Name: _____ **Student ID #:** _____

Lab #3 Solutions: COMP 3000 (Operating Systems)
March 10, 2007

1 Part A

1.1 Virtual Machine Emulators

1. **[2]** By using the `lspci` command, determine what PCI bus hardware is being emulated. Specifically, what video card does `QEMU` report to the operating system?

   Cirrus Logic GD 5446

2. **[2]** Compare the output of the `lspci` on the emulated vs the real system. What is one piece of hardware available on the real system that is not available on the emulated system?

   USB Controller, Sound Card, SMBus controller, FireWire controller

1.2 The Linux Kernel

1. **[2]** The Linux kernel is stored in the `linux-2.6.19.1` directory. What command line can be given to configure the kernel with a random assortment of configuration options?

   make randconfig

2. **[2]** Using the configuration command `make menuconfig`, browse through the menus until you find the option for *Magic SysRq Key*. What menu is the option contained in?

   Kernel Hacking

3. **[2]** Enable support for the ISA SoundBlaster 16 card under ALSA (Advanced Linux Sound Architecture) and rebuild the kernel using `make`. Where did you find the option for the SoundBlaster 16 PnP ISA card?

   Device Drivers → Sound → ALSA → ISA Devices → SoundBlaster 16 (PnP)

4. **[2]** Run the new kernel you built with SoundBlaster support. The kernel will now detect a sound card on the emulated system. The command `dmesg` will display the debug messages output while the system is booting. What IRQ is the SoundBlaster device tied to?

   5

2 Part B

2.1 Virtual Machine Emulators

1. **[5]** This lab used `QEMU` as a virtual machine emulator to allow us to build, modify, and debug a Linux kernel without having to reboot the real machine. Very briefly, how does a virtual machine emulator work (i.e. what does it do)?

   A virtual machine emulates (simulates) the behavior of the CPU and some generic I/O devices.

2. **[10]** Using the documentation for `QEMU` (and the documentation for `GDB`) (both of which are available online), answer the following questions:

   (a) **[2]** What function does the kernel hang in when you shut down your emulated kernel? To get the kernel into the *hung* state, run the kernel in the emulator, log in as root, and then type `halt`. Wait for the *System halted* message.

   default_idle

   (b) **[4]** What option did you have to pass into `qemu` to enable gdb debugging of the running kernel?

   -s to enable remote gdb debugging.

   (c) **[4]** What command did you have to give to the gdb command line to debug the kernel running in `qemu`?

   target remote localhost:1234

3. **[5]** By reading *QEMU, a Fast and Portable Dynamic Translator* by Fabrice Bellard [1], determine why QEMU is capable of running quickly compared to other emulators like Bochs[2].

   It recompiles the code to the current hardware dynamically instead of always interpreting each assembly instruction.

2.2 The Linux Kernel

In this section, you will modify the Linux kernel. If you end up breaking the kernel beyond repair, you can always retrieve a fresh copy from the `/lab-data` directory.

1. **[10]** By using any available resource, determine why operating systems are almost always traditionally written in C (hint: examine the original design purpose and use of the C language). Your answer should be about a page long and cite at least 2 reasons. Be sure to cite appropriate references.

   C allows for explicit memory management, low level control of hardware resources, and portability. Note that C was originally designed to make it suitable for implementing an operating system—UNIX. A good source is Dennis Ritchie's history of the development of the C language: `http://cm.bell-labs.com/cm/cs/who/dmr/chist.html`

2. **[10]** Modify the system kernel to print *Bye for now.* after displaying the *System halted* message. In addition, have it print the system state by calling `show_state`. The files *arch/i386/kernel/reboot.c* and *kernel/printk.c* may be useful. What function did you modify? What process is running?

   - **machine_halt is modified**
   - **the halt process is running.**

[1] http://www.usenix.org/publications/library/proceedings/usenix05/tech/freenix/bellard.html
[2] http://bochs.sourceforge.net/