Homework 1 Writeup

CS444 – Operating System II

Professor: D. Kevin McGrath

Yipeng Song, Samuel Schultz, Aaron Leondar

April 21, 2017

1 Getting Started

- Command usage:
  - 1 ssh xxxx@os-class.engr.oregonstate.edu (Replace xxxx by onid)
  - 2 mkdir /scratch/spring2017/11-07
  - 3 git clone git://git.yoctoproject.org/linux-yocto-3.14
  - 4 cd linux-yocto-3.14
  - 5 git checkout v3.14.26
  - 6 source /scratch/opt/environment-setup-i586-poky-linux.csh
  - 7 cp /scratch/spring2017/files/config-3.14.26-yocto-qemu .config
  - 8 make menuconfig
  - 9 press / and type in LOCALVERSION, press enter.
  - 10 Hit 1, press enter and then edit the value to be -11-07-hw1
  - 11 Exit the window
  - 12 make -j4 all
  - 13 cd ..
  - 14 gdb
  - 15 in terminal 2, do 6
  - 16 Under /scratch/spring2017/11-07, call cp /scratch/spring2017/files/bzImage-qemux86.bin .
  - 17 cp /scratch/spring2017/files/core-image-lsb-sdk-qemux86.ext3 .
  - 18 qemu-system-i386 -gdb tcp::5617 -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”
  - 19 In terminal 1, target remote :5601
  - 20 continue
  - 21 Type root and enter
  - 22 Type uname -a to see the kernel name

2 Flag Usage

Flags descriptions are:

- 1 -gdb: Wait for gdb connection to the kernel
- 2 -s: open gdbserver on my TCP port 5617
- 3 -nographic: disable graphical output
- 4 -kernel bzImage Use bzImage as kernel image

- 5 -drive select the file to use for the disk image
- 6 -enable-kvm Enable KVM full virtualization support
- 7 -net none Indicate that no network devices should be configured.
- 8 -usb Enable the USB driver
- 9 -localtime use local time (PST)
- 10 -no-reboot not to reboot after exit
- 11 –append use specified information as command line.

3 Concurrent Exercise

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

  To review the thread and multi-process knowledge.
- How did you personally approach the problem? Design decisions, algorithm, etc.

  I read the description a couple times and lay out the general structure of the code: producer and consumer. Since this project is small, I did not have much to deal with algorithm to achieve efficiency.
- How did you ensure your solution was correct? Testing details, for instance.

  By adding some hand-driven unit tests (via print).
- What did you learn?

  This concurrent exercise is more like a review. I pick up some old materials for Operating System I.

4 Commit History

N/A (No commit was made)

5 Work Log

Work time log as follows:

4/9/2017 18:00-18.30 Read the assignment description twice (30 minutes)

4/11/2017 11:00-12:00 Start building kernel during recitation (1 hours)

4/18/2017 11:00-12:00 Find we did something wrong last time, trying to fix the problem (1 hour)

4/20/2017 10:00-12:00 Build the kernel successfully (2 hours)

4/20/2017 14:00-18:00, 19:00-22:00 Start working on concurrency assignment (4 hours)

4/20/2017 16:00-19:00 Start editing writeup (3 hours)

4/21/2017 18:00-24:00 Finish Writeup and debugging concurrency assignment (6 hours)