Joseph Pennington

2912079

EECS 678 Lab03

|  |
| --- |
|  |
| 1. This command makes a differential image off the local copy of the cycle server base image. The new image is named “joseph.qcow”. |
|  |
| 1. Next, the image was booted up using 2048M of memory. |
| A screenshot of a cell phone  Description automatically generated |
| 1. After the image finsihed booting up, I added myself as a new user. |
| A picture containing clock  Description automatically generatedA clock mounted to the side  Description automatically generated |
| 1. Next, the command on the left opened the sudoers file where I added the two lines to give the new user system admin rights. |
| A picture containing clock  Description automatically generated |
| 1. Once the user was added to the sudoers file, this command added the new user to the sudo group. |
| A screenshot of a cell phone  Description automatically generated |
| 1. The next step was to move the kernel source and the config file to the user’s home directory. Then the user was granted ownership of those files. Ignore the errors when moving the files. |
| A picture containing drawing  Description automatically generated |
| 1. This command installs sudo onto the kernel. This enables the new user to call commands by using the sudo rights. |
| A picture containing drawing  Description automatically generated |
| 1. Next, I switched from the root user to the my new user I finished creating. |
|  |
| 1. This command installs libz-dev to help prevent any errors that may occur during the build process. |
| A picture containing drawing  Description automatically generated |
| 1. The directory was then changed to the home directory. Then it was changed to the “linux-2.6.32.60” directory. |
| A close up of a logo  Description automatically generated |
| 1. Once in the proper directory, I made a directory called “hello” and created the hello.c file inside the new directory. I then opened the file using vi. |
| A picture containing drawing  Description automatically generated |
| 1. Once calling vi, I wrote the script that will print “Hello world” and return the value of 100. |
| A picture containing clock  Description automatically generated | |
| 1. Next, I created a Makefile to compile the hello.c file. | |
| A picture containing drawing  Description automatically generated | |
| 1. I added the following line the Makefile to compile the hello.c file | |
|  | |
| 1. In the linux-2.6.32.60 directory, I added the “ hello/” to the end of this line in the Makefile. | |
| A close up of a logo  Description automatically generated | |
| 1. In the syscall\_table\_32.5 file, I added the last line to the system call table. | |
| A picture containing drawing  Description automatically generated | |
| 1. In the unistd\_32.h file, I added the last #define line and changed the number of system calls to 339. | |
| A picture containing drawing  Description automatically generated | |
| 1. The last change was to open the syscalls.h file and add the declaration of the system call at the end of the file. | |
|  | |
| 1. This command opens the kvm-kernel-build script. | |
|  | |
| 1. To prepare for the build process, I added the “-j 2” flag to help optimize the build process. | |
| A picture containing clock  Description automatically generated | |
| 1. Once the flag was added, I ran the kvm-kernel-build script. | |
| A screenshot of a cell phone  Description automatically generated | |
| 1. Once the build finished, I installed the kernel as the root user. | |
|  | |
| 1. Once the kernel was installed, I rebooted the system and selected the newly installed kernel. I then checked to make sure I was running the correct kernel version. | |
| A picture containing food  Description automatically generated | |
| 1. To begin the testing process, I then created the test\_syscall.c file and added the short script using vi. This script calls the syscall function on syscall number 338. This refers to the hello.c file I wrote earlier in the report. | |
| A screenshot of a cell phone  Description automatically generated | |
| 1. After writing the test file, I then compiled and ran the program. The test was successful because it returned that value of 100. | |
| A close up of a logo  Description automatically generated | |
| 1. The last test was to print the last 5 lines from the kernel logs. This test was also successful because “Hello world” was printed. | |