Suppose my group number is 10-01. (I will use this as an example, you should change it correspondingly in the whole project for your own group)

1. Open two terminal windows. In terminal #1, do steps 2-12. In terminal #2 do steps 2-3. In terminal #1 we will build our project and use gdb to control the emulator in the debug mode. In terminal #2, we will boot our kernels on the VM.
2. Log on to os-class : [pengc@os-class.oregonstate.edu](mailto:pengc@os-class.oregonstate.edu)(Please use your own user name )
3. Call cd /scratch/spring2017/10-01 . If you don’t have this folder(10-01), call mkdir 10-01 to create under /scratch/spring2017/ directory and cd 10-01 to change the directory
4. Call git clone git://git.yoctoproject.org/linux-yocto-3.14 to download the project from GitHub and you will get linux-yocto-3.14.
5. To switch to the correct tag, call cd linux-yocto-3.14, and then git checkout v3.14.26
6. Before we build our kernel or run qemu, we should configure the environment, so run source /scratch/opt/environment-setup-i586-poky-linux.csh
7. Follow 8-12 steps, make a kernel instance for your group.
8. Run cp /scratch/spring2017/files/config-3.14.26-yocto-qemu .config
9. Run make menuconfig and you will get a window
10. In the widow do the following: press / and type in LOCALVERSION, press enter.
11. Hit 1, press enter and then edit the value to be -10-01-hw1 (-10-01-hw1 for group 10-01). This will be appended to the kernel name
12. Run make -j4 all, your kernel instance will be built with 4 threads
13. Run cd .. and then run gdb. Stop here for now.
14. In terminal #2, do step 6.
15. To make a copy for the starting kernel and the drive file located in /scratch/spring2017/files/, do steps 16-17 under your group directory, ex. /scratch/spring2017/10-01
16. Call cp /scratch/spring2017/files/bzImage-qemux86.bin . (This . is an operand)
17. Call  /scratch/spring2017/files/core-image-lsb-sdk-qemux86.ext3 . (This . is an operand)
18. Try run the starting kernel : Call qemu-system-i386 -gdb tcp::5601 -S -nographic -kernel bzImage-qemux86.bin -drive file=core-image-lsb-sdk-qemux86.ext3,if=virtio -enable-kvm -net none -usb -localtime --no-reboot --append "root=/dev/vda rw console=ttyS0 debug" (Here I use 5601 because I took an example of group 10-01, the port number should always be 5600+ some #. In this case, it is 5600 +101.)
19. Since in step 18, we run qemu in debug mode with the CPU halted. We need to use gdb to control it. Do steps 20-21.
20. In terminal #1, it is now in gdb. Run target remote :5601 to connect the qemu.
21. Run continue. Then you will see the change in terminal #2.
22. If you succeed in running qemu, you will be asked to login. Type root and enter. Run uname -a and you will see that the kernel name.
23. Use reboot to reboot the VM.
24. Try run the kernel instance we created in steps 8-12. The kernel instance we built is located in linux-yocto-3.14/arch/x86/boot/ and it is named bzImage.
25. Run qemu-system-i386 -gdb tcp::5601 -S -nographic -kernel linux-yocto-3.14/arch/x86/boot/bzImage  -drive file=core-image-lsb-sdk-qemux86.ext3,if=virtio -enable-kvm -net none -usb -localtime --no-reboot --append "root=/dev/vda rw console=ttyS0 debug"
26. Do steps 20-22 again. You should find the difference in kernel names. The name should be customized into something like ??-10-01-hw1, because uname -a produce the LOCALVERSION string.
27. reboot the vm and use q to quit gdb. You are done with the kernel portion of HW1.

FYI, $SRC\_ROOT in the assignment discription is just a notational shorthand. In my example, it is /scratch/spring2017/10-01/linux-yocto-3.14.