Devuan Installation

My Devuan installation notes.

Latest Devuan Installation

202204100749

Using Devuan Chimaera DVD ISO (dd onto USB) → UEFI boot!

Note: If using netinstall or CD1 some firmware may not be available - which is a problem if using WiFi module.

The `must-haves` for MY Devuan installation:

- install `build-essential` (development tools)
- install `linux-headers-amd64` (to compile kernel modules)
- install `geany`, `git`, `gitk` (coding stuffs)
- install `libnss-mdns` & `avahi-daemon` (system management) **INSTALLED BY DEFAULT**
- install `curl` (some of my scripts needs that...)

The stuffs I currently need:

- install `cryptsetup` (encrypted partitions)
- install `xfig` (i need xfig2dev)

Note: these are required to compile Linux kernel: *libncurses5-dev build-essential libssl-dev libelf-dev git bison flex* (need to check if these are already installed by the above selection, e.g. build-essential and git are covered)

work in progress...?

Last XFCE Installation

- using default xfce desktop
  - this is my primary choice, but I have installed cinnamon on other machines
- replace wicd with network-manager
  - `# apt install network-manager network-manager-gnome`
  - *`network-manager-gnome`* provides `nm-applet` (not needed if using `nmcli`)
  - `# apt purge wicd wicd-gtk`
  - `# apt autoremove`

Minimal Desktop Installation

I want a basic installation with dwm.

- install base (use netinstall iso... or, maybe use debootstrap?)
- install my usual stuffs

```
# apt install build-essential linux-headers-amd64 vim git curl
```

- install xorg stuffs

```
# apt install xorg libx11-dev libxft-dev libxinerama-dev
```

- install suckless stuffs

```
# apt install stterm suckless-tools
```

- install `dwm` from source
  - use my1ubuild script
- looks good, but i need acpi stuffs

```
# apt install acpid
```

  - for laptop, maybe add `acpi-support`
  - if need command-line utility, add `acpi`

work in progress...

Devuan upgrade (chimaera to daedalus)

Personal note - basically from devuan.org.

devuan_upgrade.txt

```
upgrade chimaera to daedalus (from devuan.org)

- update/upgrade chimaera
# apt update
# apt upgrade

- update apt sources.list
# sed -i 's/chimaera/daedalus/g'  /etc/apt/sources.list

- update pkg list from daedalus
# apt update

- kill screensaver (if running)
# killall xscreensaver

- upgrade/dist-upgrade
# apt upgrade
# apt dist-upgrade

- in case of failures, fix and rerun
```

```
# apt -f install
# apt dist-upgrade

- cleanup
# apt autoremove --purge
# apt autoclean

deb http://deb.devuan.org/merged daedalus main non-free-firmware non-
free contrib
deb http://deb.devuan.org/merged daedalus-security main non-free-
firmware non-free contrib
deb http://deb.devuan.org/merged daedalus-updates main non-free-
firmware non-free contrib

#deb-src http://deb.devuan.org/merged daedalus main
#deb-src http://deb.devuan.org/merged daedalus-security main
#deb-src http://deb.devuan.org/merged daedalus-updates main
```

Install Using debootstrap

- boot using my1live-devuan
- need to install gisk debootstrap
  - optionally, install lvm2
- run

```
# apt install gisk debootstrap lvm2
```

- prepare disk (/dev/sda) layout
  - 1 uefi partition (ef00)
  - 1 root partition (8300)
  - 1 home partition (8e00/8300)
  - 1 swap partition (8200)
- run

```
# gdisk /dev/sda
```

- format/mount root partition

```
# mkfs.ext4 -L MY1BOOT /dev/sda2
# mount /dev/sda2 /mnt/disk
```

- run debootstrap

```
# debootstrap chimaera /mnt/disk http://deb.devuan.org/merged/
```

- while debootstrap runs, format other partitions

```
# mkdosfs -n MY1UEFI /dev/sda1
# mkfs.ext4 -L MY1BOOT /dev/sda2
# mkswap -L MY1SWAP /dev/sda4
```

- will use lvm in this example

```
# pvcreate /dev/sda3
# vgcreate homevg /dev/sda3
# lvcreate -l +100%FREE -n home0 homevg
# mkfs.ext4 -L MY1HOME /dev/homevg/home0
```

- mount efi/home partitions

```
# mkdir -p /mnt/disk/boot/efi
# mount /dev/sda1 /mnt/disk/boot/efi
# mount /dev/homevg/home0 /mnt/disk/home
```

- chroot and install/setup

```
# chroot /mnt/disk
# apt update
# apt install linux-image-amd64 build-essential linux-headers-amd64 vim
git lvm2
# apt install firmware-linux firmware-iwlwifi firmware-atheros
firmware-realtek
# apt install cinnamon-desktop-environment
```

- update initramfs (add lvm support)

```
# update-initramfs -u -k all
```

- i prefer all-lowercase path names

```
# vi /etc/xdg/user-dirs.defaults
```

- setup locale

```
# apt install locales
# echo "en_US.UTF-8 UTF-8 >>/etc/locale.gen"
# locale-gen
```

- setup grub

```
# apt install grub-efi-amd64
# grub-install /dev/sda
# update-grub
```

- run efibootmgr to make sure

```
# efibootmgr
```

- edit fstab

```
# vi /etc/fstab
```

- setup root password

```
# passwd
```

- change hostname

```
# vi /etc/hostname
```

- reboot

Note: *I got to Cinnamon Desktop and everything looks ok - BUT, I simply cannot run gnome-terminal! Well, I can if i run* `dbus-update-activation-environment` *first. So, I missed something coz when I reinstalled using the full dvd, everything works fine.*

work in progress...

Devuan Setups

These depends on my need when using that particular machine.

- using virtualbox from oracle (just like my slackware setup)
- using texlive (install using tlmgr)
  - alternatively, install texlive texlive-latex-extra texlive-science
- install freecad kicad openscad (project stuffs)
- install ntp ntpdate (system management)

Development

- getting *OpenGL* stuffs (glut): `# apt install freeglut3-dev`
- getting *sqlite* stuffs: `# apt install sqlite3 libsqlite3-dev`
- getting glade (will also get gtk library): `# apt install glade`
- getting *wxwidgets* stuffs: `# apt install lib-wxgtk3.0dev`
- getting *my1imgpro* stuffs: `# apt install libavcodec-dev libavdevice-dev libavformat-dev libswscale-dev`
- to compile *sdcc*: `# apt install bison flex libboost-dev texinfo`

note: Setting up mingw-w64 cross-compiler

Web Server

- Install webserver
  - `# apt install apache2`
  - default path for web is `/var/www/html`
  - edit `/etc/apache2/apache2.conf`

    - add *ServerName* (remove annoying startup message!)
    - my *my1apisrv* code need these for *www* dir config

```
    Options FollowSymLinks
    AllowOverride All
    Require all granted
```

- Install php
    - `# apt install php php-cgi libapache2-mod-php php-mysql php-sqlite3`
    - edit `/etc/php/7.0/apache2/php.ini` to enable pdo support
- Create required links in mods-enabled and conf-enabled
    - my my1apisrv code need `rewrite`
    - both folders are in `/etc/apache2`
- Install database
    - `# apt install mariadb-server`
- If running dokuwiki
    - `# apt install php-xml`
- My API client php code needs this
    - `# apt install php-curl`

Multi-Arch (a.k.a. Multi-Lib)

To run 32-bit binary:

- Enable multi-arch: `# dpkg –add-architecture i386`
- Update package list: `# apt update`
- Most probably need libc: `# apt install libc6:i386`
- Install required libraries: (<package>:i386)

To build 32-bit binary:

- Install compiler(s): `# apt install gcc-multilib g++-multilib`
    - Notice that these are 64-bit packages (no :i386 suffix) → cross compilers!
- Use `-m32` gcc option to compile!

Gaming

To play steam games:

- enable multi-arch
- install steam: `# apt install steam` (binary in `/usr/games/steam`)

NFS setup

Dumping this as it is for now:

- client

    - install

```
# apt install nfs-common
```

    - mount

```
# mount -t nfs <host>:/path <mount-point>
```

- server
    - install

```
# apt install nfs-kernel-server
```

    - [optional]

```
# mount --bind /path/to/share /mount/point
```

    - edit `/etc/exports`
    - start

```
service nfs-kernel-server start
```

Others

My iso2boot script need *isohybrid* from syslinux/isolinux project:

- `# apt install` *`syslinux-utils`*

Use Tips

Some are applicable to any APT-based distro.

Note for APT-based distro: *To remove translations, create file* `/etc/apt/apt.conf.d/99translation` *and insert* `Acquire::Languages { “none”; };`.

Note: *look into* `unattended-upgrades` *(simply* `apt install` *and run* `dpkg-reconfigure –priority=low unattended-upgrades???`*)*

List Installed Packages

Using `apt` tool:

```
apt list --installed 2>/dev/null | grep installed
```

Note that apt will issue a warning when piping its output in shell. Hence, the need to redirect `stderr` to `/dev/null`. We can further grep away the packages that were automatically installed.

Using the basic dpkg tool:

```
dpkg --get-selections | sed -n 's/^\([^\t]*\)\t.*$/\1/ p'
```

This version, however, only provides package name. To extract similar output from apt (assuming output was redirected into a file called temp.txt), run

```
cat temp.txt | sed -n 's|^\(.*\)/.*$|\1| p'
```

Adding More Repo

Get proper signature key from that source (*.asc file) and add to system using apt-key.

```
# cat <key-file.asc> | apt-key add -
```

Create a listing file for source URL in /etc/apt/sources.list.d/.

```
# echo "deb [arch=?] <new-repo-url> <version> main" >
/etc/apt/sources.list.d/new-repo.list
```

After doing an apt update, should be able to apt install <pkg>...

Reconfigure Package

Basically, run a dpkg-reconfigure <pkg>

e.g. To change timezone

```
# dpkg-reconfigure tzdata
```

Upgrading APT-based systems

To do an upgrade:

- Modify /etc/apt/sources.list and replace the release codenames

  ```
  # sed -i 's/old_release/new_release/g' /etc/apt/sources.list
  ```

  - skip -i option for a dry-run
- Update package list

  ```
  # apt update
  ```

- Upgrade distribution

  ```
  # apt dist-upgrade
  ```

- To make sure EVERYTHING is upgraded (avoids old packages held back)
  - `# apt full-upgrade`
- Do house cleaning

```
# apt autoremove
# apt clean
```

That should do it!

Search package for specific binary

- There is a specific tool for that

```
# apt install apt-file
```

- Update the package/file mapping database

```
# apt-file update
```

- Search for “top”

```
# apt-file search --regexp '/top$'
```

Avoiding marking package as manually installed

- Use dry-run (simulation) option `-s`

```
# apt install -s <pg1> ... <pkgN> 2>/dev/null|grep manually
```

- Removed the package(s) displayed from your install list and re-run without `-s`

GRUB Stuffs

To prevent GRUB from looking for other OS every time it is updated:

- edit `/etc/default/grub` and insert

```
GRUB_DISABLE_OS_PROBER=true
```

- run update-grub

To add a custom entry:

- add a menuentry in `/etc/grub.d/40_custom`
- uses the same menuentry format as in `/boot/grub/grub.cfg`

    - e.g. to boot my Slackware partition I can insert

```
menuentry 'Slackware' --class slackware --class gnu-linux --class
gnu --class os $menuentry_id_option 'my1part-<uuid>' {
    insmod part_gpt
    insmod ext2
    set root='hd0,gpt4'
    if [ x$feature_platform_search_hint = xy ]; then
      search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt4
--hint-efi=hd0,gpt4 --hint-baremetal=ahci0,gpt4  <uuid>
    else
      search --no-floppy --fs-uuid --set=root 4d95a526-2518-4fd6-
a904-f7bd2729145d
    fi
    linux /boot/vmlinuz-huge-4.4.240 root=/dev/sda4
}
```

    - use `initrd` (after `linux` line) to specify an initrd/initramfs
    - of course, <uuid> should be a valid filesystem uuid
- run update-grub

KVM Stuffs

If KVM group is missing, simply create one...

```
# [ -z "$(cat /etc/group|grep kvm)" ] && addgroup --gid 125 kvm
```

Issues

Issues... and maybe fixes (if any).

Image Magick's convert error

I found an error when using image magick's convert to create EPS from fig that provides this message:

```
convert-im6.q16: attempt to perform an operation not allowed by the security
policy `EPS' @error/constitute.c/IsCoderAuthorized/421
```

Solution:

- (as root) edit `/etc/ImageMagick-6/policy.xml`
- modify following line (change none → read|write)

```
<policy domain="coder" rights="none" pattern="EPS" />
```

GRUB Prompt

In some cases, after an installation is complete, the system boots to a GRUB prompt. The issue may be caused by a different disk assignment during installation. Things is not so bad since GRUB prompt is very shell-like (e.g. double-tab for completion):

To list available disks:

```
> ls
```

To list content of first partition on the first disk (/dev/sda1) that is using GPT:

```
> ls (hd0,gpt1)
```

So, to boot an installation on second partition instead:

```
> linux (hd0,gpt2)/boot/vmlinuz<...>
> initrd (hd0,gpt2)/boot/initrd.img<...>
> boot
```

BCM Wireless

Device:

```
# lspci -nn | grep BCM
02:00.0 Network controller [0280]: Broadcom Limited BCM43142 802.11b/g/n
[14e4:4365] (rev 01)
```

Install: broadcom-sta-dkms, firmware-linux, firmware-b43-installer (b43-fwcutter)

note: only solves the Wireless Driver part... NOT the bluetooth hardware!

```
# lsusb | grep BCM
Bus 002 Device 004: ID 0a5c:216d Broadcom Corp. BCM43142A0 Bluetooth 4.0
# dmesg | grep blue
[    8.850007] bluetooth hci0: firmware: failed to load
brcm/BCM43142A0-0a5c-216d.hcd (-2)
[    8.850197] bluetooth hci0: Direct firmware load for
brcm/BCM43142A0-0a5c-216d.hcd failed with error -2
```

So, download `BCM43142A0-0a5c-21d6.hcd` in here and place it `/lib/firmware/brcm/` (create that path if it does not exist).

XFCE Desktop

- tapping on my laptop touchpad is NOT working

    - moving around & button clicks are working
    - need to create file `/etc/X11/xorg.conf.d/90-touchpad.conf`

      90-touchpad.conf

      ```
      Section "InputClass"
              Identifier "libinput touchpad catchall"
              MatchIsTouchpad "on"
              MatchDevicePath "/dev/input/event*"
              Driver "libinput"
              Option "Tapping" "on"
      EndSection
      ```

- lid event does not trigger suspend (but manual suspend works)
    - this is a systemd-related aftermath: xfce power manager allows/assumes login to handle lid
    - fix:

      ```
      $ xfconf-query -c xfce4-power-manager -p /xfce4-power-
      manager/logind-handle-lid-switch -s false
      ```

- issues when logging out/shutdown
    - may be caused by intel graphics library? i915?
    - can install lightdm - but shutdown/restart always gets login page (=logout)
- wicd feature is an issue for system with multiple users
    - shared wifi password, no option to make private

Cinnamon DE

- login page (lightdm?) - cannot shutdown/reboot! read here...
    - found a fix here
    - i just needed to edit `/etc/pam.d/lightdm-greeter` and changed `pam_systemd.so` to `pam_elogind.so` (but, may cause issues with suspend/hibernate?)
- laptop battery quickly drain below 30 percent
    - setup higher critical value for power management

      ```
      gsettings list-keys org.cinnamon.settings-daemon.plugins.power
      gsettings set org.cinnamon.settings-daemon.plugins.power use-time-
      for-policy false
      gsettings set org.cinnamon.settings-daemon.plugins.power
      percentage-low 30
      gsettings set org.cinnamon.settings-daemon.plugins.power
      percentage-critical 25
      gsettings set org.cinnamon.settings-daemon.plugins.power
      percentage-action 23
      ```

    - or use `dconf-editor`

PulseAudio

Youtube videos keep resetting the volume settings to 100%! The culprit is `flat-volumes`. Modify `/etc/pulse/daemon.conf` and set `flat-volumes=no`.

(Re)-Compiling Syslinux

I need to recompile syslinux - just to remind myself, other than the usual development packages (build-essential, etc), I also need `nasm` and `upx-ucl`.

```
apt install nasm and upx-ucl
```

Unwanted Background Program

I found this in my old notes... I somehow found an autostart program `geoclue-demo-agent.desktop`, which I obviously do not need. So, simply remove that from autostart path

```
# rm /etc/xdg/autostart/geoclue-demo-agent.desktop
```

Sound muted on startup on my HP laptop

trying two options found... so far, not successful? keeping this here for reference.

1. option 1:
    - install alsa-utils
    - make sure sound is not muted and set to desired volume level
    - run (as root) alsactl store
2. option 2:
    - edit (as root) /etc/pulse/defaults.pa
        - can also copy this to home folder ??? for 1-user solution...
    - comment out 'load-module module-device-restore'

General Issues

- wifi firmware missing after first restart
    - manually install `firmware-ralink`
- network manager cannot connect using wifi
    - edit /etc/NetworkManager/NetworkManager.conf
    - add

```
[device]
wifi.scan-rand-mac-address=no
```

Using Wine on Devuan

Setup Wine (as root):

- enable multiarch
- install wine and winetricks

```
# apt install wine wine32 winetricks
```

Configure Wine (as user):

- if required, remove previous settings

```
$ rm -rf $HOME/.wine
```

- config for win32

```
$ WINEARCH=win32 winecfg
```

- setup nice fonts

```
$ winetricks corefonts
```

Note: Install `fuseiso` to enable mounting ISO as user

Game: Red Alert 2

Setup Installer ISO:

- create link to ISO as CDROM device

```
$ ln -sf /path/to/install.iso $HOME/.wine/dosdevices/d::
```

- mount ISO

```
$ fuseiso mount  /path/to/install.iso $HOME/.wine/drive_d
```

- create link to mount path as CDROM drive

```
$ ln -sf $HOME/.wine/drive_d $HOME/.wine/dosdevices/d:
```

Run installer

-
```
$ wine D:\\Setup.exe
```

Fixes:

- menu does not show (solution available at WineHQ)
  - download ddraw.dll from https://github.com/CnCNet/ts-ddraw/releases
  - place in RA2 install path
  - run winecfg add that to library (as window native, instead of builtin)

Application: LTSpice

- download windows version from ltspice website and run

```
$ wine /download/path/LTspiceXVII.exe
```

Devuan on Raspberry Pi

Check out here.

RasPi400

Using `rpi-devuan-chimaera-5.10.110-ext4-2022-04-16.zip`

- boots ok (default hostname=bcm2711)
- login (root:toor)
- setup

```
# run-setup
```

- change hostname (edit /etc/hostname and /etc/hosts)
- reboot
- remove default user

```
# userdel -r devuan
```

- most development stuffs i want are preinstalled! yay!
- install vim

```
# apt install vim
```

- remove nano

```
# apt remove nano
```

  - also

```
# rm .nanorc
```

- install xorg stuffs

```
# apt install xorg libx11-dev libxft-dev libxinerama-dev
```

- install suckless stuffs

```
# apt install stterm suckless-tools
```

- create user and login as that user
- get my1shell and my1ubuild

```
# git clone https://codeberg.org/azman/my1shell
# git clone https://codeberg.org/azman/my1ubuild
```

- install `dwm` from source
- install browser & font

```
# apt install surf fonts-liberation2
```

work in progress... 2 issues: {surf cannot validate cert}{reboot hangs}

Note: *Will simply use Raspberry Pi OS for now...*