Introduction to QEMU/Linux Assignment

CS 256/456
Fall 2011
Li Lu

UNIVERSITY of ROCHESTER

About QEMU - I

- What is QEMU?
  - A processor emulator
  - Emulate user level processes for target CPU on host CPU
  - It can be viewed as a hosted virtual machine monitor

About QEMU - II

- Why QEMU?
  - For the kernel programming in the following assignments
  - Isolate what you did from the real machine
  - Make the programming and debugging easier

QDGL

- QDGL: (Q)EMU – (D)ebian (G)NU/(L)inux
  - QEMU: the system emulator
  - Debian GNU/Linux: operating system
  - Linux kernel: testbed for kernel development

Content of QDGL

- QEMU: binaries of QEMU and other necessary files
- hda.img: Debian's virtual disk image
- linux-2.6.26.5.tar.bz2: Linux kernel

Quick Start

- Get QDGL by copy
  - cs256: `cp –r ~cs256/QDGL ~`
  - cs456: `cp –r ~cs456/QDGL ~`
- Space requirement
  - Need 562MB for QDGL
- Need 400MB for kernel compilation
- Total: at least 1GB free disk space in your accoun

Test it

- Start QEMU and default Debian installation
  - `cd QDGL`
  - `./qemu/bin/qemu -m 64M -L ./qemu/share/qemu/ -hda hda.img -nographic`
- Login:
  - User: `root` or `cs2456`
  - Password: `******`
- Shutdown
- `Poweroff` (or `sudo poweroff` if logged as cs2456)

New Kernel

- Extract new kernel
  - `tar -jxvf linux-2.6.26.5.tar.bz2`
- Compile or re-compile
  - `cd linux-2.6.26.5`
  - `make`
  - `./qemu/bin/qemu -m 64M -L ./qemu/share/qemu/ -hda hda.img -nographic -append "root=/dev/hda1 console=ttyS0,115200n8 console=tty0" -kernel linux-2.6.26.5/arch/i386/boot/bzImage`
- Check: `uname -a`

Adding a system call

- What is system call?

```
#include <linux/unistd.h>

int main() {

    int i = getuid()

    printf (“Hello World! My uid is %d\n”, i);

}
```

- Syscall: interface between two levels of Linux
  - Kernel level: implements system call
  - User level: includes `unistd.h`, calls `getuid()`

Kernel level - I

- Implement the function

```
#include <linux/kernel.h>

asmlinkage long sys_cs2456_test(int arg0)

{

    printk("--syscall arg %d", arg0);

    return((long) arg0);

}
```

- Put this function in certain file under certain directory, then modify the Makefile accordingly

Kernel level - II

- Add the function pointer in:

```
arch/x86/kernel/syscall_table_32.S

.long sys_cs2456_test     /* 327 */
```

- Add the system call number in:

```
include/asm-x86/unistd_32.h

#define __NR_cs2456_test     327
```

Recompile and check

- Re-compile
  - `cd linux-2.6.26.5`
  - `make`
  - `./qemu/bin/qemu -m 64M -L ./qemu/share/qemu/ -hda hda.img -nographic -append "root=/dev/hda1 console=ttyS0,115200n8 console=tty0" -kernel linux-2.6.26.5/arch/i386/boot/bzImage`
- Check
  - `uname -a`

User level

- Start QEMU
- Write the user program to use the system call

```
#include <linux/unistd.h>

#include <sys/syscall.h>

int main(int argc, char *argv[])

{

    syscall(327 , 2456);

    return 0;

}
```

Summary

- QEMU: start, login, shutdown
- Linux kernel source code
- Example: adding a system call

- Thanks to Konstantinos Menychtas, who setup the platform of QEMU.

- Thanks to Hongzhou Zhao and Zhuan Chen, who gave this talk in previous years.

- We might experience problems with the new department upgrades, so we need your cooperation and help on the bugs and problems you encounter.

UNIVERSITY *of* ROCHESTER