Roadmap

Duzy Chan

2016-03-04

Contents

1 TODO [0/6] Making a base system :CC:Base:

The bootstrap is creating a basic root filesystem (rootfs). Further efforts should be done in the base. The base is providing an identical environment to everyone (devs, geek, hacker, etc) involved.

1.1 TODO [1/1] Preparing Root filesystem :CC:Base:RootFS:

This is where we start things up. It should boot into a shell terminal with networks and Debian APT supported.

1.1.1 Why rootfs? :CC:Base:RootFS:Why:

Base on the needs of the Community Cube system, we might not have much customization on the rootfs. I think this is good. But one the most critical reason is to have a **predictable** and **identical** base system for the team and others might be involved. This is very important to make the system unified so that contributions is happing in the same environment.

1.1.2 DONE Creating the rootfs

I think the initial rootfs is done in the bootstrap.

1.2 TODO [0/1] Creating bootable images for QEMU

We are going to use the kernel image from Debian instead building it ourselves. The bootable images are making the root filesystem to alive for usage (development and test purposes).

1.2.1 TODO Bring up the image in QEMU

1.3 TODO Network configuration

This is done in the QEMU to bring up the network manager.

1.4 TODO [/] The package manager :CC:Base:APT:

With the networking enabled for the base, a fully funtional debian package manager should be configured. This is the basic facility to extend the system to different variants.

1.5 TODO [/] Creating ISO image for distribution

We're going to use DebianInstaller to avoid too much extra workload.

1.5.1 TODO Setup Installer

We should pull in the DebianInstaller, customization might happen base on our needs.

1.5.2 TODO Making ISO image

This is building the binary image. We're going to derive the tool from Debian.

1.6 TODO [0/1] Methods to work on the rootfs (development purpose) :CC:Base:Dev:

We're going to use QEMU to quickly work on the base. This will give others (devs or someone from the community) an easy way to get involved. We also need a managed way to assembly a good result from the efforts of the community.

1.6.1 Very common packages :CC:Base:Dev:Initialization:

Some packages might be widely used by other packages or in real life. These packages could be installed before futher installation. The list of such packages can grow in the future development and needs. Hopefully the base could fit the needs.

1.6.2 TODO [0/2] A method to wrap external efforts :CC:Base:Dev:Modulization:

An external effort could be contributed by anyone from the community. We need to assembly these individual contribution. This is involving the post-installation script to make Community Cube base system ready.

- **TODO** A module manager to assembly individual efforts **:CC:Base:Dev:Modulization:Manag**
  A well designed manager to assembly and build external contribution into the system. A contribution should be wrapped into a **module**.
- **TODO** [0/1] A common way to contribute an effort **:CC:Base:Dev:Modulization:Interfaces:**
  The devs, external individuals are trying to contribute a valuable efforts to the system. The job needs to be done in a common way. Our own customization and parameterization should also be done in such way. So that we will enable different forces to work on different kind of task, and tested in the development QEMU. If some task requires to be tested in a real environment, it could be done after the integration. All these stuffs are considered by a module. A **module** could be made to bring up **iptables**, **squid**, etc.
  - **TODO** A module sample
    A sample to demonstrate how it's made and develop it in the QEMU.

1.7 [0/1] To run in a physical machine :CC:Base:Physical:

We will need the ISO image and burn it to a DVD or USB stick for installation on a Intel or AMD machine.

1.7.1 TODO Instructions for making a bootable DVD or USB stick

This should be quite easy, existing tools are out there. Just need to demonstrate how to do it for end users.

1.8 [/] Things pended for discussion

Deciding how much are going to customize it. Actually we should put all package installation to the module installation.

1.8.1 [ ] systemd

1.8.2 [ ] init.d

1.8.3 [ ] packages

1.8.4 ...

2 TODO [0/2] Extending the base to Community Cube system

This is basically working the module manager. A the modules are designed bring up a part of the Community Cube system. A module has a **install** script, which will be executed during the installation of the Community Cube system. We need to define modules to perform tasks for the features.

2.1 TODO System

2.2 TODO [0/3] Web Service

2.2.1 TODO Nginx (or Apache?)

2.2.2 TODO Web apps

This is actually defined by the business modle.

2.2.3 TODO OwnCloud

2.2.4 ...

2.3 Do we need a GUI or CPanel?

This is our user interface concern. Here, a GUI means setting up a **Window System**, say X. And a CPanel is a web interface towards a Community Cube box (a maching running our system).

What should we choose? It's decided by the end use case. A ComCube is basically running remotely, might not be touch physically by a human, so a CPanel should be prefered.

2.3.1 Security

Enabling a CPanel should put extra security concern because it's remotely controled.

2.3.2 System workloads for windowing

Enabling a **GUI**, say the **Window System**, is putting system workloads.

3 Customization and Parameterization over the base

This part will be ignored for now. This could be done by modules with the module manager.

3.1 GUI or CPanel

Refer to the user interface concern.

4 Creating an installer for the prepared distribution

4.1 Installer same as https://wiki.debian.org/DebianInstaller

(to be expended...)

4.2 Making ISO image with the Installer

Refer to "Creating ISO image for distribution". (to be expended...)