

Chromebrew

Making Chromebooks useful.

Define: Chromebrew

github: Chromebrew

- package manager for ChromeOS
- All packages are compiled
- Open source
- Easy to add to
- Each module is programmed in ruby.

Why?

- Add packages
- It's cheap
- Use that old chromebook you have laying around.
- Programming on a long-haul flight (11hr battery life)
- Run packages natively
- Quicker to start.
- Good for web programmers

Limitations

- Cannot run GUI applications
- Limited amount of packages in repository so far. (mostly programming tools)
- Stuck with the ChromeOS kernel
- Have to compile, so it takes some time to install packages.
- Chromebooks go to sleep when closing the lid This means any services will be suspended including your SSH connection.
- Keyboards suck on Chromebooks.
- Memory is usually limited.

My hardware

Lenovo N20 (2014)

- 11.6-inch, 1,366 x 768 display
- Intel Celeron N2830 processor Dual core 2.16Ghz (Turbo 2.4Ghz)
- 2GB RAM (Not upgradable)
- 16GB storage
- SD card slot
- Bluetooth 4.0
- WIFI 802.11 a/b/g/n/ac
- 1.3kg
- About \$300 new
- Silent, no fans.
- Runs kernel 4.4 under ChromeOS

New model

Lenovo N22 (basically the same but faster)

- Intel® Braswell N3050 Processor
- 720p HD 1 MP Rotatable Camera
- 2 GB LP-DDR3 On-board (Max. 4 GB)
- 16 GB eMMC
- Up to 14 Hours
- Display 11.6" HD (1366 x 768) TN
- 1.2kg
- 2 x USB 3.0, Card Reader, HDMI
- 2 x 2 Intel® WiFi a/c 7265
- Bluetooth® 4.1
- \$299

Requirements

- A Chromebook in dev mode
- All Chromebook architectures are supported x86 64bit & 32bit, arm (Exynos 5)

Setup

- Quick (30minutes in total)
- Easy

Enter dev mode

Go into dev mode 'Ctrl+F3+power', which will factory reset the device into development mode. So do this on a new chromebook, or backup beforehand. Make sure on the first screen you click Enable Developer mode

Boot up for the first time, hit Ctrl+d to skip the screen Login as your google user

Set a password for default user

1. Hit Ctrl+Alt+F2 and login as root with the root password you setup during developer mode.
2. Type `chromeos-setdevpasswd`
3. Enter your new password
4. Hit Ctrl+Alt+F1 when you are done.
5. Reboot

Installation

- Ctrl+Alt+t

- Type shell

```
crosh> shell
```

- Download the script

```
wget -q -O https://raw.githubusercontent.com/skycocker/chromebrew/master/install.sh
```

- Review the script

- Run the script as root.

```
su  
./install.sh
```

Installing htop

I had to install the following so any ncurses program would work (like htop).

```
crew install glibc223  
crew install htop
```

Update & upgrade

Update package listing

crew update

Upgrade crew

crew upgrade

Usage

```
crew <command> <package> <keep[temporary files]>
```

Interesting Packages

- emacs
- git
- ruby\latest
- node\current
- clisp
- postgres
- openssh
- nmap
- ncd
- htop
- screen
- qemu

Making a package

All installs are programmed in ruby.

1. Fork the chromebrew on github
2. Copy an existing module and modify
3. Test it out
4. Create a pull request.

VIM module example

```
require 'package'

class Vim < Package
  version '7.4'
  source_url 'ftp://ftp.vim.org/pub/vim/unix/vim-7.4.tar.bz2'
  source_sha1 '601abf7cc2b5ab186f40d8790e542f86afca86b7'

  depends_on 'ncurses'

  def self.build
    system "make"
  end

  def self.install
    system "make", "DESTDIR=#{CREW_DEST_DIR}", "install"
  end
end
```

References

- [chromebrew/tig.rb at master · skycocker/chromebrew · GitHub](#)
- [Developer Information for Chrome OS Devices - The Chromium Projects](#) This details which kernel you will be running and how to get into dev mode on your chromebook.

Questions

| | |
|-------|--|
| Email | map7777@gmail.com |
|-------|--|

| | |
|---------|--|
| Twitter | @map7 |
|---------|--|

| | |
|--------|--|
| Github | github: map7 |
|--------|--|