ComS 252 Homework 7: Compiling a custom kernel

Group assignment (check syllabus for group penalty details)

Due October 24, 2023

1 Objectives

For this assignment, you will configure, compile and install a custom kernel from source code. About 6 points of your grade is based on how *small* you can make your kernel (including modules) by turning off features. Of course, you must not turn off necessary features.

2 Build a virtual machine

1. Download the ISO file to initialize the virtual machine for homework.
2. In VirtualBox, create a new virtual machine for this assignment. You will want a lot of disk space for this machine; building *everything* required more than 25 Gb of disk space on my VM.
3. Set the ISO file as the optical disk, and boot up the VM.
4. Select "Build Hw07 virtual machine" from the boot menu.
5. After installation completes, remove the homework initialization ISO from the Optical Drive. You are encouraged to take a snapshot of the VM at this point, in case you need to roll back to a fresh install.
6. At first boot, the VM initializes itself by fetching and running a script from the server. This requires Internet access, and VPN access if you are off campus. The script will, among other things, create a user account with your ISU username. All user accounts will initially have passwords that are the account name, followed by "`pw`".
7. When the VM shuts down after initialization, you are again encouraged to take a snapshot of the VM. That way, if you make a mistake and accidentally trash the user files, you can easily roll back to a freshly initialized VM.

3 Check sound

Part of this assignment is to be sure that your custom kernel supports the sound card. First, make sure the VM is configured for sound. In Linux, you should run `alsamixer` to adjust the volume and to make sure the sound is not muted. There is a sound file in your user's home directory, under `audio/`, with extension `.ogg`. You should be able to play using the `play` utility:

```
play (soundfile)
```

When you play the file, it should be audible on your host machine (assuming of course that your host machine has working sound).

Sound can be a little finicky in a virtual machine; if the sound player hangs in Linux, try disabling and re-enabling audio output. Another thing to try is to switch the type of the audio controller in the Audio settings for the VM.

4 Obtain the kernel source

Go to http://www.kernel.org to obtain a recent, *stable* or *longterm* kernel, with version number between 5.15.128 and 6.5. The text–based web browser, `lynx`, is installed; you are encouraged to use this to download the source tarball. Note that the submission script expects the following:

- The (compressed) source tarball should be in your user's home directory; do not delete it after unpacking it.
- The tarball was unpacked in your user's home directory.
- The kernel was configured and built in the unpacked directory.
- The kernel and modules were installed in their default locations.

5 Configure the kernel

Configure the kernel, as `user`, using one of the methods discussed in lecture. You should configure a kernel that will run on, and support the hardware of, the virtual devices on the virtual machine. Similarly, your kernel should support the current filesystem types that the virtual machine uses. You do not need to support other types of hardware or filesystems. In particular, your kernel must support (at least)

- the RAM and motherboard of the VM;
- the drive(s) and filesystem(s) on the VM, otherwise your system will fail to boot;
- the network interface of the VM, so you can submit your work when you have booted from your kernel;
- and the sound card of the VM.

There are *lots* of device drivers enabled by default, and most or all of the unnecessary device drivers will need to be disabled to obtain a small kernel.

6 Name the kernel (IMPORTANT!)

After you complete the configuration, but before you build the kernel, edit the `Makefile` and set

```
EXTRAVERSION = -username
```

where `username` is the name of the user account on the virtual machine. If you are installing version 5.15.128 of the kernel, then this will cause your kernel to be named

```
5.15.128-username
```

The Turnin script expects to find the username in the kernel name, otherwise it does not think it is "your" kernel. If for any reason you need additional text, then you may add extra text after the username, separated by another dash. For example:

```
EXTRAVERSION = -username-7
```

Theoretically, you can build and install multiple kernels by changing this text (and running "`make clean`") between builds, assuming you have enough drive space.

7 Compile the kernel

Students are *strongly* encouraged to shut down the VM, and take a snapshot, before each build attempt. That way, if a kernel is unbootable, it is easy to roll-back to the machine state just after configuring, and the kernel can be re-configured.

Compile the kernel and the modules, as an ordinary user (not `root`). This can take a **long** time[1], especially if lots of kernel features are enabled. Check any "Energy Saving" settings of your host machine; you don't want to check on your kernel build hours later, only to find that your computer went to sleep after the first 15 minutes.

8 Install the kernel

Install the kernel (i.e., copy the kernel image and modules to the appropriate place). You must do this as `root`. It is *strongly* recommended that you use the appropriate `make` target(s) for this, rather than copying by hand.

9 Test your kernel

Boot into your kernel (it should appear in the GRUB menu) and make sure everything works, especially your network connection and sound card. The login screen (and the GRUB menu entry) should show your kernel name, for example:

```
Fedora 28 (Twenty Eight)
Kernel 5.15.128-username on an x86_64 (tty1)

Hw07 login:
```

Reaching the login screen is a good sign that your kernel is working, except perhaps for network and sound support. If you instead get a kernel panic, or a dracut shell, or an emergency-mode shell, then likely you will need to reconfigure to add support for some hardware or critical feature.

10 Submitting your work

From your user account, run "`sudo Turnin`" to submit your work. Again, this requires Internet access (and VPN access, from off campus), as this will collect and upload your work to the homework server.

Feedback on your submission is collected in a text file, that you can view later using "`cat submit.log`" or "`less submit.log`".

To shutdown the VM cleanly, run "`poweroff`".

[1] On my desktop machine, building *everything* took around 7 hours.