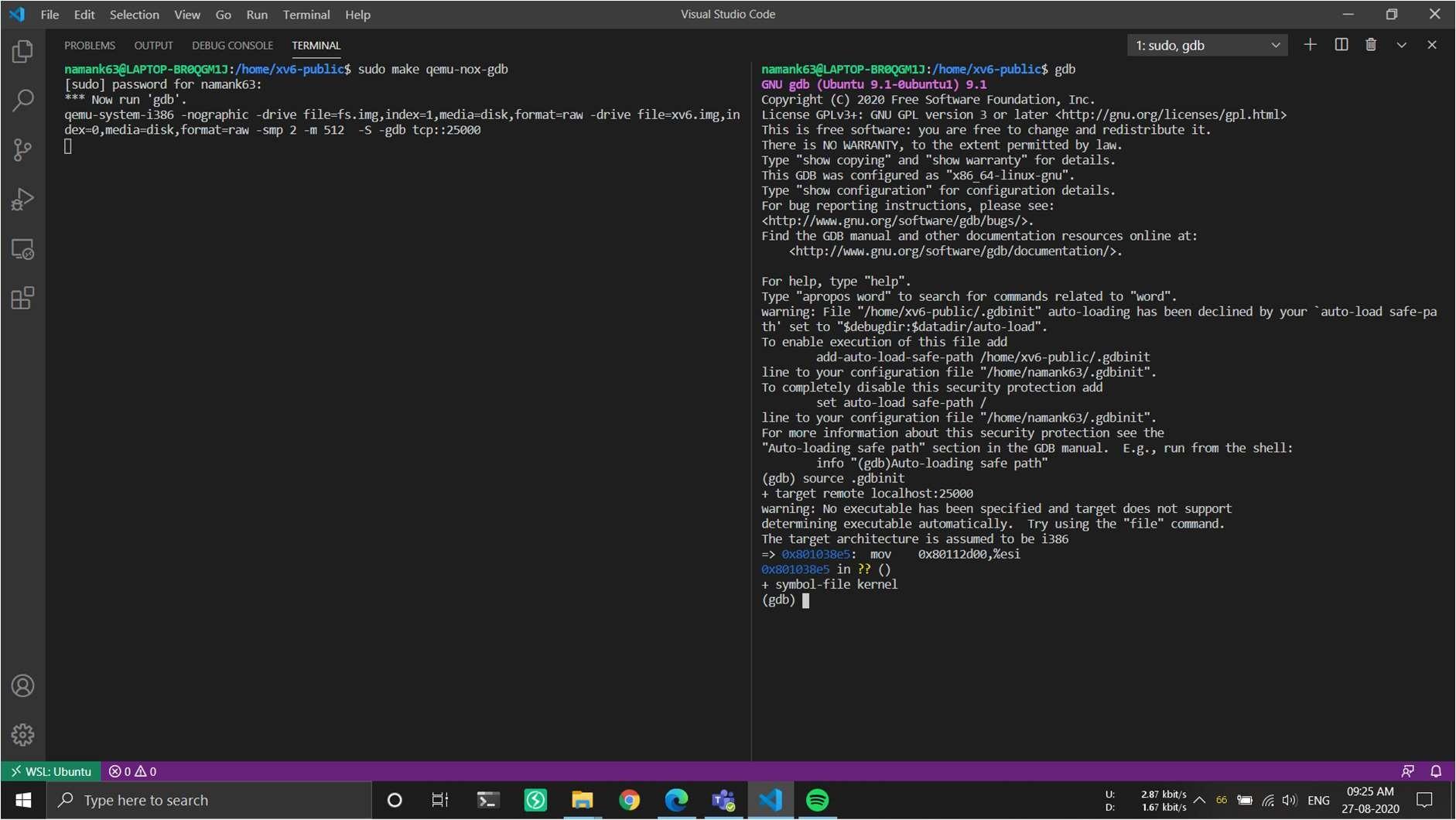
Step 3: In the 1st terminal I ran sudo make qemu-nox-gdb, to start qemu along with GNU Debugger.



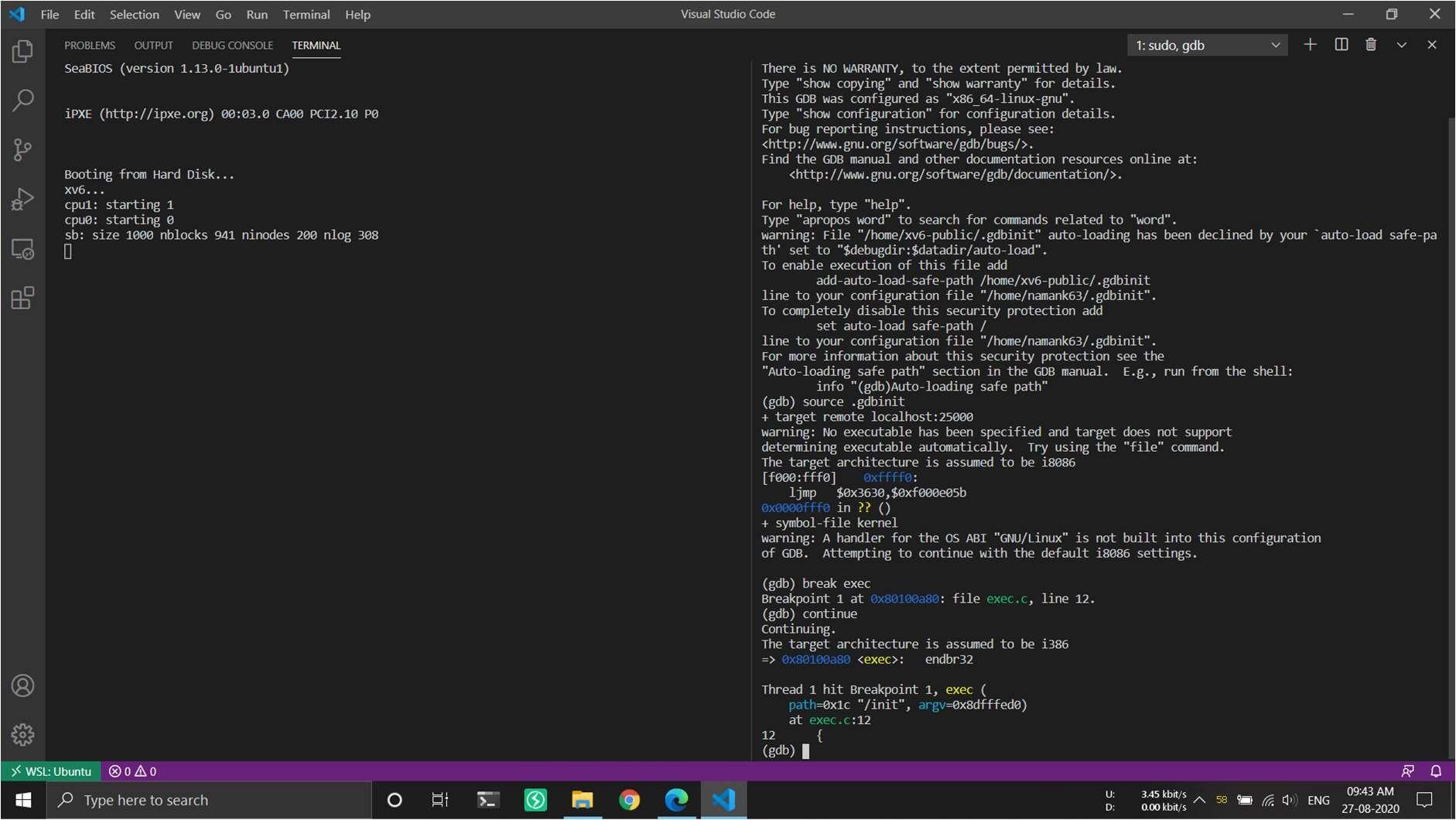
Step 4: And in 2nd terminal I ran just gdb, to run debugger.



Step 5: continuing in the second terminal I ran source .gdbinit and now gdb is connected to the xv6 operating system.



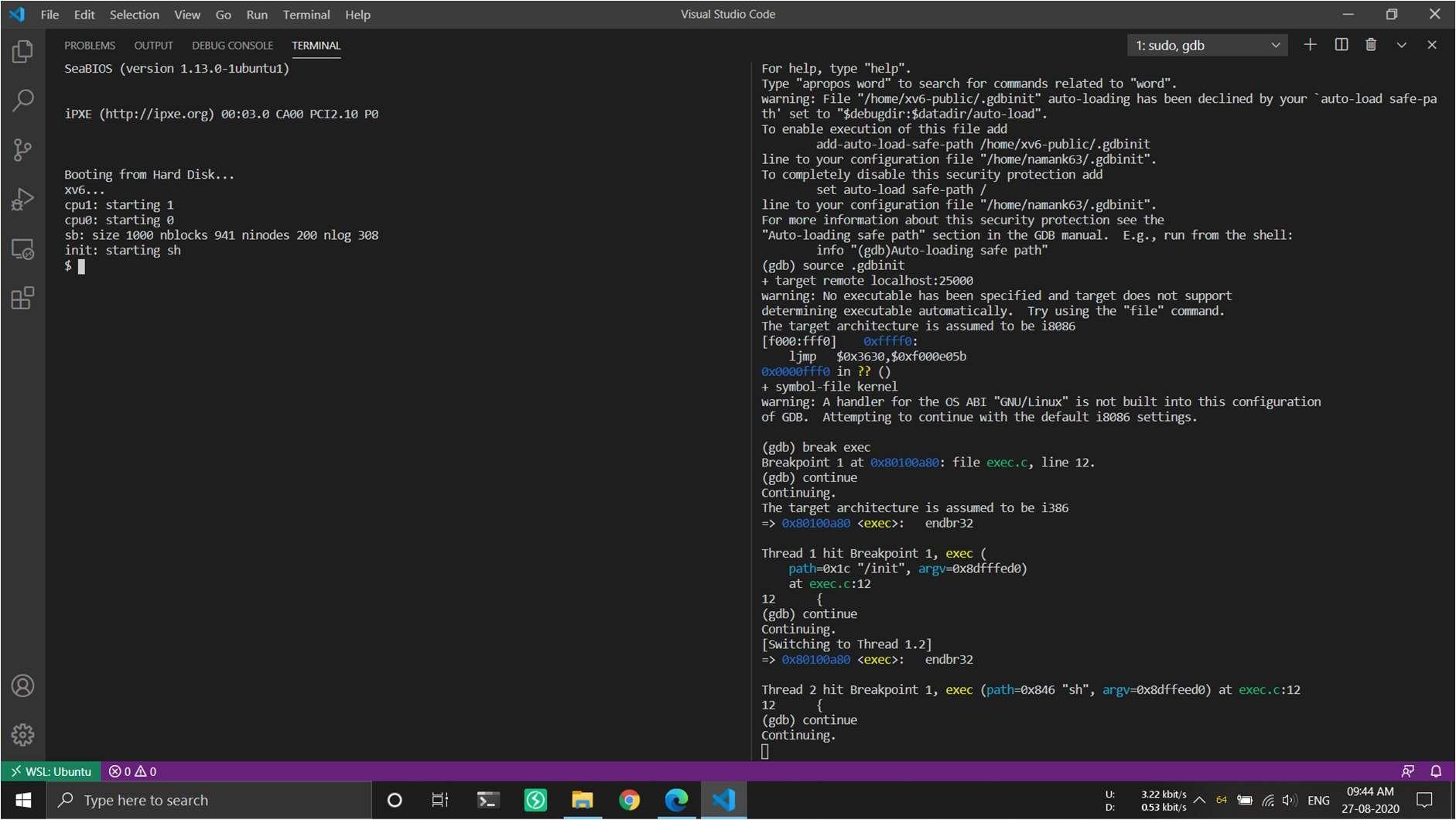
Step 6: Then break exec, the exec system call is used to execute a file which is residing in an active process. When exec is called the previous executable file is replaced and new file is executed. More precisely, we can say that using exec system call will replace the old file or program from the process with a new file or program. The entire content of the process is replaced with a new program.



Step 7: Then I ran continue 3 times…and got $ in the emulator (xv6 OS), First time to reach the first instance where exec() is called.

Second time it takes us to the next point where the exec() is invoked. We stop execution after the OS is initialized at the stage where it is starting the first process (init).

Third time in the xv6 OS window run any shell command like ls, cat etc. At this stage, init started a shell process which is the xv6 shell we get when the OS boots. If you continue again, gdb will not return since it is waiting for a command to be started in the shell. Switch to the other window and try typing a command (for example, cat README) at which time you will get another break as the shell forks then execs the cat program. Feel free to look around at the program when it breaks to see how we reach the system call which should give you ideas about how to add one.



Step 8: Now when I got $ prompt I ran ls command for the first time.

