Assignment 1 (GeekOS "Hello World")

26 January 2011

due 2 February at noon

Install VirtualBox 3.x

- Download my LinuxVM image (about 800M) from https://s3.amazonaws.com/liucs.net/LinuxVM.zip This virtual machine image is preconfigured with the software we'll need for this course. (If you already have a recent Linux installation you'd like to use, it may be possible by installing the right packages. I can help with the details.)
- Extract the files from the zip. You can do this on Windows with right-click > Extract all, or on Mac by double-clicking.
- 3. Download VirtualBox for your **host OS** (Windows or Intel Mac). There is a fairly recent version 4, but I'd rather stick with the 3.2 line because I'm more confident it will work. You can get it here: http://www.virtualbox.org/wiki/Download_Old_Builds_3_2
- 4. Install VirtualBox. If the installer gives you any trouble, try rebooting.
- 5. Run the VirtualBox application. You may just cancel the registration form.

Configure and boot Ubuntu Linux VM

- 6. From the File menu, select Import Appliance. Push the Choose button, navigate to the LinuxVM folder you extracted, and select the Linux.ovf file inside.
- 7. Click Next, then Import. The import process can take some time. Once it has finished, you can delete the zip file, the extracted folder, and the VirtualBox installer.
- 8. Now you should have a Linux VM in the left panel of VirtualBox. **Before you start it**, select it and click the Settings button on the tool bar. In the System section, you may have to reduce the base memory of the virtual machine. It should be no more than 75% of the memory of the host machine. For example, if you have 2G RAM, set the VM base memory to 1024MB (1G). If you have only 1G RAM, set the VM base memory to just below 768M.
- 9. Also in the settings, select the Shared Folders section. Click the plus-folder icon on the right. Use the file chooser to navigate somewhere (not within Program Files for Windows users!) and create a brand new folder called cs643host. Select that folder for the "Folder Path" in the dialog. For the Folder Name, use just the word host (must be all lowercase letters).



10. Apply the settings and **start the Linux VM**. You may have to dismiss a dialog about capturing the keyboard or mouse; these things should resolve themselves once the VM is running. You will see the Ubuntu logo (on black background) and then a little progress animation



- 11. Once the system finishes booting, Firefox should appear. At the top of the (virtual) screen, you will see a menu with "Applications, Places, System", then icons for Firefox, gEdit, Terminal, then the date and some icons for speakers, networking, and the "adminuser" identification all the way at the right. Now we need to test a few things.
- 12. Make sure that networking works by using Firefox (within the VM, not on the host!) to do a Google search. If the network seems to be down (but is working on your physical host machine) then you may have to look for the VirtualBox networking setting and change it from NAT to Bridged. The networking status icon on Ubuntu may also help you.
- 13. Next, we will check that you can connect to the shared folder on your host system. Open up the terminal application and type **exactly** the following (copy/paste to make sure you get the spacing and back-quotes correct). It should ask for your password, which is adminuser (the same as your Linux login name).

sudo mount -t vboxsf -o uid=`id -u` host /mnt/host

14. If the previous command does not complain, then try:

touch /mnt/host/testing.txt

15. If that does not complain, then check the cs643host folder you created on your host machine, and an empty file called testing.txt should have appeared there. You are good to go!

If any errors occurred in this section, contact me ASAP, and do not move on to part 2. Tell me exactly which step failed, and what happened, with exact error messages. You should learn how to take a screen capture on your system, so that you can capture all the details in long error messages. Telling me "it didn't work" with no other details will not get the problem solved quickly!

Check out and build GeekOS

16. On your Linux virtual machine, open up a terminal. Type the following to check out the sources for your projects. Replace YOURNAME with your Subversion user name. (The user name and password are the same ones you created on the liucs.net site.)

svn checkout --username YOURNAME https://liucs.net/svn/cs643s11/u/YOURNAME cs643

If you see a message like this:

svn: OPTIONS of 'http//....': 200 OK

it's an **error** (despite the fact that it says 'OK'!) and probably means you mistyped the repository URL. Check carefully.

17. The checkout command will ask for your subversion password. After you type it, Ubuntu will offer to store the password in a keyring, so you don't have to enter it every time. This is a good idea. You may use another password to encrypt the keyring if you wish... Ubuntu will ask for that one periodically. Or it's okay to just hit "create" with no password, and then "Use unsafe storage."

0		reate Default Keyring _ ×
	Choose password for default keyring	
	The applic password, you need t	ation 'Subversion' (/usr/bin/svn) wants to store a but there is no default keyring. To create one, to choose the password you wish to use for it.
Password:		
Confirm password:		
		New password strength
		Create Create

- 18. You now have a working copy of the repository in your home folder. You can browse it in the Ubuntu/Gnome file browser by selecting Places → Home Folder from the top menu. All of your projects will be completed here.n
- 19. Now we will build the initial GeekOS "kernel." Use the following command in the terminal to switch to the build folder for assignment 1:

cd ~/cs643/a1/build

20. Type the following command to build the project. It should be done pretty quickly, because the amount of code is tiny (so far).

make

21. If you are doing this step long after part 1, it may have to remount the shared folder that you set up before. The Makefile will do this for you, but you might have to type the password again: adminuser. Here's a transcript of a successful build:



22. If the make completed successfully, then it should have created a file fd.img in the build/ folder, and also copied it to /mnt/host, which means it will appear in your cs643host folder on your physical machine. Look for it now. It should be a small file, just a few kilobytes.



Ensure that you can see fd.img on your physical machine before moving on.

Create a new VM to run GeekOS

What we created in part 2 is a "floppy disk image". Now we will load that onto a virtual machine, to boot from the floppy disk. The image is tiny because this GeekOS does almost nothing at the moment... it will just initialize the screen, put some colorful text on it, and then sit there in an infinite loop! Let's try it.

- 23. Switch back to the main VirtualBox window, the one with the New, Settings, Start, Discard buttons on the toolbar.
- 24. Click new to create a new virtual machine.
- 25. Click next and set the name to geekos, the OS and version to "Other".



- 26. Click next and **set the memory to 8 MB.** If it is much larger, we'll have problems later because GeekOS misinterprets the amount of memory reported by the BIOS, and then crashes trying to initialize it!
- 27. Click next and **uncheck the Boot Hard Disk** checkbox. For now, GeekOS will run from the floppy with no other disk. Just like the original IBM PC!
- 28. Click next. We'll get a warning... confirm your decision to omit a hard disk by clicking Continue. Click Finish.

- 29. Now we need to attach the floppy disk. With the new geekos VM selected, push the Settings toolbar button. Go to the Storage section.
- 30. In the storage tree, under floppy controller, you should see a floppy icon labeled "Empty". (If you don't see it, you can add a floppy controller using one of the plus signs at the bottom of the panel.) Click it. On the right side, you should see attribute drop-downs for "Slot" and "Floppy Device". Beside the device there is a folder icon, click that. (The mouse cursor is pointing to it in the image below.)



- 31. You're now in the virtual media manager. You have to tell VirtualBox where to find your floppy image. Click the add button, then navigate to your cs643host shared folder and select the fd.img file. Click select to dismiss the virtual media manager. The storage tree should now show fd.img under the floppy controller. Click OK to save this setting.
- 32. Now let's try it! With the geekos VM selected, press the Start button on the VirtualBox toolbar. It should show the usual Sun BIOS screen, and then switch to a black screen with green "Welcome to GeekOS" and a colorful message below.



33. Don't leave the GeekOS running too long, close it and power off the machine. GeekOS doesn't have the sophisticated power management that the Linux VM does, so it will start to make your computer run hot!

Make a minor change to GeekOS display, test, commit

Now, it's your turn to hack on the code for the kernel! In the Linux VM, start the gedit text editor application — it's the icon between firefox and the terminal, or on the Applications menu under Accessories. In gedit, use File/Open to navigate through the folders cs643/a1/src/geekos and open main.c. You will see a Main function in C that accesses the video memory as an array called VIDMEM.

Using what you learned in class about video memory on PCs (and the sample code in main.c, also shown below in appendix), make the kernel write your name in the **center** of the screen (horizontally and vertically... it can be approximate). Use at least one color besides gray on black.

After making a modification, switch back to the terminal and type make (you should still be in the cs643/a1/build folder there). If there are no errors, then you can im-

mediately switch to the VirtualBox VM screen and start GeekOS to run it with your changes.

Once you are happy with your modifications, you must submit your code. To commit the changes to main.c you must be in the cs643/a1 folder, **not** the build sub-folder. Put your commit message between quotes, like this:

```
cd ~/cs643/a1
svn commit -m "my first modification to an OS"
```

There are several uncommitted files in the build folder, but you should **not** add these. If you want to remove them, you can type make clean from the build folder.

You can commit as often as you want, up to the deadline. In fact, it's a good idea to commit frequently while you're working, that way you always have a backup and can use the history features of Subversion. (You can even commit changes after the deadline, but I probably will ignore them for the purpose of determining your score.)

Good luck!

Appendix: Code using VIDMEM

From a1/src/geekos/main.c:

```
for( i = 0, j = 0; message[i]; i++, j += 2 )
1:
2:
       ł
3:
         VIDMEM[j] = message[i];
         VIDMEM[j+1] = ATTRIB(BLACK, BRIGHT|GREEN);
4:
5:
      }
6:
7:
    VIDMEM[160] = 'Y';
8:
    VIDMEM[161] = ATTRIB(BLUE, BRIGHT|AMBER);
9: VIDMEM[162] = 'o';
10: VIDMEM[163] = ATTRIB(BLUE, BRIGHT|RED);
11: VIDMEM[164] = '.';
12: VIDMEM[165] = ATTRIB(CYAN, BRIGHT|GRAY);
13: VIDMEM[166] = '.';
14: VIDMEM[167] = ATTRIB(CYAN, BRIGHT|AMBER);
15: VIDMEM[168] = '.';
16: VIDMEM[169] = ATTRIB(MAGENTA, BRIGHT|GRAY);
```

Appendix: Constants in screen.h

From a1/include/geekos/screen.h:

```
1:#define BLACK02:#define BLUE13:#define GREEN24:#define CYAN35:#define RED4
```

```
6: #define MAGENTA 5
 7: #define AMBER
                    6
 8: #define GRAY
                    7
 9: #define BRIGHT 8
10: #define ATTRIB(bg,fg) ((fg)|((bg)<<4))</pre>
11:
12: #define NUMCOLS 80
13: #define NUMROWS 25
14:
15: #define TABWIDTH 8
16:
17: #ifdef GEEKOS
18:
19: /*
20: * VGA hardware stuff, for accessing the text display
21:
     * memory and controlling the cursor
22:
     */
23: #define VIDMEM_ADDR 0xb8000
24: #define VIDMEM ((uchar_t*) VIDMEM_ADDR)
25: #define CRT_ADDR_REG 0x3D4
26: #define CRT_DATA_REG 0x3D5
27: #define CRT CURSOR LOC HIGH REG 0x0E
28: #define CRT_CURSOR_LOC_LOW_REG 0x0F
```