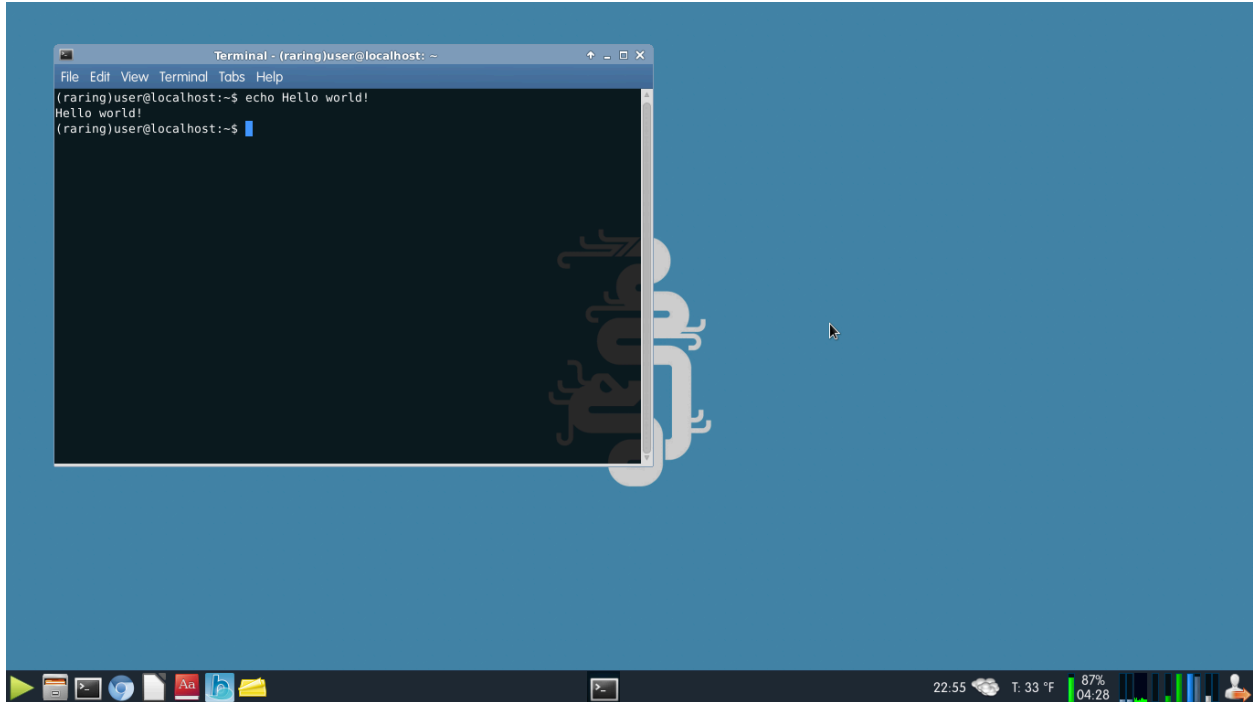*How to Use Crouton to Install an Ubuntu Chroot on the Acer C720 Chromebook*
or
How I Turned My Chromebook Up to 11 with Linux
Last Updated: February 3rd, 2015

Ubuntu 13.04 with Xfce - This can be on your Chromebook's screen in about 90 minutes.

Overview

This document describes the steps necessary to use [David Schneider](#)'s [Crouton](#) (Chromium OS Universal Chroot Environment) to install an [Ubuntu](#) chroot with the Xfce desktop environment ([Xubuntu](#)) onto an external storage device for use on the [Acer C720 Chromebook](#). At the conclusion of this sequence, you should be all set to start using Linux to the fullest on your C720. Crouton works on all other Chromebooks as well, even those with ARM processors, but keep in mind you may face limitations on the software you can use in Linux on an ARM system. This will take about 60-120 minutes depending on your skill level and experience with the process.

There are three main parts to it:

1. Transition Chromebook to Developer Mode
2. [Setup Storage Device and Run Crouton](#) (YouTube video link)
3. Update and Install Important Software in Ubuntu

<u>Follow all of the steps</u>. Pay attention to the notes in grey.

Yes, USB flash drives and external HDD/SSDs are perfectly fine!

If you are using a USB flash drive, external HDD/SSD that connects to a USB port, or micro SD card with USB adapter, you will need to replace all instances of "SD\ Card" in this guide with "USB\ Drive", but otherwise just about every step should be the same. Keep in mind, an external HDD is going to pull more power than a solid state device like a flash drive, SD card, or SSD and will reduce the Chromebook's battery endurance during use and may deliver somewhat laggy performance when it spins up. For best results, your external media options rank as follows:

1. USB 3.0 flash drive or a fast micro SD card in a USB 3.0 adapter connected via the Chromebook's USB 3.0 port
2. USB 2.0 flash drive
3. (Micro) SD card connected via the Chromebook's SD card port
4. External HDD/SSD

Warning!

Although it serves several useful functions, a chroot is inherently insecure. Once installed and running on your Chromebook, if your Ubuntu OS running in a chroot is infected by a root exploit, potentially malicious code will be able to access all hardware devices on the Chromebook--including memory. Before transitioning to developer mode and creating a chroot, you should understand that you are losing some of the "built in" security provided by the Chromebook/Chrome OS platform. Encrypt your chroot! How to do so is explained below. Practice safe browsing habits and set a different root password for your Ubuntu chroot from that you use in Chrome OS.

Part 1 - Transition Chromebook to Developer Mode

1. If new, unbox the Chromebook and check for damage.
2. Perform an initial boot to confirm that the hardware functions properly.
   (A connection to the Internet is not necessary for this step. You can log in as a guest. It doesn't really matter as you are about to wipe local user data on the Chromebook anyway. Initial issues with keyboard and/or touchpad response may be solved by rebooting w/o AC power connected.)
3. Reboot the Chromebook for transition to developer mode.
   (This will erase all user data stored locally on the Chromebook's SSD. Note that hereafter restoring the Chromebook to the normal/verified mode will similarly wipe any user data you stored on the SSD while using the system in developer mode.)
   a. Press and hold [esc]+[refresh]+[power].
   b. When shown the Recovery Mode screen, press [ctrl]+D.
   c. When prompted to turn OS verification off, press [enter].
   d. Wait for the system to transition to developer mode.
   e. When complete, the system will reboot and display this scary message:
      "OS verification is OFF"
      Press [ctrl]+D or wait 30 seconds (the system will beep loudly after 20 seconds) to boot into ChromeOS developer mode.
      (This screen will appear **every time you power the system on** while in

developer mode. If you hit any key combination other than [ctrl]+D, the Chromebook will revert to normal/verified mode and wipe user data stored on the SSD. This is a **security feature** designed to protect your data in the event the system is lost or stolen. As far as I know there is no safe way to disable this, and quite frankly you probably shouldn't anyway.)

4. Connect to the Internet and login.
5. Open a shell.
   (From the Chrome OS desktop, do this by hitting [ctrl]+[alt]+T to open a terminal window in Chrome and executing the command **shell** from the crosh> prompt. All following instructions prefixed with a $ are commands entered from the shell.)
6. $ **sudo chromeos-setdevpasswd**
   (This command prompts you to enter a new root password for Chrome OS. Select a strong password or phrase, enter, and verify it. You will need to enter this when running commands as root (usually prefixed with 'sudo') in Chrome OS. The greater vulnerability of the Chromebook's developer mode and the nature of chroots necessitate the use of a strong root password.)

On to Part 2...

Part 2 - [Setup Storage Device and Run Crouton](#) (YouTube video link)

1. Connect to the Internet and login.
   (A fast, secure Internet connection is **strongly** recommended for the remainder of these steps as the system will need to reach out and download a wide variety of software.)
2. Insert the SD card into the Chromebook's SD card port.
   (Do not insert any storage devices other than the SD card you intend to load with Ubuntu during the remainder of these steps.)
3. Open a shell.
   (From the Chrome OS desktop, do this by hitting [ctrl]+[alt]+T to open a terminal window in Chrome and executing **shell** from the crosh> prompt.)
4. **$ sudo umount /dev/sdb***
   (This command ensures the SD card and any partitions on it are unmounted from the system. The SD card itself should appear as /dev/sdb on the C720 Chromebook. It may be named otherwise on a different system.)
5. [[deprecated, skip this step]]
   (Explanation: **parted** has been deprecated from crosh as of 2015 and replaced with **partx**. Previous versions of this guide, to include the video tutorial, describe the use of **parted** at this step. You will be unable to do so.)
6. [[deprecated, skip this step]]
7. [[deprecated, skip this step]]
8. [[deprecated, skip this step]]
9. **$ sudo fdisk /dev/sdb**
10. Within fdisk:
    a. **> d**
       (This command will delete the partition currently existing on the SD card. If there are multiple partitions on the card, you will need to enter **d** more than once, deleting each successive partition with each entry until there are none remaining on the SD card.)
    b. **> n**
    c. **> p**
    d. **> 1**
    e. **> [enter]**
    f. **> [enter]**
       (These commands write a partition table that defines one new primary partition on the SD card. When it comes to defining the start and end sectors for the partition, you can hit [enter] to use default values.)
    g. **> v**
    h. **> w**
       (These commands will verify the partition table that you defined and then write it to the SD card and exit parted.)
11. **$ sudo fdisk -l**
    (Check the partition table printed by fdisk to confirm what you defined and wrote to the SD card in parted. You should see one **Linux** type partition under /dev/sdb.)
12. **$ sudo umount /dev/sdb1**
13. **$ sudo mke2fs -t ext4 /dev/sdb1**
    (These commands ensure the partition is unmounted and create an ext4 file system

within it.)

14. $ **sudo mount /dev/sdb1 /media/removable/SD\ Card**
15. Reboot the Chromebook and login. (seriously, do it)
16. Eject the SD card from Apps > Files in Chrome OS and remove it from the port.
17. Insert the SD card into the Chromebook's SD card port.
18. Open a shell.
19. $ **mount | grep "sdb"**
    (Confirm the SD card partition has successfully mounted and note its mount location. This document assumes the partition is mounted at /media/removable/SD\ Card.)
20. Download the latest crouton file from https://github.com/dnschneid/crouton.
    (The link to this file should be near the top of this page and will be labeled something like "Chromium OS Universal Chroot Environment http://goo.gl/fd3zc". Confirm the file downloaded successfully by looking in Files > Downloads in the Chrome OS.)
21. $ **sudo sh -e ~/Downloads/crouton -r raring -t xfce -p /media/removable/SD\ Card**
    (Optional: The -r [release name] flag specifies an Ubuntu release other than *precise* (12.04), which is the default release Crouton will install unless told otherwise. In the case of the above highlighted example, *raring* (13.04) would be installed. This release with the Xfce DE seems to work just fine, so by all means give it a try. If you omit the -r [release name] flag, Crouton will install *precise* (12.04). You can check which releases are supported by Crouton currently by running **sudo sh ~/Downloads/crouton -r list** from the shell.
    The current Ubuntu LTS release is *trusty* (14.04). *Trusty* seems to work just fine with crouton, so by all means change **-r raring** to **-r trusty** in this step if you would prefer to install that more recent version of the OS.
    Once executed, this command will start a lengthy (~30 minutes on a fast residential Internet connection) bootstrapping and installation process to set up an Ubuntu chroot with Xfce desktop environment, requisite dependencies, and support software.)
22. When prompted, provide a user name and root password for use in Ubuntu.
23. $ **sudo sh -e ~/Downloads/crouton -u -e -k ~/Downloads -p /media/removable/SD\ Card -n raring**
    (This command will instruct Crouton to update the newly created chroot to ensure it is as current as possible. This should take far less time than the initial install.
    -e will instruct Crouton to encrypt the chroot. The result is that your entire Linux install will be encrypted. Crouton will ask for a passphrase and confirmation when you run it with this flag and will then generate entropy from mouse movements to create a key file. It will store the key file, which is absolutely necessary to decrypt and use the encrypted chroot, at the location you specify with -k [location]. The key file save location can be anywhere you like, and best practice would be to save it to a physical storage location separate from that which contains the encrypted volume, but keep in mind you won't be able to use the chroot if you lose the key file.
    The location specified after -p should be the same as that specified after -p in Step 21. This is the location of the ./chroots directory. If you installed the chroot to the Chromebook's internal SSD, the default location should be /usr/local
    Note that in the above command, *raring* is specified after the -n flag. If you installed a different release than *raring* or if you gave the chroot a name other than the default, you must specify the appropriate name here. The default chroot name is the name of the release.
    If you installed *trusty* instead of *raring* in Step 21, change **-n raring** to **-n trusty** in this

step.)
24. $ **cd /media/removable/SD\ Card/bin**
25. $ **sudo bash startxfce4 -b**

(After entering the Chrome OS root password (and encryption passphrase if you encrypted the chroot), this command will start the chroot and load the Xfce desktop environment for the first time. At this point, Ubuntu has been successfully installed on the SD card. Unless you already know what you're doing, it is probably a good idea to select 'default configuration' when prompted for your panel preferences upon initial start up.)

(Once Ubuntu is running, you can switch between it and Chrome OS seamlessly by pressing [shift]+[ctrl]+[alt]+[back] or [shift]+[ctrl]+[alt]+[forward]. Note that you *do not* have to keep the Chrome tab with the shell running the chroot open. To stop the chroot, log out from Ubuntu.)

(Optional: If you wish you can use startxfce4 -b to run the chroot in the background. This will simply return you to the chronos@localhost prompt in the shell after starting the chroot and will not display debug information in that tab.)

On to Part 3...

Part 3 - <u>Update and Install Important Software in Ubuntu</u>

1. Open a terminal within Linux.
   (If you do not see a launcher icon for the terminal after entering the chroot, you can open a terminal by pressing [alt]+[forward] and typing "xfce4-terminal" into the search field. All following commands prefixed with a $ are executed from the terminal in Linux.)
2. $ **sudo apt-get update && sudo apt-get upgrade**
3. Log out of Linux, wait for the script to stop, restart the chroot, and open a terminal within Linux.
4. $ **sudo apt-get install software-center**
   (This command will download and install Ubuntu Software Center and Update Manager software. This step is included because many users prefer the familiarity of the Software Center over installing software from the CLI. Software Center is *not* necessary if you would rather apt-get your programs. Some secondary functions that are bundled with Software Center, such as apt-add-repository, however, may need to be manually installed if you do not want this program.)
5. $ **sudo apt-get install synaptic**
   (This command will download and install the Synaptic Package Manager. This is a pretty good tool even if you prefer to apt-get your software!)
6. $ **sudo apt-get install openjdk-7-jre**
   (This command will download and install a Java Runtime Environment.)
7. Use Ubuntu Software Center (or apt-get) to install the following software:
   (If you installed Ubuntu 13.10 (saucy), it may be necessary to run Ubuntu Software Center as root from the command line interface. If you encounter problems installing software with it, particularly an authority denial, open a terminal and execute the following command:
   $ **sudo software-center**)
   a. Ubuntu Restricted Extras (*sudo apt-get install ubuntu-restricted-extras*)
   b. VLC Media Player (*sudo apt-get install vlc browser-plugin-vlc*)
8. Use Synaptic Package Manager to install the following packages:
   a. pulseaudio
      (This package may already be installed on Ubuntu 13.10)
9. $ **sudo apt-get install gtk2-engines-pixbuf**
10. $ **sudo apt-get update**
11. Log out of Linux, wait for the script to stop, and restart the chroot.
12. Report your successes and failures, stay up to date on the latest changes, and share tips with other Crouton and Chromebook users at:
    a. www.reddit.com/r/Crouton
    b. www.reddit.com/r/chromeos
    c. https://plus.google.com/communities/109120069102230291151
    d. https://plus.google.com/u/0/communities/112113270537114386848, and
    e. https://github.com/dnschneid/crouton/issues (for reporting issues to the devs)

Enjoy!