Lab 1

Kevin Martin
CIS657 Monday @ 10:00pm EST
Syracuse University

January 19, 2020

1 Introduction

Before I begin, I would like to state that I did not follow the recommended installation instructions.

I am running Linux on my home computer, a distribution called Arch. It is a somewhat "minimalist" version, and while popular, is not one of the major distributions. As such, I wanted to make sure I could at least get Xinu up and running, so I downloaded and configured before the first lives session.

I used the VirtualBox version that is in the Arch User Repository (AUR), which happens to be version 6.1.0-1. The recommended method of installation for Arch programs is having a package manager take care of everything from the AUR.

I am using the correct version of Xinu, from the shared Google Drive link. Other than my version of VirtualBox, the guide and assignment worked as intended. As such, I believe that there shouldn't be any issues going forward with this project, but I understand there exists the potential for complications because of this.

2 Configuration

I followed the instructions from Lab 1, and was able to untar the xinu-x86-vim.tar.gz and compile the Xinu kernel using the make commmand:

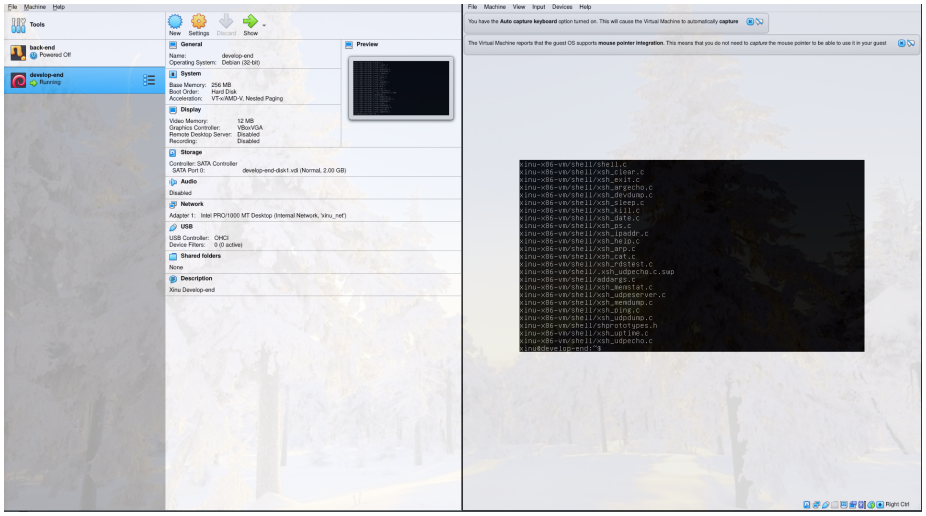

Figure 1: Successful untaring.

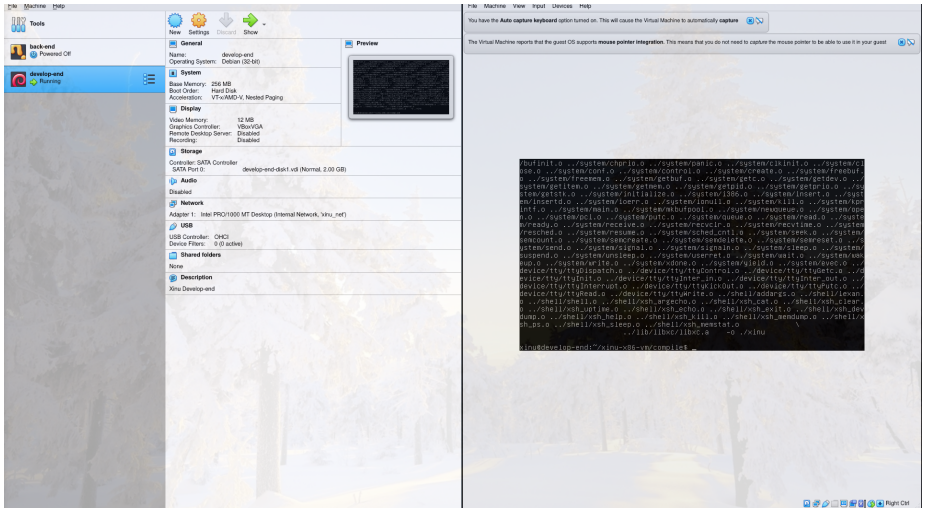

Figure 2: Successful compilation.

Finally, I was able to issue "sudo minicom", turn on the back-end machine, and display the big Xinu logo:

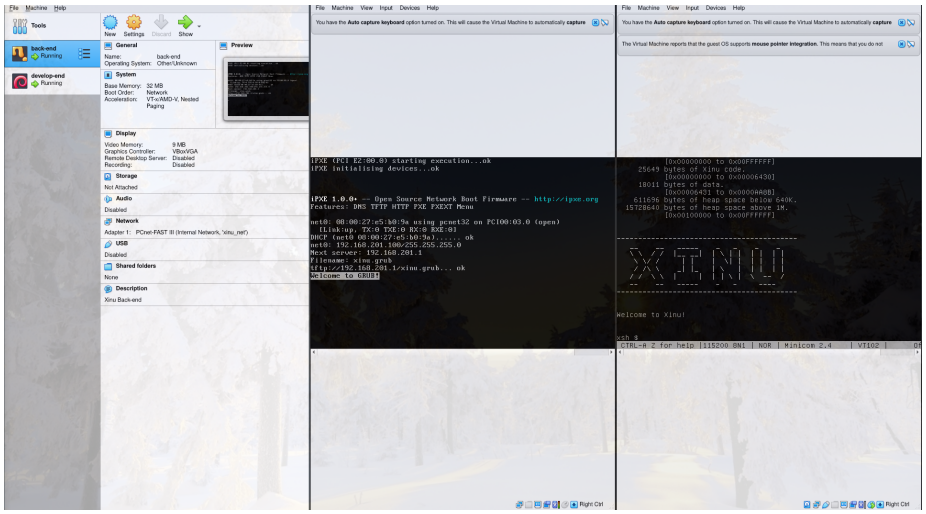

Figure 3: Xinu rocks!

3 Part I

After exiting the minicom, I was able to navigate around the file system using the familiar Linux commands.

Following the directions, I auto-mounted a folder from my host computer to a newly created "shared" folder. I copied "main.c" with the command "cp" to make sure I could see it in both places:

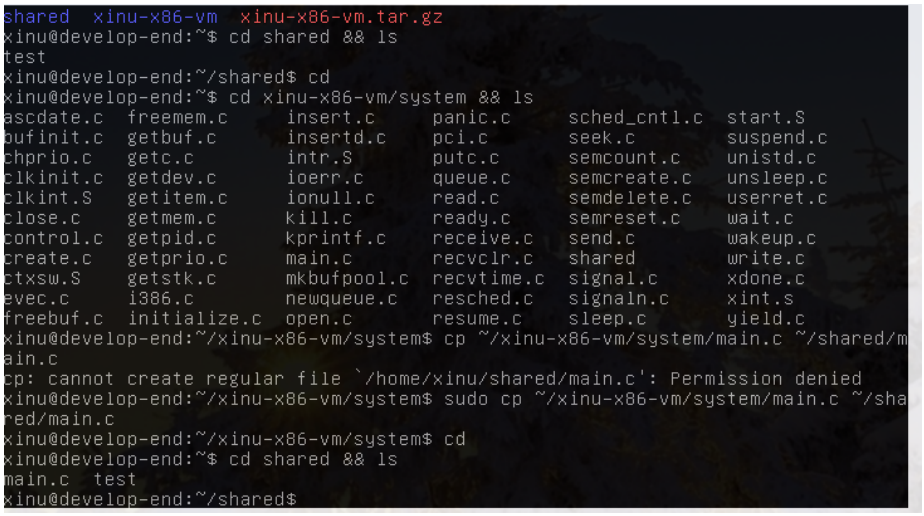

Figure 4: Main.c present in the shared directory.

4 Part II

To investige the files "main.c" and "queue.c", I used the text editor Vi, which came shipped with the Xinu (Linux) kernel. I use Vim as my text editor for C/C++ files, and recently LaTex. [1] This made navigation very familiar:

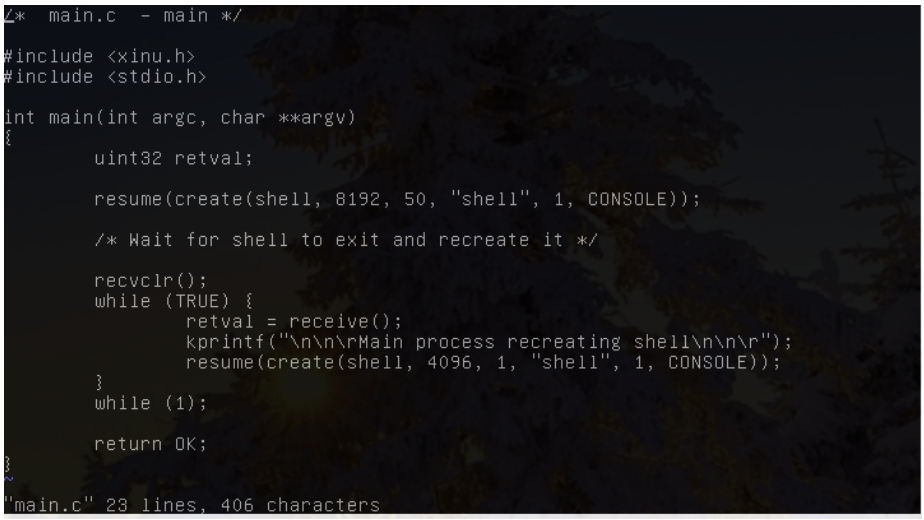

Figure 5: Main.c, opened with Vi inside Xinu.

Question 1. *While looking through "main.c", it appears to simply create a terminal, using the resume/create process. This makes sense as on startup, all the user would have access to or need would be a terminal.*
On inspection of "queue.c", it appears to provide the framework for working with the queue table (queuetab). Apart from simple "enqueue" and "dequeue" processes, the functions provide checks to make sure only valid process IDs are added or removed. The functions also take care of the pointers, ensuring that a new process will point to the correct node, and, similarly, a process being removed will not break the chain of pointers.

Question 2. *The Linux command that enabled me to copy the "xinu-x86-vm" directory to the "shared" directory is:*

sudo cp -r ~/xinu-x86-vm ~/shared

"Sudo" as the user xinu is apparently not the administrator, "cp" to copy, the modifier "-r" for "recursive" which includes all the sub-directories and contents,

[1] This entire homework assignment has been written in Vim using a LaTex plugin. The two work great together!

"~" specifying the root directory starting point, the first statement after the modifier is the item to copy, and the second is the destination.

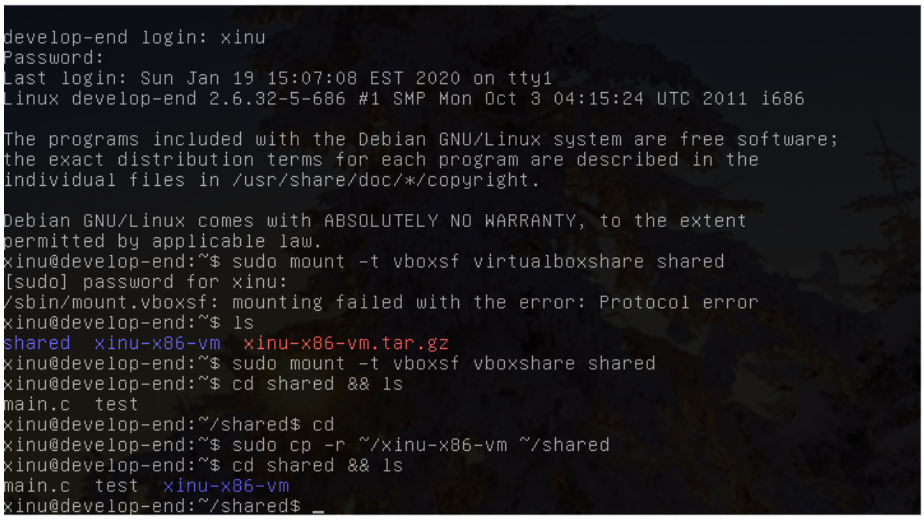

Figure 6: Full command to copy directory.

Figure 7: My host computer, showing the contents of the shared directory.

Question 3. *After I made a copy of of the "xinu-x68-vm" direcotry, I was able to recompile the "upload.sh" script, and see the Xinu logo once again:*

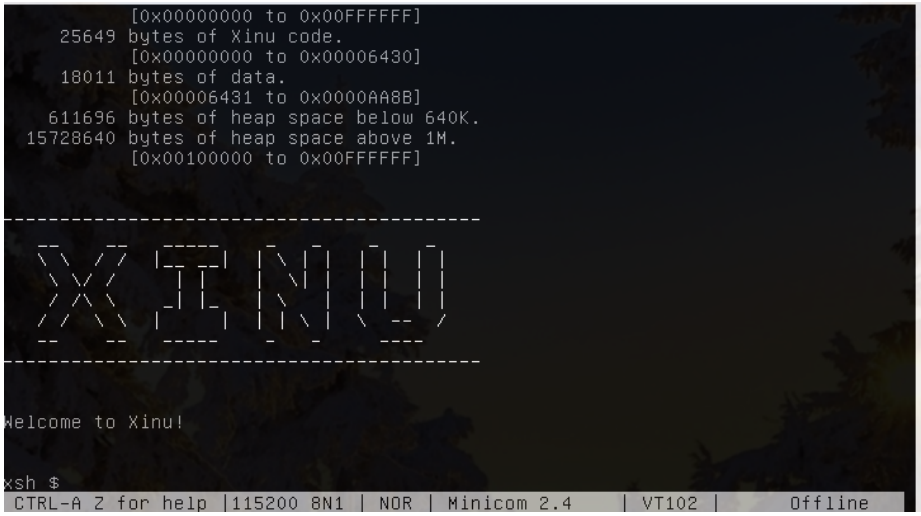

Figure 8: Successful Xinu logo.

5 Part III

To edit "main.c", I chose to simply edit it using Vi within Xinu. I have a copy of the entire directory stored on my host computer, so just in case I needed that code again, I would have it. To further protect myself, I just added the required text above the orginal "main.c" in the file, with block quotes around it (using /* and */):

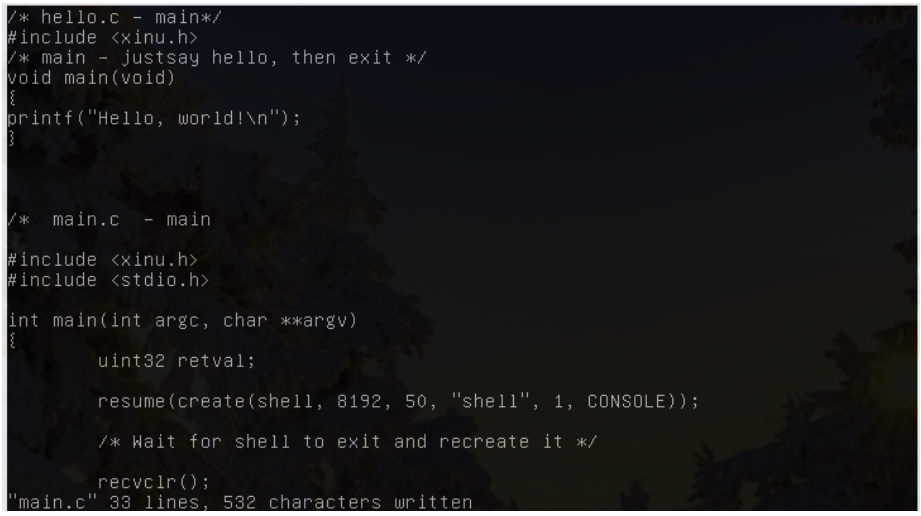

Figure 9: Main.c edited with Vi, in Xinu.

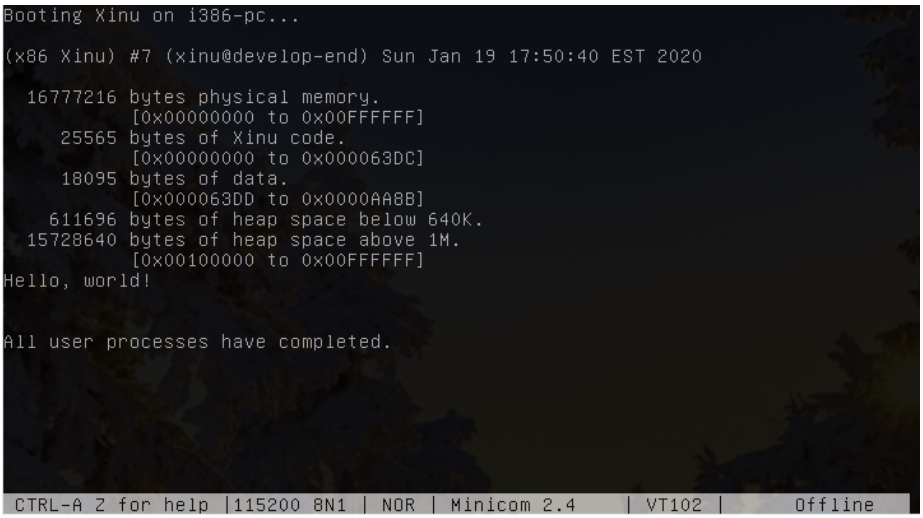

Figure 10: Successful "Hello, world!" after restarting the minicom.

6 Part IV

Question 4. *To edit "main.c" this time, I used the copy on my host machine (while keeping the original function commented out). My Vim text editor has error checking for C built in, so it was easier for me to check for errors before compiling. After saving on my host machine, I copied the file out of the shared directory and into the system directory, and the recompiled:*

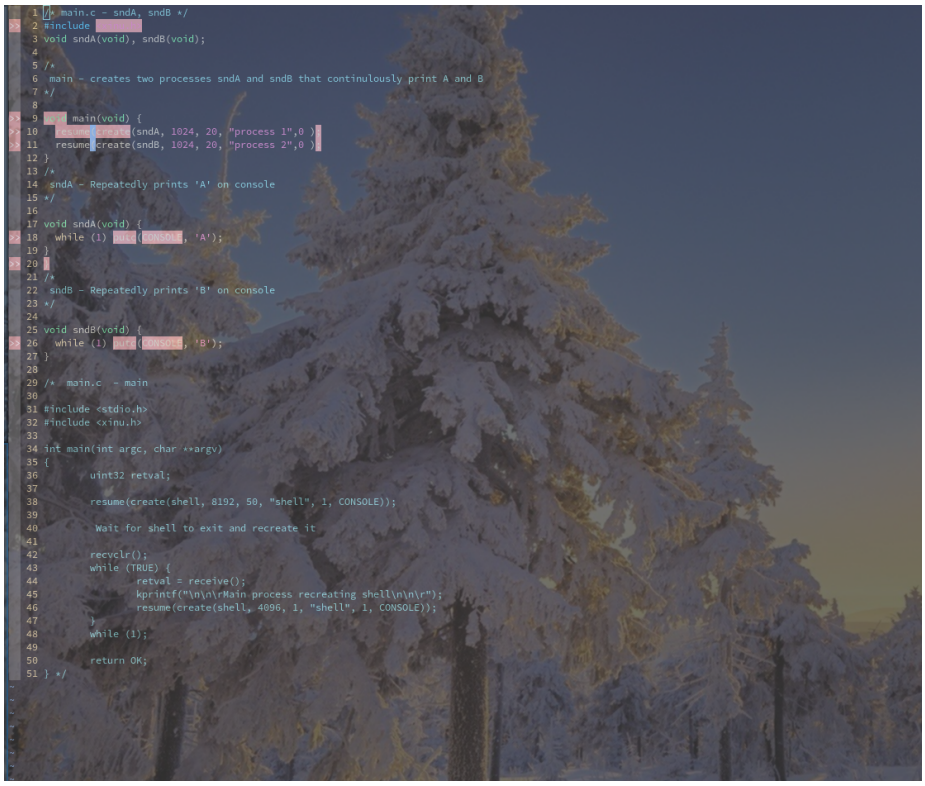

Figure 11: Main.c edited with Vim on my host machine.

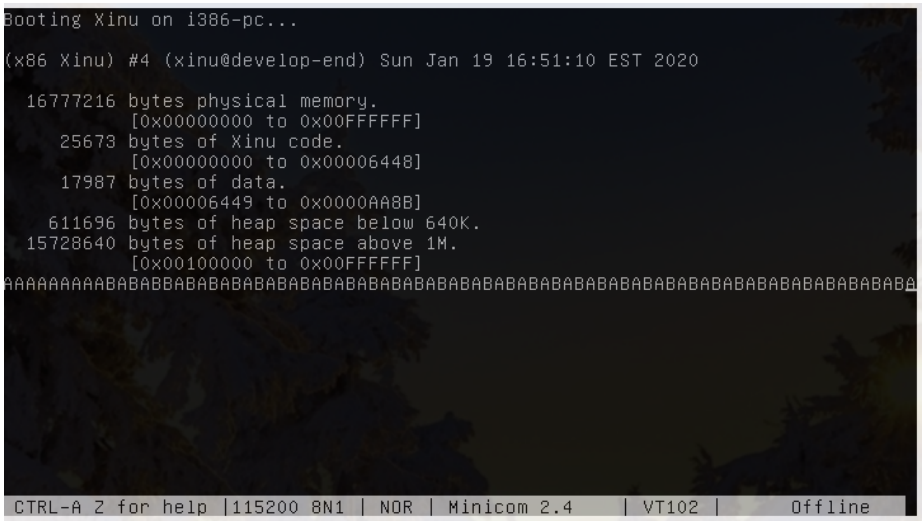

Figure 12: Output of newly modified main.c.

7 Part V

Question 5. *After witnessing the relatively similar output of As to Bs, I assume changing the priortiy of sndA from 20 to 40 would make the program print more Bs. My guess is the lower number (i.e. closer to 1), the higher the priority.*

Question 6. *I modified the main.c file direclty in Xinu again, changing the priority of sndA to 40. The result is almost completely As:*

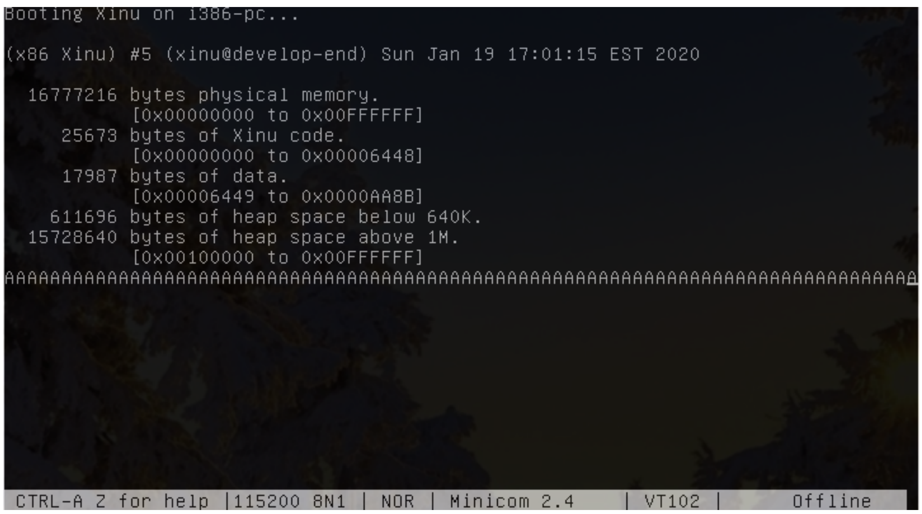

Figure 13: Output with newly changed priority.

I believe the reason there are actually more As than Bs is because the higher priority number means more weighting is given to that task's priority. Upon thinking about it further, I could see this being useful as new processes are added: if a more important process is created and we need to assign higher priority, we can simply use the max of the current highest, and increment from there.