TEAM 13-03

JOSEPH STRUTH — JOSH ASHER — BRYAN LIAUW

CS444

Spring 2017

Abstract

Our first venture into this class involves getting used to the tools and environment we will be working in. This
document takes you through the process of setting up a Qemu virtual machine, compiling a kernel, and getting it
running. We also dive into our first concurrency assignment which is a simple producer consumer look at concurrent
programming. We need these basic steps in preparation of the harder one to come.

April 21, 2017

1 COMMAND LOG

1) **mkdir /scratch/spring2017/13-03** - Create our directory.
2) **cd /scratch/spring2017/13-03** - Change into that directory.
3) **git clone git://git.yoctoproject.org/linux-yocto-3.14** - Clone the yocto 3.14 repo into our directory.
4) **cd linux-yocto-3.14** - Change to yocto directory.
5) **git checkout v3.14.26** - Checkout the v3.14.26 branch of yocto.
6) **source /scratch/opt/environment-setup- i586-poky- linux.csh** - To set the proper environmental variables for when we compile.
7) **cp /scratch/spring2017/files/config-3.14.26- yocto-qemu .config** - Copy over the example configuration file for the kernel.
8) **make menuconfig** - Opens the kernel configuration menu.
9) **/** - Open menu config search.
10) **LOCALVERSION** - Locate the local version, hit 1 to change it.
11) **-13-03-hw1** - Change LOCALVERSIOn to our group name.
12) **make -j4 all** - Start compilation of the kernel.
13) **gdb** - Open gdb, do this from another terminal or using screen.
14) **cp /scratch/spring2017/files/bzImage-qemux86.bin** - Copy practice kernel into our directory.
15) **cp /scratch/spring2017/files/core-image- lsb-sdk-qemux86.ext3 .** - Copy virtual machine into our directory.
16) **qemu-system-i386 -gdb tcp::5633 -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"** - To launch the virtual machine using the practice kernel.
17) **target remot :5633** - This is done in the gdb screen to connect to the virtual machine.
18) **continue** - This is done in the gdb screen to let the kernel conintue loading, as it was paused on launch.
19) **root** - On the VM screen, used to login to Linux.
20) **uname -a** - Displays information from the practice kernel about its version.
21) **reboot** - Reboots the VM, which actually shuts it down.
22) **qemu-system-i386 -gdb tcp:5633 -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"** - Run the copy of the kernel we compiled that has the altered LOCALVERSION variable.
23) **target remote :5633** - From gdb window log into our VM running our kernel.
24) **continue** - From gdb window allow the kernel to continue loading.
25) **root** - Login to our kernel.
26) **uname -a** - Check that our change to LOCALVERSION was correct by display the kernel version information.
27) **reboot** - Shut down our VM.

2 QEMU FLAGS

- **qemu-system-i386** The QEMU 32 bit program.
- **-gdb tcp::5633** Wait for gdb connection on tcp port 5633.
- **-S** Do not start CPU at startup.
- **-nographic** Disables VGA output so no graphics are displayed for the output. The serial port is directed to your console.
- **-kernel linux-yocto-3.14/arch/x86/boot/bzImage** Select which bzImage to use. In our case the one we compiled.
- **-drive** Select a new drive.
- **file=** Option of drive command (which are separated by commas), name of drive file you want ot use goes after the equals.
- **if=** Defines what type of interface the drive is connected to. In our case virtio.
- **-enable-kvm** Enable the Kernel-based Virtual machine.
- **-net none** Do not configure any network devices.
- **-usb** Enable usb drivers.
- **-localtime** Couldn't find this on the man page, I am guessing it uses the computer local time.
- **–no-reboot** Exit instead of rebooting.
- **–append "root=/dev/vda rw console=ttyS0 debug"** Use the following command line for the Linux command line.

3 CONCURRENCY QUESTIONS

3.1 Question 1

The main point of this assignment is that this is a simple concurrency problem where only two threads interact with each other through a common middle ground buffer. Since there are only two threads, locking and accessing and maintaining it is rather basic.

3.2 Question 2

A way we approach this problem is to first create the 3 components of the program: the producer, consumer and buffer. This is done by doing the functions for the consumer and producer first before worrying about getting random value to determine how long the items live in the buffer. After getting the desired interactions between the two functions and the buffer, we implemented the random afterwards.

3.3 Question 3

We go through the testing by first testing how the program works under normal circumstances without the extreme in the constraints. After it works, we tested it with the constraints, which is manipulating the time to consume to make it fill the buffer until it is full. Afterwards, if it already works under this constraint we will go to the next constraint which is empty buffer. If this has worked too, we can assume this would work under most circumstances.

3.4 Question 4

We learn how to do basic concurrency problems. This assignment helps us remember how to do basic thread programming from our 344 class. Since this is pretty basic, it helps us get to speed and therefore does not overwhelm us.

4 GIT LOG

Detail	Author	Description
805c492	Joshua Asher	Added the reports direcotry with tex template for homework.
ab7a9b7	Joshua Asher	Commiting Changes to HW1.tex, didn't work remotely
2dbc92b	Bryan Liauw	Testing git and added changes to HW1.tex
2dbc92b	Josh Asher	Testing New branch
20d3cc1	Joseph Struth	Added my code for concurrency 1.
20d3cc1	Josh Asher	Added the qemu flags to the report for homework 1.
2dbc92b	Joseph Struth	Pushed changes to HW1.tex
20d3cc1	Bryan Liauw	More changes to HW1.tex.

5 WORK LOG

Date	Author	Details
4-13-2017	Josh Asher	Logged into os class and fooled around.
4-15-2017	Josh Asher	Took my turn to compile the code and run through the steps of homework one. That included starting the practice kernel on the qemu VM and then running the one we compiled.
4-15-2017	Bryan Liauw	Done my own implementation of concurrency one with normal rand to understand how it works.
4-16-2017	Josh Asher	Wrote the concurrecy example myself and got it working. We will probably use Joshephs code it was more elegant.
4-17-2017	Josh Asher	Check over Joseph's code and ran it for a long time looks good.
4-17-2017	Josh Asher	Got git setup in the direcotry, having some issues but working on them.
4-17-2017	Josh Asher	Added a reports directory with homework folders and posted a working tex template and Makefile.
4-20-17	Bryan Liauw	Added answers to concurrency questions based on my perspective.
4-20-17	Josh Asher	Added the command line section to the homework. Also added the explanation of the qemu flags to the reports.
4-20-17	Josh Asher	Working on getting the git log in latex. Found a nice script and trying to implement it.
4-21-17	Josh Asher	Got the git log script working and working on my part of the work log.