CS 444 - Fall 2017 - Concurrency 1

Producer and Consumer

Kristen Patterson, Kenneth Steinfeldt

Abstract

Some sort of abstract.

I. Command Logs

The first thing we did to complete running QEMU was we created a repo and cloned in Yocto.

```
git clone https :// github.com/patterkr/CS444-23.git
cd CS444-23/
git clone git :// git.yoctoproject.org/linux-yocto-3.19
```

Then we made sure to run the right version specified in the assignment.

```
git checkout v3.19.2
```

Before we could run the qemu we had to copy over the starting kernel and the drive file.

```
cp /scratch/files/bzImage-qemux86.bin /scratch/fall2017/23/CS444-23/linux-yocto-3.19
cp /scracth/files/core-image-lsb-sdk-qemux86.ext4 /scratch/fall2017/23/CS444-23/linux-yocto-3.19
```

Then we entered bash to source the environment.

```
source /scratch/opt/poky/1.8/environment-setup-i586-poky-linux
```

After it was properly sourced and the files were copied over, we ran the really long qemu command replacing ???? with our port which was 5523.

```
qemu-system-i386 -gdb tcp::5523 -S -nographic -kernel bzImage-qemux86.bin
-drive file=core-image-lsb-sdk-qemux86.ext4,if=virtio -enable-kvm -net none
-usb -localtime --no-reboot --append "root=/dev/vda rw console=ttyS0 debug".
```

Now that the QEMU was in debug mode, we opened up a new terminal and ran code in bash to connect to it using our port 5523.

```
source /scratch/opt/poky/1.8/environment-setup-i586-poky-linux
$GDB
target remote : 5523
continue
```

Next we needed to build Yocto so we opened another new terminal window and copied over the configure files and made it using the allotted 4 jobs.

```
cp /scratch/files/config-3.19.2-yocto-standard
/scratch/fall2017/23/CS444-23/linux-yocto-3.19/.config
make -j4 all
```

Finally, to make sure it worked, we went back to the terminal where we connected to the QEMU with GDB, logged into root, and ran the following commands.

```
uname -r
```

Which showed that we were running in Yocto standard.

II. Concurrency Solution

III. QEMU Flags Explanation

A. gdb

Wait for gdb connection on tcp::5523.

B. S

Do not start CPU at startup.

C. nographic

Totally disable graphical output so QEMU is a simple command line application.

D. kernel

Use bzImage as kernel image.

E. drive

Defines a new drive file.

F. enable-kvm

Enables KVM full virtualization support. KVM is Kernel Virtual Machine.

G. net

Indicate that no network devices should be configured.

H. usb

Enable the usb driver.

I. localtime

Use local date and time.

J. no-reboot

Exit instead of rebooting

K. append

Use cmdline as kernel command line.

IV. Concurrency Questions

A. What do you think the main point of this assignment is?

B. How did you personally approach the problem?

C. How did you ensure your solution was correct?

D. What did you learn?

V. Version Control Log

Author	**Date**	**Message**
Kristen Patterson	October 4, 2017	Initial commit
Kristen Patterson	October 4, 2017	Update README.md
Kristen Patterson	October 4, 2017	Built the kernel
Kristen Patterson	October 7,2017	Started concurrency programming, implemented buffer, need to research inline asm
Ken Steinfeldt	October 8, 2017	create stack(ish) and start p and c funcs
Ken Steinfeldt	October 8, 2017	begin produce and consume funcs
Ken Steinfeldt	October 8, 2017	Merge pull request 1 from patterkr/implement-stack
Kristen Patterson	October 9, 2017	Implemented rng and started tex doc

VI. Work Log

A. Kristen Patterson

I started working on Wednesday, October 4th. I first setup the environment and did all the commands the properly build and run the kernel. It took me about 3 hours to setup the environment and build the kernel. Then on Saturday the 7th I started the concurrency assignment by bringing in the mt19937ar.c and the Makefile. I also implemented some of the buffer to be used as well as a function to check if it was empty or full. Then on Monday the 9th I finished the random number generator function and finished my half of the tex document.

B. Ken Steinfeldt