

Project One Writeup

Spring 2017

Shuai Peng (pengs), Anya Lehman (lehmana), Andrew Bowers (bowerand),

April 20, 2017

Abstract

This is our write up for the project one *Getting Acquainted*.

1 COMMAND LOG

- 1) To start we logged on to the os-class `ssh username@os-class.oregonstate.edu`
- 2) Then we used `cd` to get to the correct folder in `scratch/spring2017` `cd /scratch/spring2017`
- 3) Then we made a group folder for us all to work in `mkdir 11-04`
- 4) Next we struggled to make said folder accessible to all of our group members by changing the permissions on it so not just the group member that created the directory could work in it `chmod 777 11-04`
- 5) Then we called `git clone` to download the project from the GitHub account and we checked to make sure we got all the correct files `git clone git://git.yoctoproject.org/linux-yocto-3.14`
- 6) Then we switched to the tag we needed by using `cd` again and going into the directory that was cloned into our folder `cd linux-yocto-3.14`
- 7) Following this we checked out the v3.14.26 `git checkout v3.14.26`
- 8) Next came configuring the environment which we did by calling `source /scratch/opt/environment-setup-i586-poky-linux`
- 9) Then we made a kernel instance for our group
- 10) Then we copied in the files that let us configure `cp /scratch/spring2017/files/config-3.14.26-yocto-qemu .config`
- 11) Then we ran `make menuconfig`
- 12) A window popped up
- 13) In this window we pressed `/` and typed `LOCALVERSION`
- 14) Next we pressed `1` and edited the value to be `-11-04-hw1` to make that the name of the kernel
- 15) Then we built our kernel with four threads by running `-j4`
- 16) Then we ran `cd ..` followed by `gdb`
- 17) Our next step was to move onto a different laptop and called `source /scratch/opt/environment-setup-i586-poky-linux` again
- 18) Then we made a copy for the starting kernel and the drive file located in the scratch directory by calling `/scratch/spring2017/files/core-image-lsb-sdk-qemux86.ext3`
- 19) Then we tried running the starting kernel `qemu-system-i386 -gdb tcp::5604 -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"`
- 20) Since we previously ran the qemu in debug mode, we used `gdb` to control it so, back on the original computer, we connected the qemu by running `target remote :5604`
- 21) Then we rebooted the VM `reboot`
- 22) Then we tried running the kernel instance we had created `linux-yocto-3.14/arch/x86/boot/`
- 23) Then we ran `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"`
- 24) Finally we rebooted the vm and used `q` to quit

2 EXPLAITION OF THE FLAGS

3 QUESTIONS REGARDING CONCURRENCY

- 1) *What do you think the main point of this assignment is* The main point of this assignment is to get us to learn more about conncerency and to build up our ability to understand and think in parallel. Parallel computing is the simultaneous use of multiple resources to solve a computational problem. This is important because compared to serial computing, parallel computing is much better for tasks such as modeling, simulating, and working on more complex, real world problems. The conceptual problem of this assingmnt is not as challenging as understuanding how to implement it. The secondary point of this assignment is to get us familiar with the techniques we need for the rest of this class.
- 2) *How did you personally approach the problem? Design decisions, algorithm, etc.* To start this problem, we first worked out the pseudo code with pen and paper. We designed it so that the Consumer is a function and the Producer is a function. Then we also created a function that would produce a random number. We went about this by using a struct to hold the values that the Producer creates and the Consumer are taking. The first number in each struct will be the randomly generated number for the consumer to print and the second will be the time the consumer should wait. This struct is generated in our main function.
- 3) *How did you ensure your solution was correct? Testing details, for instance.* To ensure that our solution was correct...
- 4) *What did you learn?* This assignment taught us how to properly create a program that uses parallel programming. We re-learned a lot of old material about writing in c and learning how to create a struct and how to create an array when we realized that we couldn't create a struct.

4 VERSION CONTROL LOG

File Version	Group Member(s)	What Was Done
V1	Andrew	Created the conccerency file to work from
V2	Shuai with consultation of group members	Created the functions

5 WORK LOG

Date	Group Member(s)	Start Time	End Time	Total Time Worked	Accomplished
April 11th	All	11:00am	12:00pm	1 hour	Met our group members and exchanged contact information. Set up our working environment including creating our kernel and making the group folder to work in. Made sure all members of the group could access the group folder and the group GitHub account.
April 18th	All	11:00am	12:00pm	1 hour	Set up the folder to contain the work we will be doing in this project. Started on the writeup by setting it up and completing the write up for the log of commands. Got the github to work. We also read through the concurrency assignment.
April 19th	All	2:30pm	8:30pm	5 hours	Begain the concurrency assignment. Created the psudeo code for the exersise and discused how we wanted to go about completing the exersise. Created the c file and made our struct and all of our functions. Also worked on the write up to create the work log and begain to asnwer the questions regarding the concurrency exercise.