

DAY1

webOS

Open Source Edition

Introduction To WebOS

Dr. Daniel D || Mr. Praveen Kulkarni
Assistant Professor
Christ (Deemed to be University)

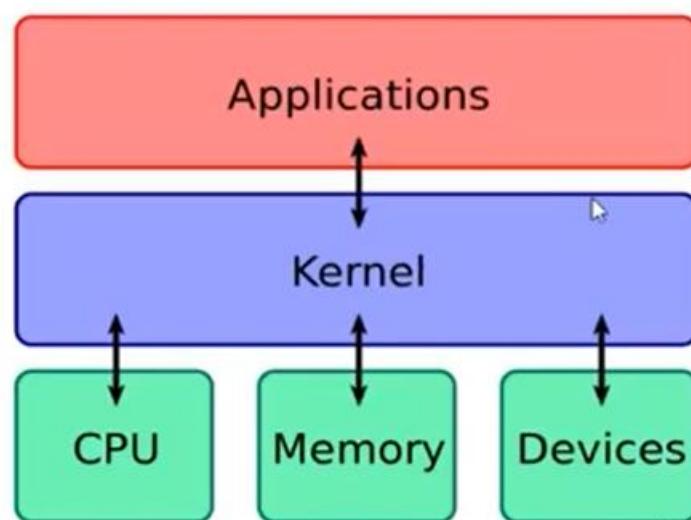
I
Mr. Harsha B M || Ms. Dalia
Engineers
CDI – Christ (Deemed to be University)

What Is An OS?

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. (Wikipedia)

- What is a Kernel?**

It is an integral part of any operating system. It is one of the first programs loaded on startup (after the boot loader). It handles the rest of startup as well as input/output requests from software, translating them into data-processing instructions for the central processing unit.



Contd...

- **System Call**

- A system call is a way for programs to **interact with the operating system**. System call **provides** the services of the operating system to the user programs via Application Program Interface(API) or System Call Interface.

- **Types of System Calls** : There are 5 different categories of system calls
 - **Process control**: end, abort, create, terminate, allocate and free memory.
 - **File management**: create, open, close, delete, read file etc.
 - Device management
 - Information maintenance
 - Communication

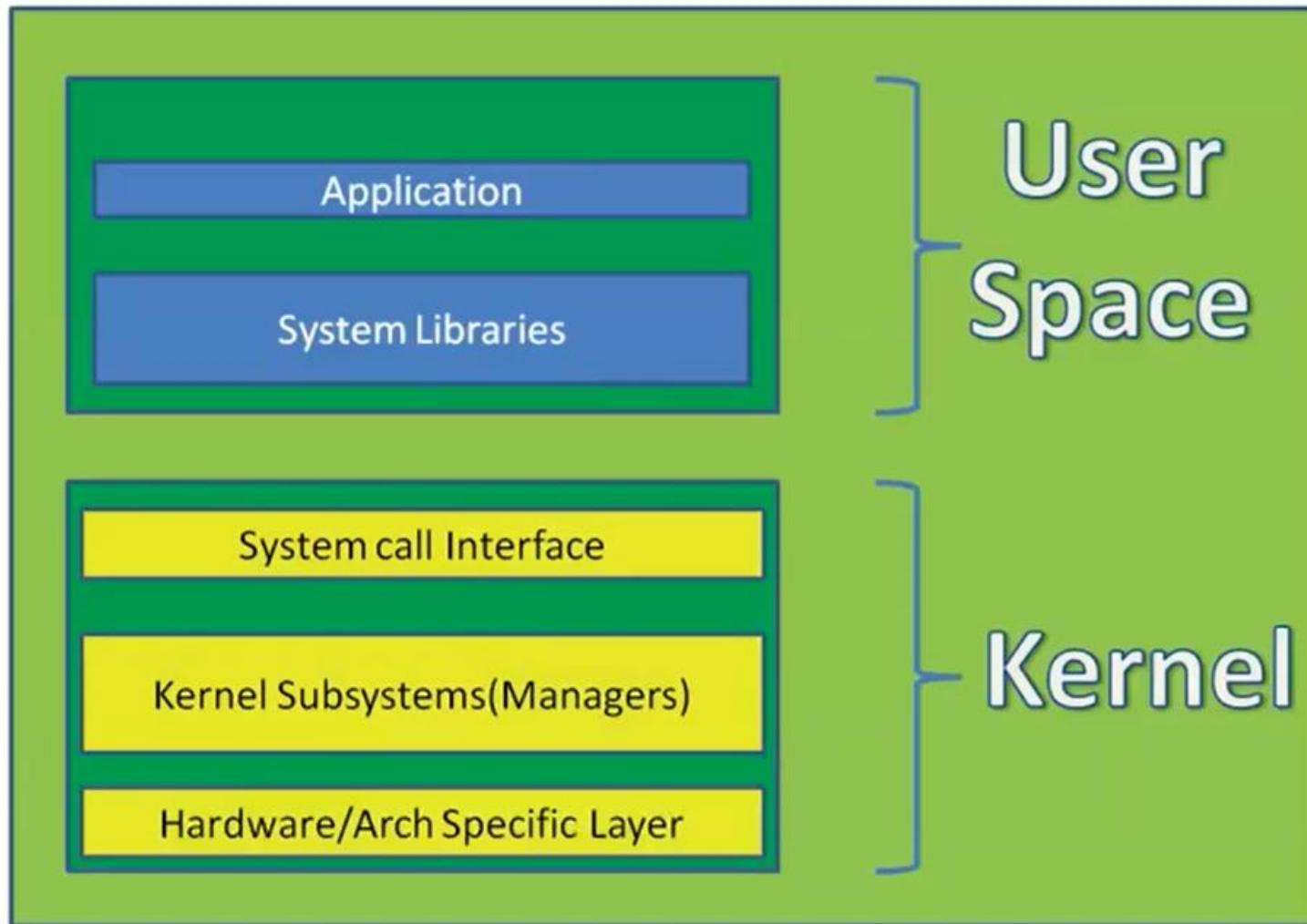
- **Necessary Managers For An OS**

- Process Manager (PM)
 - Storage/File system Manager (VFS)
 - Device Manager (device drivers)
 - Memory Manager (MM)
 - Network Manager
 - Device Drivers
 - ...etc

Examples of Windows and Unix System Calls

	WINDOWS	UNIX
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

What Is An OS?

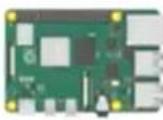


What Is webOS?

webOS is a web-centric and usability-focused software platform for smart devices, which has proven its performance and stability in over 70 million LG Smart TVs. Since its adaptation to display products, webOS has come a long way and evolved into a software platform applicable to a broader range of products.



Open
since
2018



Raspberry Pi 4
Support



Highly
Extensible



Open Source Chromium
Web Engine



Dual-Display
Support

Strategic
Partnership with
The Qt Company



Tier 3
Supported
Platform
for ROS 2



Key Features & Benefits

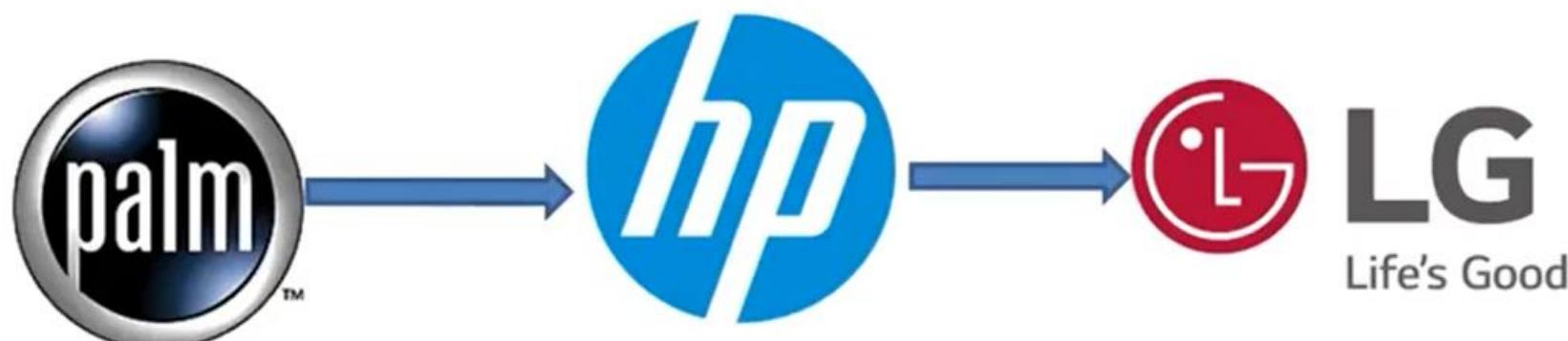
- Support for a wide array of development technologies
- Extensible and pluggable architecture
- Integration with up-to-date open source technologies
- Open platform applicable to diverse verticals

Applications

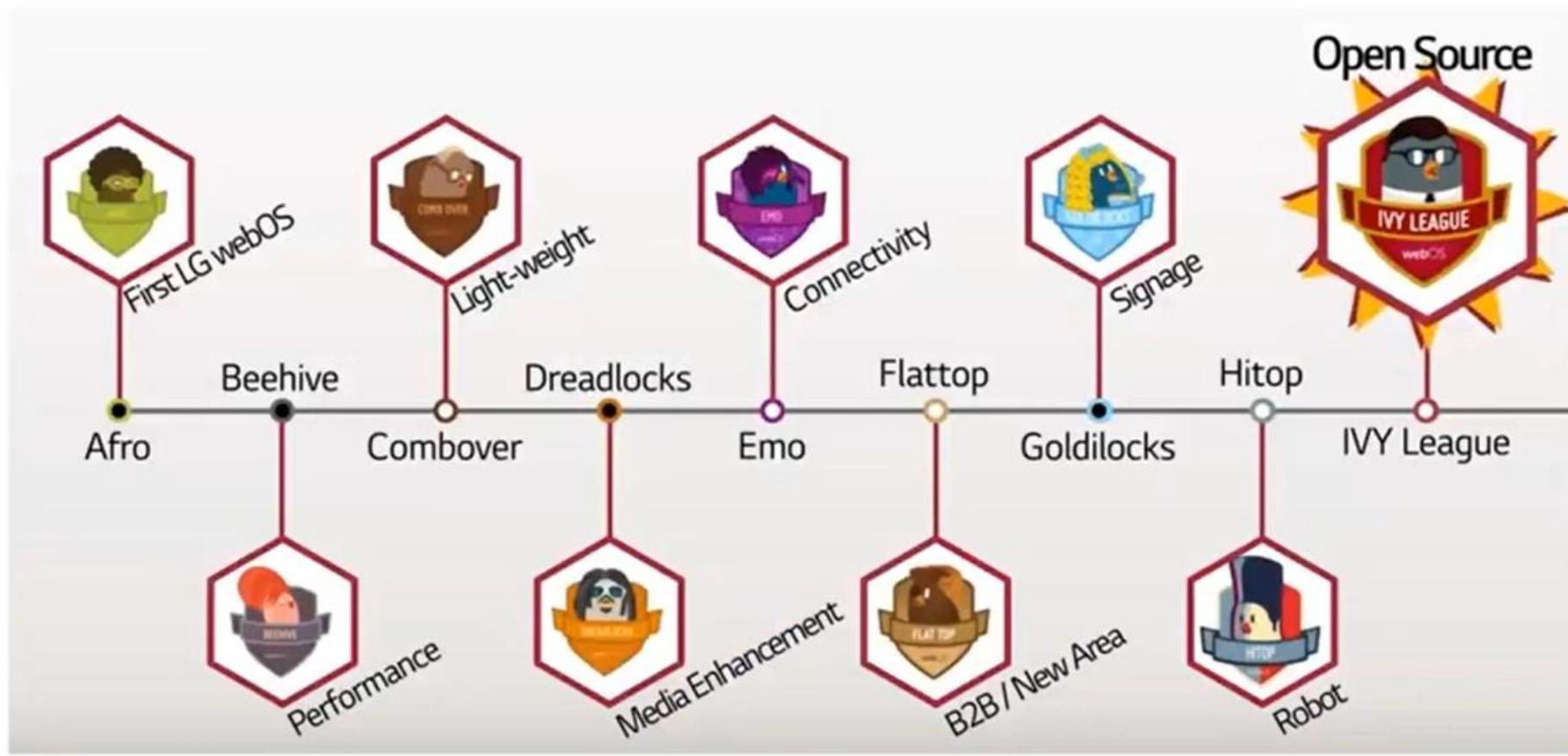
- Smart TV
- Smart Fridge(Home appliances)
- Smart Watch
- Smart Projects
- Palmtops & Tablets(
- Digital displays
- Robotics & Automobiles

History of WebOS

- WebOS is a Linux operating system for smart devices, and formerly a mobile operating system. It was initially developed by Palm, which was later acquired by Hewlett-Packard, and then LG Electronics. Palm launched WebOS in January 2009. Various versions of WebOS have been featured on several devices, including Pre, Pixi, and Veer phones and the HP TouchPad tablet. The latest version, 3.0.5, was released on 12 January 2012.



WebOS Evolution



WebOS Evolution



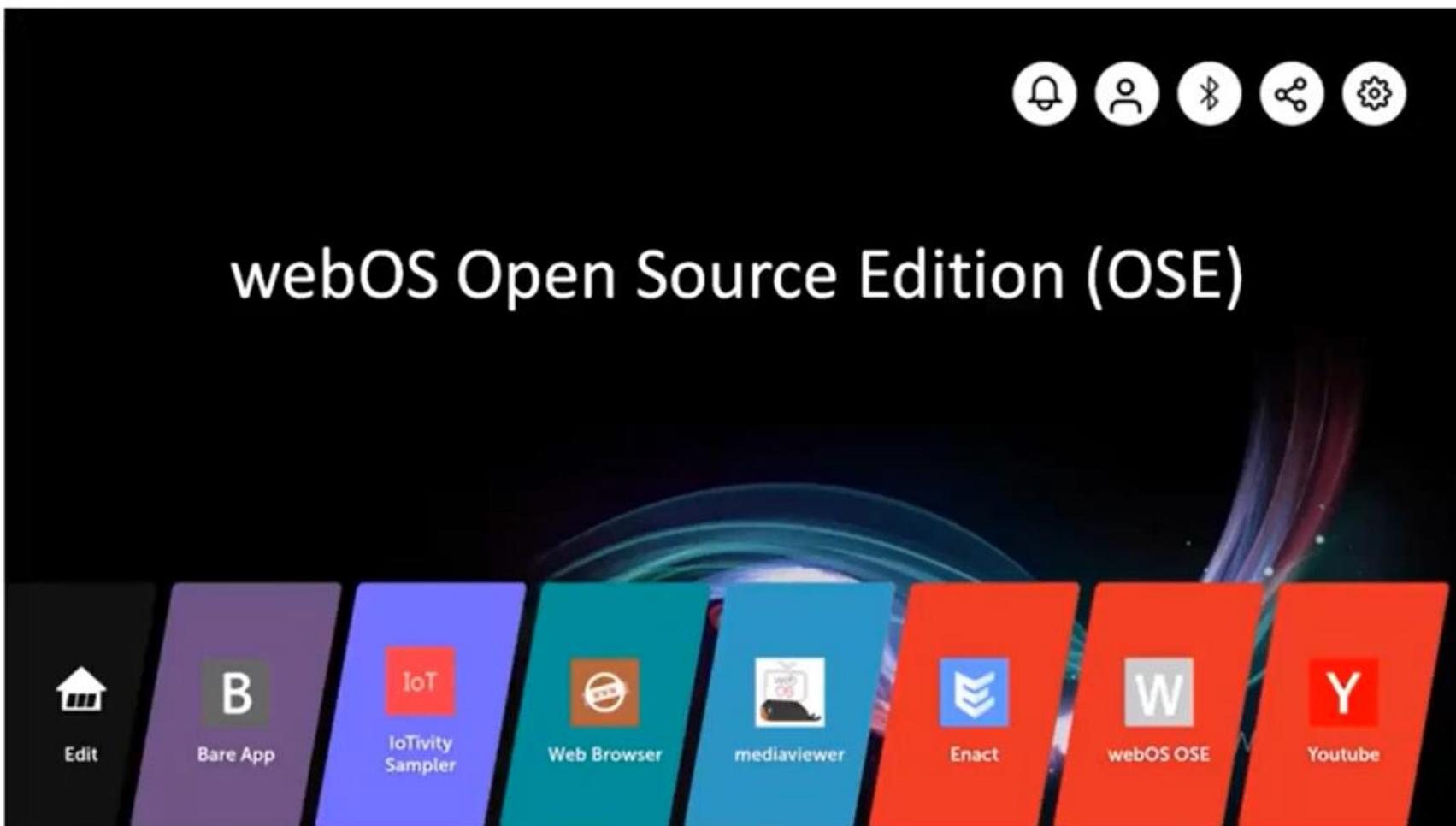
Extensible webOS



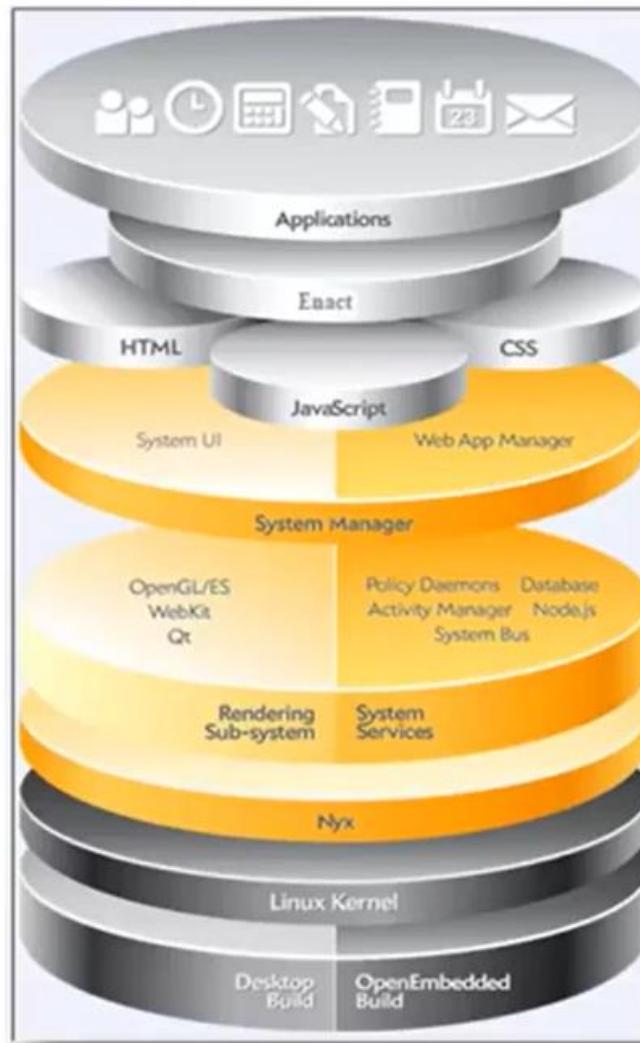
Can we customise the webOS?

- webOS OSE uses Yocto Project's Open Embedded build system as a part of its development environment in order to enhance flexibility of code management through customization and layering.
- The Yocto Project (YP) is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.

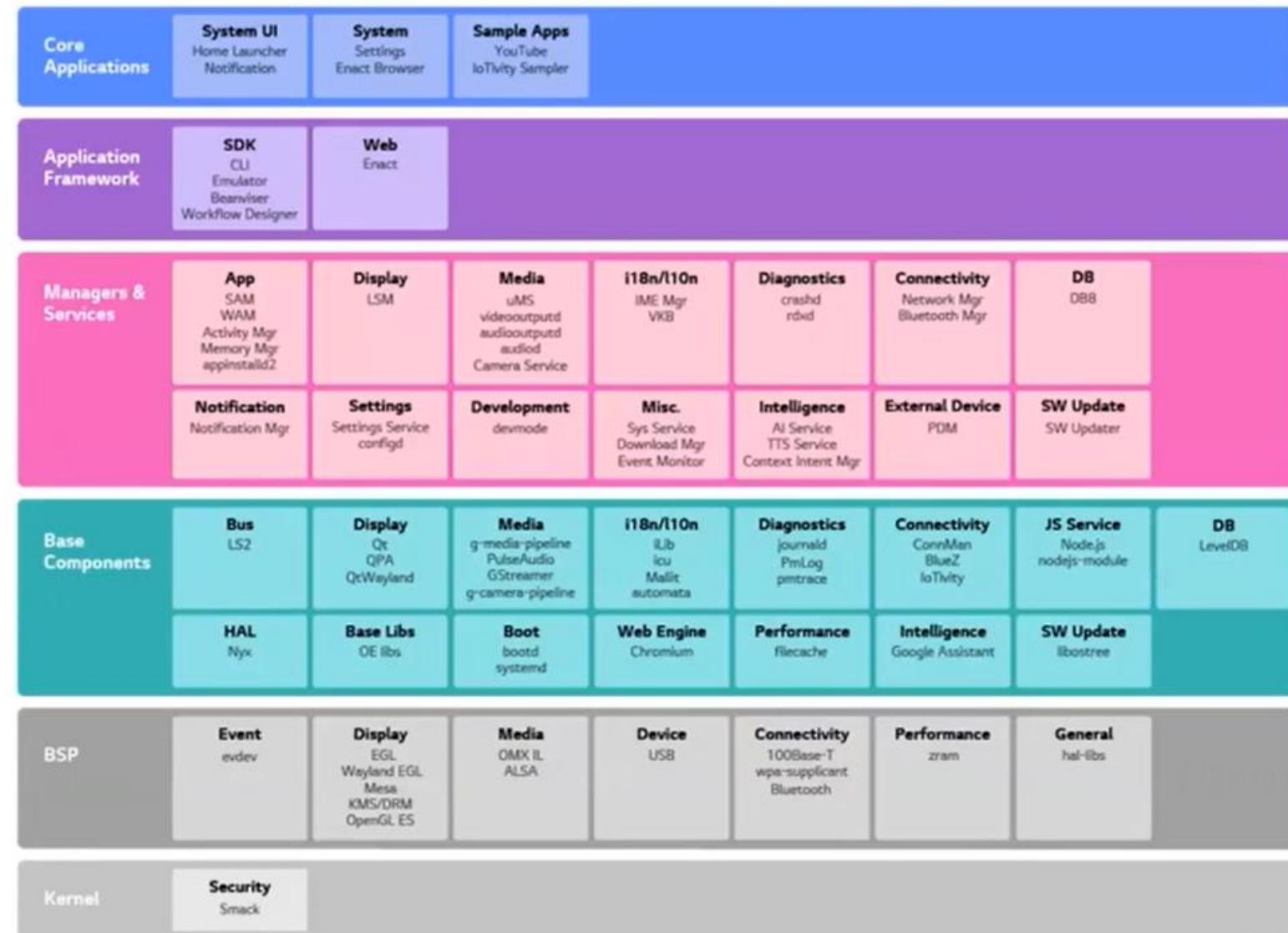
webOS OSE



A Broad Architecture View



Detailed Architecture View



Application Framework

- Enact Framework
 - Web app framework optimized for developing web apps for webOS OSE
 - Open source
- Software Development Kit (SDK)
 - Provides a development environment for web apps and services
 - Command line interface (**CLI**) - allows you to develop and test apps
 - Emulator - enables you to test the webOS application and platform on a virtual environment
 - Beanviser - performance monitoring and diagnostics tool
- Native Application Development (NDK)
 - Qt5 is part of WebOS
 - NDK has Qt and QML related development environment for Creating Native Apps

Managers and Services

- **System and Application Manager(SAM)** - manages each app throughout its lifecycle
- **Web Application Manager (WAM)** - launching and managing web apps
- **Activity Manager** - managing and executing activities requested from services
- **Luna Surface Manager (LSM)** - displays graphical elements on the screen and event handling for input devices such as keyboard and pointer
- **uMedia Server** - provides interfaces for media playback
- **DB8** - embedded JSON database that supports data storing and retrieving in the structure of a key-value store. Uses Level DB as the backend database

Base Components

- **LS2** - IPC mechanism used between components in webOS
- **Web Engine** - loads and parses a web app
- **Node.js** - JavaScript framework that is typically run on a server
- **Connectivity** – Provides Ethernet, Wi-Fi, Bluetooth features
- **Diagnostics** – Provides logging

Luna Bus

- What is Luna bus? – system bus for IPC
- Alternative to Luna bus in Linux – D-bus(IPC for Linux)
- Why Luna Bus (Luna bus Vs D-bus) ?
 - No of request are less.
- Security Available in Luna bus
 - LS2 clients ask for access to a specific group. If access is granted, LS2 clients can call APIs from the group.
- IPC vs RPC ?
 - IPC can have significant overhead for parsing and security.

Future Of WebOS

- WebOS Auto
- WebOS IOT



References

- Images Can be downloaded from
 - <http://build.webos-ports.org/webosose/>
- OSE Community - <https://forum.webosose.org/>
- WebOS Blog - <https://www.webosose.org/blog/>
- Social Links
 - https://twitter.com/webos_ose
 - <https://www.youtube.com/channel/UC8wy-KVwjZ9CPrVeq4DvmA>
- FAQs
 - <https://www.webosose.org/about/governance-licensing/faq/>
- Docs
 - <https://www.webosose.org/docs/home/>

Thank You

Dr. Daniel D

Email:

daniel.d@christuniversity.in

Mob: +91-9095003488

Q & As – CU students

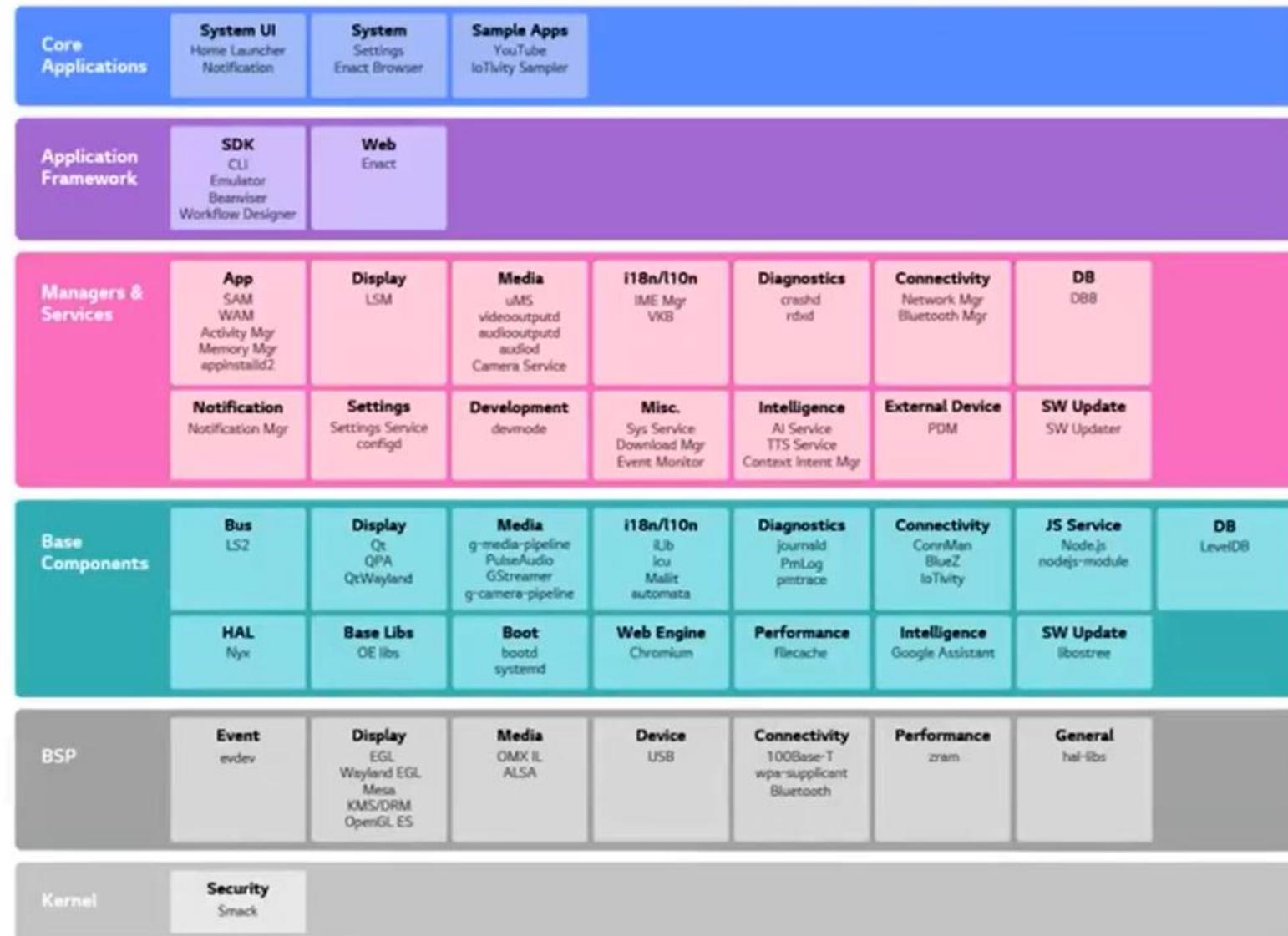
- Regular B.Tech Classes
- Placement Training and Placements
- Attendance
- CIAs (MSE)
- Raspberry Pi 4 Kit
- Emulator
- Ubuntu 18.04 LTS dual boot.

DAY2-1

webOS
Open Source Edition

Architecture

Detailed Architecture View



Next Topics

- Preparation of webOS development
- Explaining the build setup to create an image.

Sudo apt update

Sudo apt upgrade

Install Google chrome

Sudo apt install nodejs

To verify the installed version : "nodejs --version"

Sudo apt install npm

To verify the installed version : "npm --version"

sudo npm install -g @enact/cli

I

From app store install

Nodepad++

Visual studio code

Sudo apt update

Sudo apt upgrade

Install Google chrome

Sudo apt install nodejs

To verify the installed version : "nodejs --version"

Sudo apt install npm

To verify the installed version : "npm --version"

sudo npm install -g @enact/cli

From app store install

Notepad++

Visual studio code

I sudo apt install git|

Activities Terminal ▾ Tue 11:39
bm@bm-desktop:~



File Edit View Search Terminal Help

```
bm@bm-desktop:~$ git clone https://github.com/webosose/build-webos.git
Cloning into 'build-webos'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (110/110), done.
remote: Total 1067 (delta 54), reused 56 (delta 0), pack-reused 957
Receiving objects: 100% (1067/1067), 610.40 KiB | 32.00 KiB/s, done.
Resolving deltas: 100% (548/548), done.
bm@bm-desktop:~$
```

Activities Terminal ▾ Tue 11:39
bm@bm-desktop: ~/build-webos

File Edit View Search Terminal Help

```
bm@bm-desktop:~$ git clone https://github.com/webosose/build-webos.git
Cloning into 'build-webos'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (110/110), done.
remote: Total 1067 (delta 54), reused 56 (delta 0), pack-reused 957
Receiving objects: 100% (1067/1067), 610.40 KiB | 32.00 KiB/s, done.
Resolving deltas: 100% (548/548), done.
bm@bm-desktop:~$ cd build-webos
bm@bm-desktop:~/build-webos$
```

Google Chrome

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the foreground, displaying the output of a 'git clone' command for a 'build-webos' repository from GitHub. The command was run from the user's home directory (~). The terminal window has a title bar showing 'Activities Terminal' and the date/time 'Tue 11:39'. The window title also includes the command 'bm@bm-desktop: ~/build-webos'. The terminal content shows the progress of cloning the repository, including object enumeration, counting, compressing, receiving, and resolving deltas. Below the terminal, a docked application menu is visible, featuring icons for various applications like a browser, file manager, and system tools. A tooltip for 'Google Chrome' is visible near the bottom left of the screen.



File Edit View Search Terminal Help

```
bm@bm-desktop:~$ git clone https://github.com/webosose/build-webos.git
Cloning into 'build-webos'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (110/110), done.
remote: Total 1067 (delta 54), reused 56 (delta 0), pack-reused 957
Receiving objects: 100% (1067/1067), 610.40 KiB | 32.00 KiB/s, done.
Resolving deltas: 100% (548/548), done.
bm@bm-desktop:~$ cd build-webos
bm@bm-desktop:~/build-webos$ sudo scripts/prerequisites.sh
[sudo] password for bm:
Get:1 http://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Get:2 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu bionic InRelease [15.4 kB]
Get:3 http://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,147 B]
Get:4 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu bionic/main amd64 Packages [9,356 B]
Get:5 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu bionic/main i386 Packages [9,296 B]
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:7 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu bionic/main Translation-en [1,360 B]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [46.1 kB]
Hit:9 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:10 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [49.2 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:13 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1,031 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [723 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [346 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [295 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11 Metadata [279 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:20 http://in.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [9,288 B]
Fetched 3,074 kB in 37s (83.4 kB/s)
Reading package lists... Done
Hit:1 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu bionic InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:6 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists.. 23%
```



File Edit View Insert Format Styles Table Form Tools Window Help



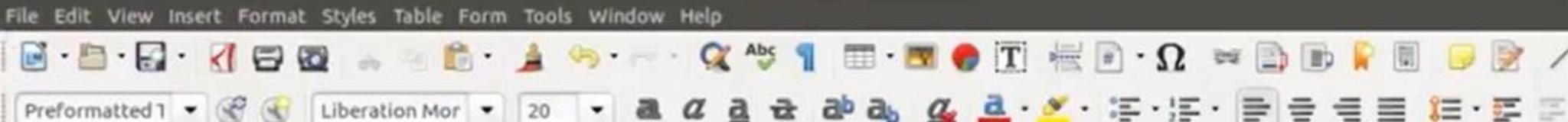
Preformatted 1

Liberation Mor

20



1. Sudo apt install git
2. git clone https://github.com/webosose/build-webos.git
3. cd build-webos
4. sudo scripts/prerequisites.sh
5. cat /proc/cpuinfo | grep "physical id" | sort | uniq | wc -l
6. cat /proc/cpuinfo | grep "cpu cores" | uniq
7. ./mcf -p <number of physical CPU cores / 2> -b <number of physical CPU cores / 2> <target-device-name>



2. git clone https://github.com/webosose/
build-webos.git

3. cd build-webos

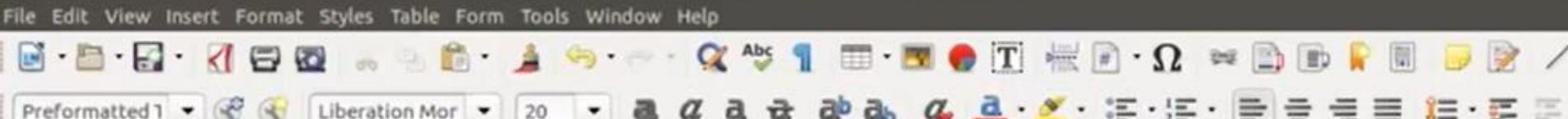
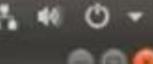
4. sudo scripts/prerequisites.sh

5. cat /proc/cpuinfo | grep "physical
id" | sort | uniq | wc -l

6. cat /proc/cpuinfo | grep "cpu cores"
| uniq

7. ./mcf -p <number of physical CPU
cores / 2> -b <number of physical CPU
cores / 2> <target-device-name>

./mcf -p 2 -b 2 raspberrypi4



2.git clone https://github.com/webosose/
build-webos.git

3.cd build-webos

4. sudo scripts/prerequisites.sh

5. cat /proc/cpuinfo | grep "physical
id" | sort | uniq | wc -l

6. cat /proc/cpuinfo | grep "cpu cores"
| uniq

7. ./mcf -p <number of physical CPU
cores / 2> -b <number of physical CPU
cores / 2> <target-device-name>

./mcf -p 2 -b 2 raspberrypi4
./mcf -p 2 -b 2 raspberrypi3
./mcf -p 2 -b 2 qemux86

DAY2-2

Meeting is in progress

```
File Edit View Search Terminal Help
bm@bm-desktop:~$ sudo apt update
[sudo] password for bm:
Hit:1 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu bionic InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [46.1 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [49.2 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 186 kB in 19s (9,810 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
bm@bm-desktop:~$ sudo
```



Meeting is in progress

```
bm@bm-desktop:~  
File Edit View Search Terminal Help  
Hit:4 http://dl.google.com/linux/chrome/deb stable InRelease  
Hit:5 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [46.1 kB]  
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [49.2 kB]  
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2,464 B]  
Fetched 186 kB in 19s (9,810 B/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
All packages are up to date.  
bm@bm-desktop:~$ sudo apt upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
The following packages were automatically installed and are no longer required:  
  eflbootmgr libegl1-mesa libfwup1 libllvm9 libwayland-egl1-mesa  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
bm@bm-desktop:~$
```



File Edit View Search Terminal Help

Fetched 186 kB in 19s (9,810 B/s)

Reading package lists... Done

Building dependency tree

Reading state information... Done

All packages are up to date.

bm@bm-desktop:~\$ sudo apt upgrade

Reading package lists... Done

Building dependency tree

Reading state information... Done

Calculating upgrade... Done

The following packages were automatically installed and are no longer required:

efibootmgr libegl1-mesa libfwupd1 libllvm9 libwayland-egl1-mesa

Use 'sudo apt autoremove' to remove them.

0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

bm@bm-desktop:~\$ sudo apt install nodejs

Reading package lists... Done

Building dependency tree

Reading state information... Done

nodejs is already the newest version (8.10.0~dfsg-2ubuntu0.4).

The following packages were automatically installed and are no longer required:

efibootmgr libegl1-mesa libfwupd1 libllvm9 libwayland-egl1-mesa

Use 'sudo apt autoremove' to remove them.

0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

bm@bm-desktop:~\$

Activities Terminal

Tue 15:04

Meeting is in progress Building webOS OSE | we

bm@bm-desktop:~/build-webos

File Edit View Search Terminal Help

```
bm@bm-desktop:~$ git clone https://github.com/webosose/build-webos.git
fatal: destination path 'build-webos' already exists and is not an empty directory.
bm@bm-desktop:~$ cd build-webos
bm@bm-desktop:~/build-webos$
```

Search

Contents

- Before You Begin
- Cloning the Repository
- Installing the Required Tools and Libraries
- Configuring the Build
 - Setting the Parallelism Values
 - Configuring the Build for the Target Device
- Building the Image
 - Building webos-image
 - Building webos-image-devel
- Checking the Created Image
- Cleaning
- Building and Cleaning Individual Components
- Next Steps

To build a webOS OSE image, `build-webos` repository is used. This repository contains the top level code that aggregates the various `OpenEmbedded` layers into a whole from which webOS OSE images can be built.

Set up `build-webos` by cloning its Git repository, and `cd` into the cloned directory:

```
$ git clone https://github.com/webosose/build-webos.git
$ cd build-webos
```

Installing the Required Tools and Libraries

Before you can build, some tools and libraries need to be

christkengeri.webex.com is sharing your screen. Stop sharing Hide

Activities Terminal

Tue 15:05

Meeting is in progress Building webOS OSE | we

File Edit View Search Terminal Help

```
bm@bm-desktop:~/build-webos$ git clone https://github.com/webosose/build-webos.git
fatal: destination path 'build-webos' already exists and is not an empty directory.
bm@bm-desktop:~/build-webos$ cd build-webos
bm@bm-desktop:~/build-webos$ sudo scripts/prerequisites.sh
[sudo] password for bm:
latest version of scripts/prerequisites.sh appears to have been successfully run
, use -f option to force script to run.
bm@bm-desktop:~/build-webos$
```

Search

Contents

- Before You Begin
- Cloning the Repository
- Installing the Required Tools and Libraries
- Configuring the Build
 - Setting the Parallelism Values
 - Configuring the Build for the Target Device
- Building the Image
 - Building webos-image
 - Building webos-image-devel
- Checking the Created Image
- Cleaning
 - Building and Cleaning Individual Components
- Next Steps

Google Assistant

- Setup
- Firmware-Over-the-Air Setup
- GDB Debugging Setup
- Native Development Kit Setup
- Development
- Web Apps
- JS Services

Installing the Required Tools and Libraries

Before you can build, some tools and libraries need to be installed. If you try to build without them, BitBake will fail a sanity check and tell you what's missing, but not really how to get the missing pieces. On Ubuntu, you can force all of the missing pieces to be installed by entering:

```
$ sudo scripts/prerequisites.sh
```

Note

For more details about BitBake, refer to the [BitBake](#)

christkengeri.webex.com is sharing your screen. Stop sharing Hide

Activities Terminal

Tue 15:05

Meeting is in progress Building webOS OSE | we +

File Edit View Search Terminal Help

```
bm@bm-desktop:~/build-webos
fatal: destination path 'build-webos' already exists and is not an empty directory.
bm@bm-desktop:~$ cd build-webos
bm@bm-desktop:~/build-webos$ sudo scripts/prerequisites.sh
[sudo] password for bm:
latest version of scripts/prerequisites.sh appears to have been successfully run
, use -f option to force script to run.
bm@bm-desktop:~/build-webos$ ./mcf -p 1 -b 1 qemux86
2020-07-28T15:05:33 INFO __main__ MCF-6.2.5: Updating build directory
2020-07-28T15:05:33 INFO __main__ Sanity checking [bitbake]
2020-07-28T15:05:33 INFO __main__ Checking for local changes in [bitbake]
2020-07-28T15:05:36 INFO __main__ [bitbake] is up-to-date.
2020-07-28T15:05:36 INFO __main__ Sanity checking [oe-core]
2020-07-28T15:05:37 INFO __main__ Checking for local changes in [oe-core]
```

Search

Contents

- Before You Begin
- Cloning the Repository
- Installing the Required Tools and Libraries
- Configuring the Build
- Setting the Parallelism Values
- Configuring the Build for the Target Device
- Building the Image
- Building webos-image
- Building webos-image-devel
- Checking the Created Image
- Cleaning
- Building and Cleaning Individual Components
- Next Steps

Google Assistant

Setup

Firmware-Over-the-Air Setup

GDB Debugging Setup

Native Development Kit Setup

Development

Web Apps

JS Services

Installing the Required Tools and Libraries

Before you can build, some tools and libraries need to be installed. If you try to build without them, BitBake will fail a sanity check and tell you what's missing, but not really how to get the missing pieces. On Ubuntu, you can force all of the missing pieces to be installed by entering:

```
$ sudo scripts/prerequisites.sh
```

Note

For more details about BitBake, refer to the [BitBake](#)

christkengeri.webex.com is sharing your screen. Stop sharing Hide

Activities Terminal Tue 15:06

Meeting is in progress Building webOS OSE | we... +

File Edit View Search Terminal Help

2020-07-28T15:06:01 INFO __main__ Checking for local changes in [meta-webosose]
2020-07-28T15:06:02 INFO __main__ Updating [meta-webosose]
2020-07-28T15:06:03 INFO __main__ Done updating [meta-webosose]
2020-07-28T15:06:03 INFO __main__ Sanity checking [meta-raspberrypi]
2020-07-28T15:06:03 INFO __main__ Checking for local changes in [meta-raspberrypi]
2020-07-28T15:06:03 INFO __main__ [meta-raspberrypi] is up-to-date.
2020-07-28T15:06:03 INFO __main__ Sanity checking [meta-security]
2020-07-28T15:06:03 INFO __main__ Checking for local changes in [meta-security]
2020-07-28T15:06:04 INFO __main__ [meta-security] is up-to-date.
2020-07-28T15:06:04 INFO __main__ Repo Update Summary
2020-07-28T15:06:04 INFO __main__ =====
2020-07-28T15:06:04 INFO __main__ No local changes found
2020-07-28T15:06:04 WARNING __main__ WARNING: ICECC package installation check failed, disabling build from using ICECC.
2020-07-28T15:06:04 INFO __main__ MCF-6.2.5: Configuring build directory BUILD
2020-07-28T15:06:06 INFO __main__ MCF-6.2.5: Done configuring build directory BU
ILD
2020-07-28T15:06:06 WARNING __main__ No mirror for github.com was detected, please define mirrors in ~/.gitconfig if some are available
2020-07-28T15:06:06 INFO __main__ Done.
bm@bm-desktop:~/build-webos\$ source oe-init-build-env
Filtered environment for qemux86@webos development
bm@bm-desktop:~/build-webos\$

Setup tools added to webos-image , including GDB and strace (system call tracer).

Building webos-image

To kick off a full build of webOS OSE, enter the following:

```
$ source oe-init-build-env  
$ bitbake webos-image
```

Alternatively, you can enter:

```
$ make webos-image
```

This may take in the neighbourhood of two hours on a multi-core system.

christkengeri.webex.com is sharing your screen. Stop sharing Hide

Contents

- Before You Begin
- Cloning the Repository
- Installing the Required Tools and Libraries
- Configuring the Build
 - Setting the Parallelism Values
- Configuring the Build for the Target Device
- Building the Image
 - Building webos-image
 - Building webos-image-devel
- Checking the Created Image
- Cleaning
- Building and Cleaning Individual Components
- Next Steps

Activities Terminal

Tue 15:07

Meeting is in progress Building webOS OSE | we... +

bm@bm-desktop: ~/build-webos

File Edit View Search Terminal Help

```
[1] 2020-07-28T15:06:03 INFO __main__ [meta-raspberrypi] is up-to-date.
2020-07-28T15:06:03 INFO __main__ Sanity checking [meta-security]
2020-07-28T15:06:03 INFO __main__ Checking for local changes in [meta-security]
2020-07-28T15:06:04 INFO __main__ [meta-security] is up-to-date.
2020-07-28T15:06:04 INFO __main__ Repo Update Summary
2020-07-28T15:06:04 INFO __main__ =====
2020-07-28T15:06:04 INFO __main__ No local changes found
2020-07-28T15:06:04 WARNING __main__ WARNING: ICECC package installation check failed, disabling build from using ICECC.
2020-07-28T15:06:04 INFO __main__ MCF-6.2.5: Configuring build directory BUILD
2020-07-28T15:06:06 INFO __main__ MCF-6.2.5: Done configuring build directory BUILD
2020-07-28T15:06:06 WARNING __main__ No mirror for github.com was detected, please define mirrors in ~/.gitconfig if some are available
2020-07-28T15:06:06 INFO __main__ Done.
bm@bm-desktop:~/build-webos$ source oe-init-build-env
Altered environment for qemux86@webos development
bm@bm-desktop:~/build-webos$ make webos-image
for MACHINE in qemux86 ; do . ./oe-init-build-env && bitbake webos-image; done
Altered environment for qemux86@webos development
Loading cache: 100% [########################################] Time: 0:00:07
Loaded 3639 entries from dependency cache.
```

without development trace

Google Assistant

- Setup
- Firmware-Over-the-Air Setup
- GDB Debugging Setup
- Native Development Kit Setup
- Development ▾
- Web Apps ▾
- JS Services ▾

Building webos-image ↗

To kick off a full build of webOS OSE, enter the following:

```
$ source oe-init-build-env
$ bitbake webos-image
```

Alternatively, you can enter:

```
$ make webos-image
```

This may take in the neighborhood of two hours on a multi-core workstation with a fast disk subsystem and lots of memory, or many more hours on a laptop with less memory and slower

christkengeri.webex.com is sharing your screen. Stop sharing Hide

Contents

- Before You Begin
- Cloning the Repository
- Installing the Required Tools and Libraries
- Configuring the Build
 - Setting the Parallelism Values
 - Configuring the Build for the Target Device
- Building the Image
 - Building webos-image
 - Building webos-image-devel
 - Checking the Created Image
- Cleaning
 - Building and Cleaning Individual Components
- Next Steps

DAY3-1



YOCTO PROJECT

It's not an embedded Linux
Distribution,
It creates a custom one for you.

Agenda

- ✓ Yocto project overview
- ✓ Features & Advantages
- ✓ Challenges
- ✓ Yocto Project Layer Model
- ✓ Components & Tool in Yocto Project
- ✓ Poky – Reference Embedded Distribution
- ✓ OE – OpenEmbedded build system
- ✓ Build-weboS
- ✓ WeboSose Build Flow

Overview of Yocto Project

- What is a Yocto project?
 - The Yocto Project is an open source collaboration project that helps developers create custom Linux-based systems that are designed for embedded products regardless of the product's hardware architecture.
- What Yocto provides for the developers?
 - shared technologies, software stacks, configurations, and best practices used to create these tailored Linux images.

[YoctoProject](#)

Features & Advantages

- ***Widely Adopted Across the Industry***
- ***Architecture Agnostic*** - Yocto Project supports Intel, ARM, MIPS, AMD, PPC and other architectures(custom made BSPs & QEMUX)
- ***Images and Code Transfer Easily***
- ***Flexibility*** - create an internal Linux distribution, It can be used across multiple product groups, through customization and layering.

- *Ideal for Constrained Embedded and IoT devices.*
- *Comprehensive Toolchain Capabilities.*
- *Mechanism Rules Over Policy*
- *Uses a Layer Model*
- *Supports Partial Builds* - shared-state cache (sstate) scheme.
- *Rich Ecosystem of Individuals and Organizations*
- *License Manifest*

Challenges

- *Steep Learning Curve*
- *Understanding What Changes You Need to Make For Your Design Requires Some Research*
- *Project Workflow Could Be Confusing*
- *Working in a Cross-Build Environment Can Feel Unfamiliar*
- *Initial Build Times Can be Significant*

Yocto Project Layer Model

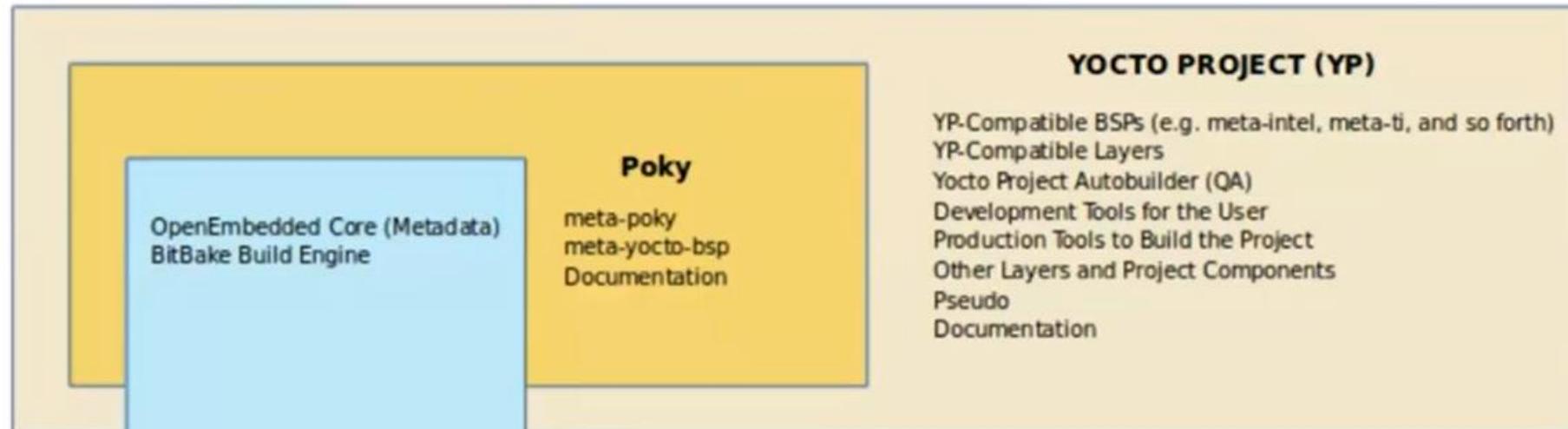
- "Layer Model" for embedded and IoT Linux creation
- Layers are repositories that contain related sets of instructions that tell the **Open Embedded build system** what to do
- supports **collaboration** and **customization**
- Isolating information into layers.

Components and Tools

- Development Tools
- Production Tools
- Open-Embedded Build System Components
- Reference Distribution (Poky)
- Packages for Finished Targets (*Opkg* Open PackaGe management (opkg) – Package Management tool)
- Archived Components (Hobs, Build Appliance, CROPS)

Development Methods

- *Native Linux Host – uses bb (Native Linux)*
- *CROss PlatformS (CROPS) – Non Native to Linux*
- *Toaster – Web based interface*
- *Eclipse™ IDE*

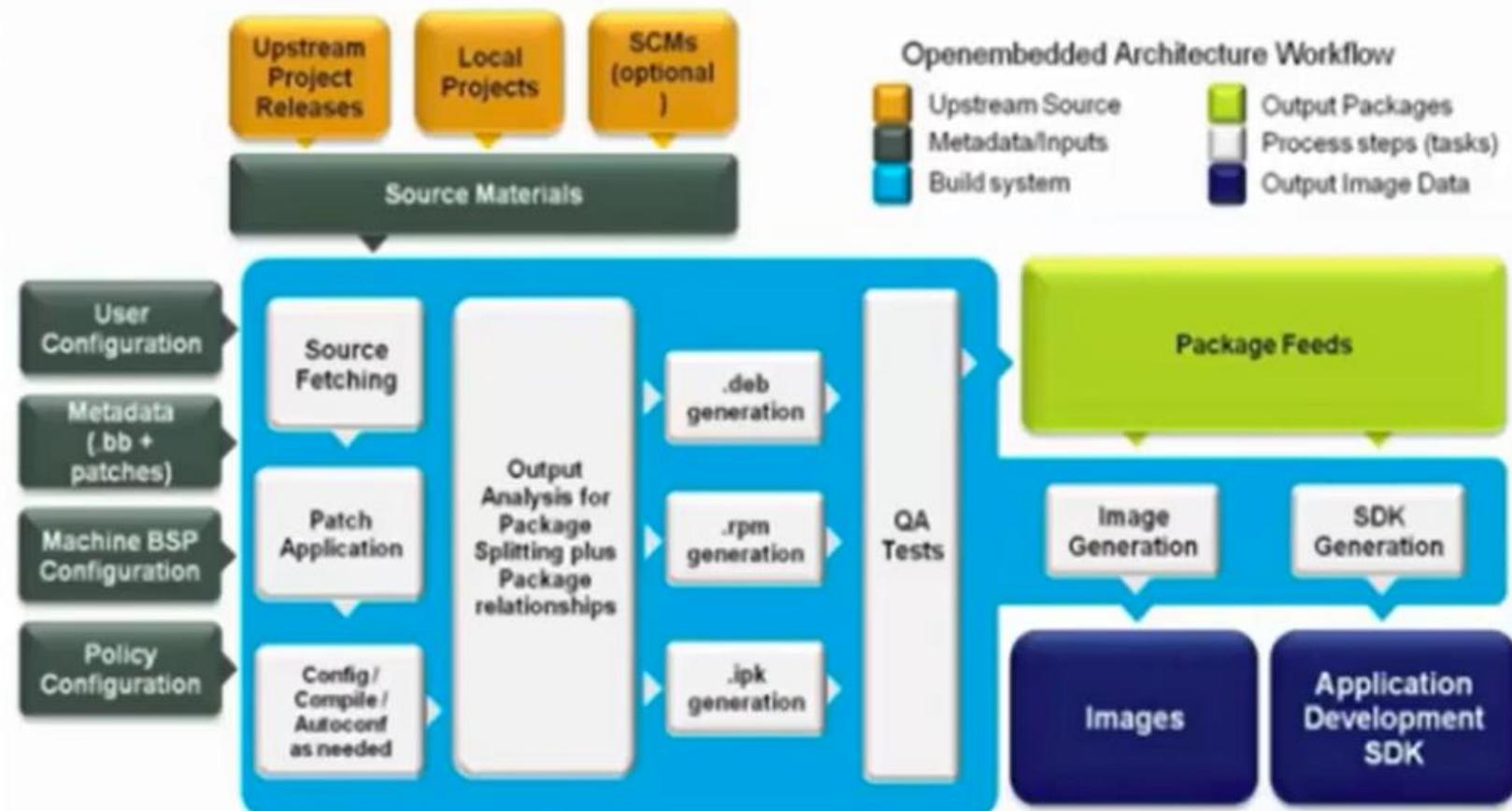


meta-openembedded

OpenEmbedded

- BitBake is a task executor and scheduler that is the heart of the OpenEmbedded build system.
- meta-poky, which is Poky-specific metadata
- meta-yocto-bsp, which are Yocto Project-specific Board Support Packages (BSPs).
- OpenEmbedded-Core (OE-Core) metadata, which includes shared configurations, global variable definitions, shared classes, packaging, and recipes.
- Recipes are the logical units of software and images to be built.

OpenEmbedded Build System Workflow



Contd..

- Developers specify architecture, policies, patches and configuration details.
- The build system fetches and downloads the source code from the specified location. The build system supports standard methods such as tarballs or source code repositories systems such as Git.
- Once source code is downloaded, the build system extracts the sources into a local work area where patches are applied and common steps for configuring and compiling the software are run.
- The build system then installs the software into a temporary staging area where the binary package format you select (DEB, RPM, or IPK) is used to roll up the software.
- Different QA and sanity checks run throughout entire build process.
- After the binaries are created, the build system generates a binary package feed that is used to create the final root file image.
- The build system generates the file system image and a customized Extensible SDK (eSDSK) for application development in parallel.

Terms to Remember

- Poky: a reference distribution of the Yocto Project and reference test configuration

Poky = metadata + bitbake

- Bitbake: a task execution engine
- Metadata : task definitions
 - recipes (.bb): Logical units of software to build
 - classes (.bbclass): common logic for build & packaging
 - configuration (.conf): Global definition variables
- OE-Core is metadata comprised of foundation recipes, classes, and associated files

build-webos

- Cloning the source :

```
$git clone https://github.com/webosose/build-webos.git
```

```
$ cd build-webos
```

- Install the required dependencies :

```
$ sudo scripts/prerequisites.sh
```

- Clone metalayers & configure :

```
$ ./mcf -b <bb_number_threads> -p <parallel_make_threads>  
<MACHINE>
```

- Building webos-image

```
$source oe-init-build-env
```

```
$ bitbake webos-image
```

WeboSose Build Flow

./mcf:

- Clone metalayers mentioned in weboslayers.py & Create configuration using build-templates.

bitbake webos-image :

- Flow of build command Goes as below

Webos-image.bb - inherit webos-image.bbclass

|

webos-image.bbclass - IMAGE_INSTALL = "packagegroup-* .bb"

|

Packagegroup-* .bb

|

RDEPENDS = "\\"

<recipe_name1> \

<recipe_name2> \ .

"

Activities Firefox Web Browser

Wed 10:37

Building webOS OSE | Building webOS OSE | Open Source Edition - Mozilla Firefox

Inbox - praveen.kulkarni | Meeting is in progress | Building webOS OSE | +

https://www.webosose.org/docs/guides/setup/building-webos-ose/

webOS Open Source Edition

About Docs Blog Community Search

Guides

Getting Started

- Hello, webOS OSE!
- Introduction to LS2 API

Setup

- System Requirements
- Building webOS OSE**
- Flashing webOS OSE
- Network Setup
- Dual Display Setup
- Google Assistant Setup
- Firmware-Over-the-Air Setup
- GDB Debugging Setup
- Native Development Kit Setup

Development

- Web Apps
- JS Services
- QML Apps
- Native Apps
- Native Services
- Configuration Files

\$ bitbake webos-image-devel

Note

- For details on setting up the environment to debug webOS OSE with GDB, see [GDB Debugging Setup](#).
- For more information on how to use strace, refer to [the article on strace](#).

Checking the Created Image

To see if the image has been created successfully, check the following files:

- For Raspberry Pi 4, the resulting image will be created at `BUILD/deploy/images/raspberrypi4/webos-image-raspberrypi4.rootfs.wic`.
- For Raspberry Pi 3, the resulting image will be created at `BUILD/deploy/images/raspberrypi3/webos-image-raspberrypi3.rootfs.rpi-sdimg`.
- For the emulator, the resulting image will be created at `BUILD/deploy/images/qemuX86/webos-image-qemuX86-master-*-.wic.vmdk`.

Once you checked the image file, move on to the [Next Steps](#).

Cleaning

To blow away the build artifacts and prepare to do the clean build, you can remove the build directory and recreate it by typing:

```
$ rm -rf BUILD
$ ./mcf.status
```

What this retains are the caches of the downloaded source (under `./downloads`) and shared state (under `./sstate-cache`). These caches will save you a tremendous amount of time during development as they facilitate incremental builds, but can cause seemingly inexplicable behavior when corrupted. If you experience strangeness, use the command presented below to remove the shared state of suspicious components. In extreme cases, you

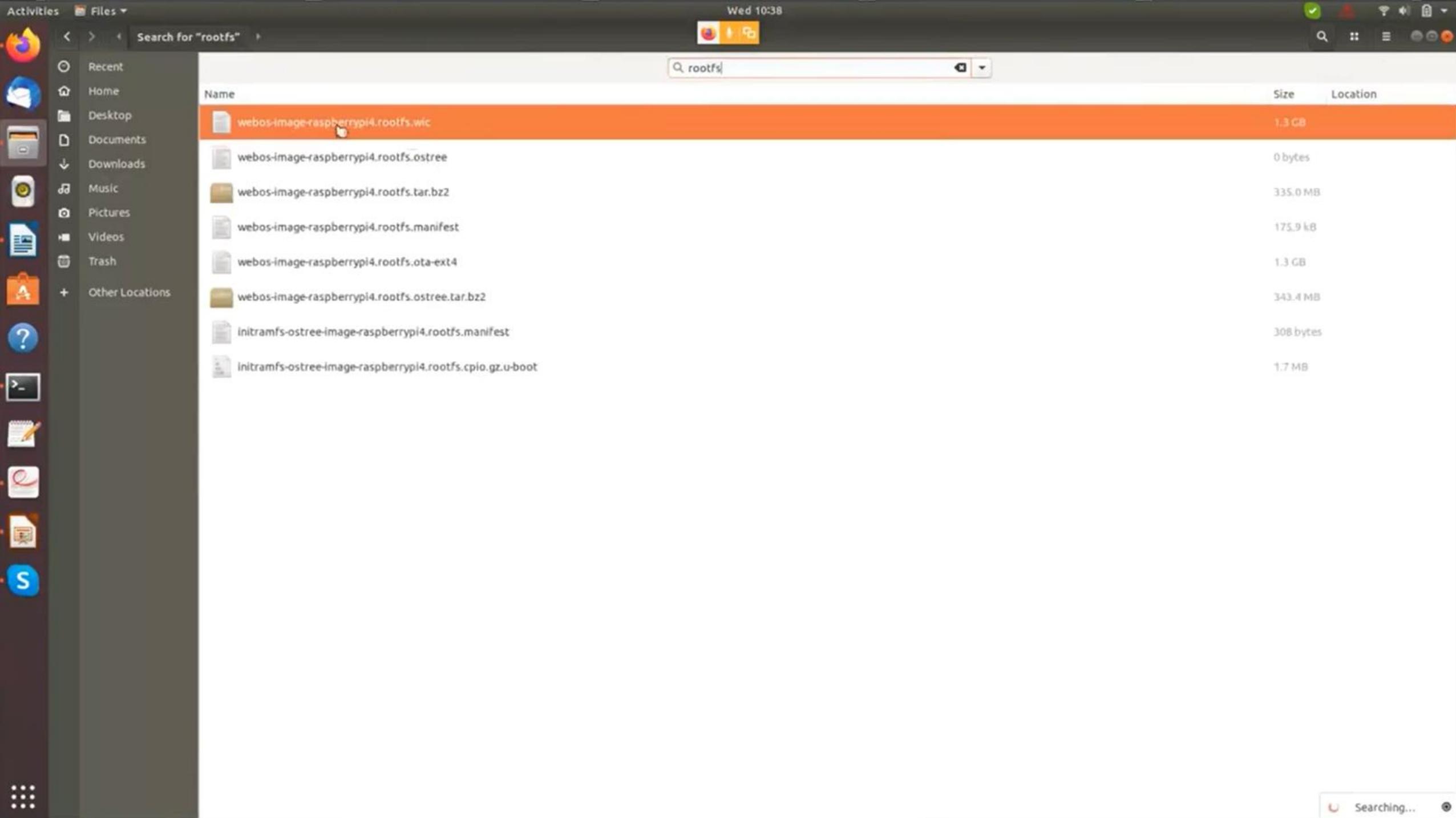
Contents

- Before You Begin
- Cloning the Repository
- Installing the Required Tools and Libraries
- Configuring the Build
 - Setting the Parallelism Values
 - Configuring the Build for the Target Device
- Building the Image
 - Building webos-image
 - Building webos-image-devel**
 - Checking the Created Image
- Cleaning
 - Building and Cleaning Individual Components
- Next Steps

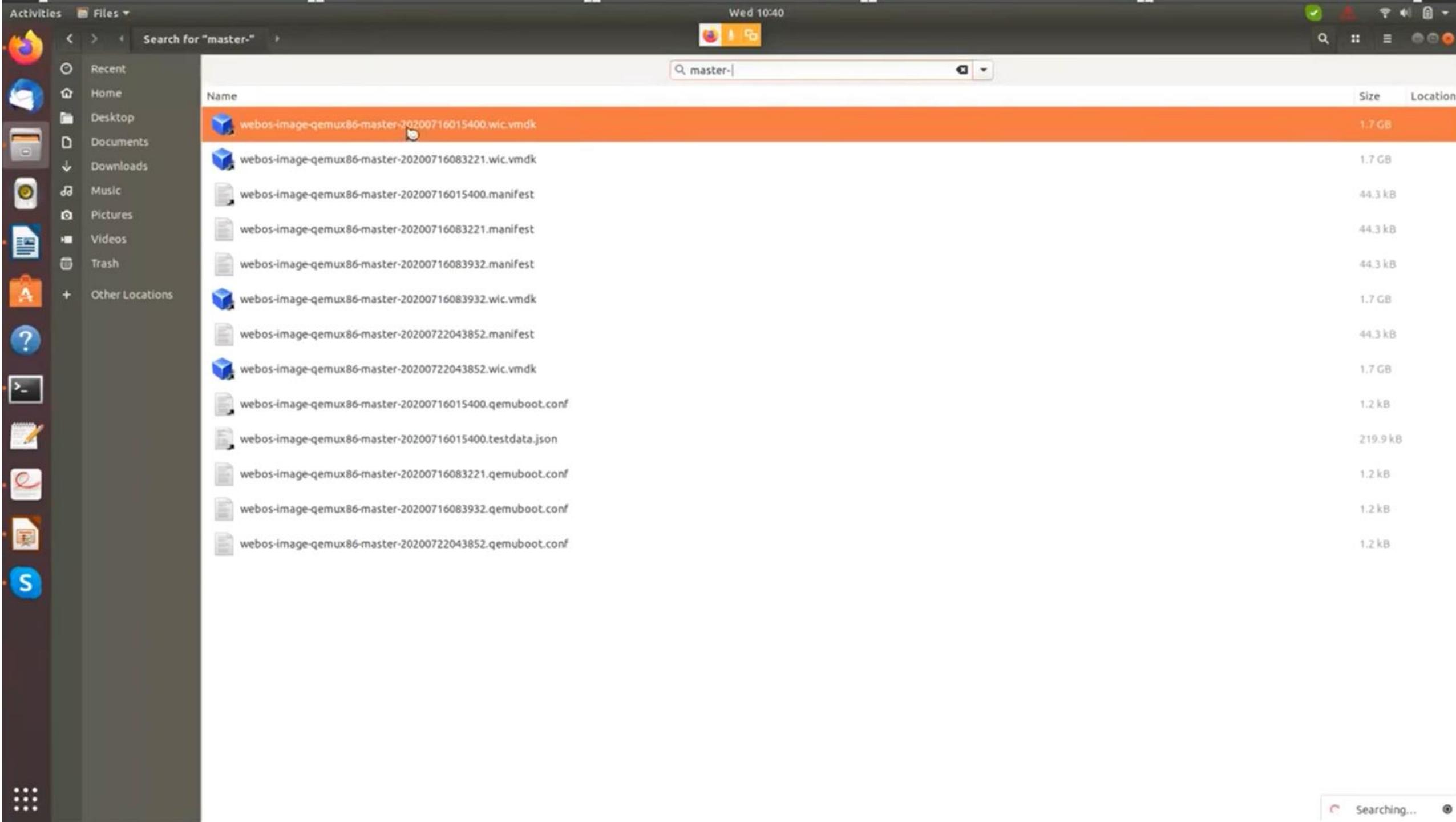


-  Recent
 -  Home
 -  Desktop
 -  Documents
 -  Downloads
 -  Music
 -  Pictures
 -  Videos
 -  Trash
 -  + Other Locations

bcm2835-bootfiles	ostree_repo	at86rf233.dtb	at86rf233-raspberrypi4.dtb	at86rf233-raspberrypi4-master...	bcm2708-rpi-b.dtb	bcm2708-rpi-b-plus.dtb	bcm2708-rpi-b-plus-raspberr...	bcm2708-rpi-b-plus-raspberr...	bcm2708-rpi-b-raspberr...	bcm2708-rpi-b-raspberr...	bcm2708-rpi-b-raspberr...	bcm2708-rpi-b-raspberr...	bcm2708-rpi-b-raspberr...	bcm2708-rpi-b-raspberr...	bcm2708-rpi-b-raspberr...	
bcm2709-rpi-2-b.dtb	bcm2709-rpi-2-b-raspberr...	bcm2709-rpi-2-b-raspberr...	bcm2710-rpi-3-b.dtb	bcm2710-rpi-3-b-plus.dtb	bcm2710-rpi-3-b-plus-rasp...	bcm2710-rpi-3-b-plus-rasp...	bcm2710-rpi-3-b-raspberr...	bcm2710-rpi-3-b-raspberr...	bcm2710-rpi-cm3.dtb	bcm2710-rpi-cm3-raspberr...	bcm2710-rpi-cm3-raspberr...	bcm2711-rpi-4-b.dtb	bcm2711-rpi-4-b-raspberr...	bcm2711-rpi-4-b-raspberr...	boot.scr	dwc2.dtb
dwc2-raspberrypi4.dtb	dwc2-raspberrypi4-master...	dwc2-raspberrypi4.dtb	gpio-key-raspberrypi4.dtb	gpio-key-raspberrypi4-master...	hifiberry-amp.dtb	hifiberry-amp-raspberr...	hifiberry-amp-raspberr...	hifiberry-dac.dtb	hifiberry-dacplus.dtb	hifiberry-dacplus-raspberr...	hifiberry-dacplus-raspberr...	hifiberry-dac-raspberr...	hifiberry-digidi...	hifiberry-digidi...	hifiberry-digidi...	hifiberry-digidi...
i2c-rtc.dtb	i2c-rtc-raspberrypi4.dtb	i2c-rtc-raspberrypi4.dtb	initramfs-ostree-image-ra...	initramfs-ostree-image-ra...	initramfs-ostree-image-ra...	initramfs-ostree-image-ra...	initramfs-ostree-image-ra...	initramfs-ostree-image-ra...	lqaudio-dac.dtb	lqaudio-dacplus.dtb	lqaudio-dacplus-raspberr...	lqaudio-dacplus-raspberr...	lqaudio-dac-raspberr...	lqaudio-dac-raspberr...	lqaudio-dac-raspberr...	mcp2515-can0.dtb
mcp2515-can0-raspberr...	mcp2515-can0-raspberr...	pi3-disable-bt.dtb	pi3-disable-bt-raspberr...	pi3-disable-bt-raspberr...	pi3-minuart-bt.dtb	pi3-minuart-bt-raspberr...	pi3-minuart-bt-raspberr...	pitft22.dtb	pitft22-raspberrypi4.dtb	pitft22-raspberrypi4-master...	pitft28-resistive.dtb	pitft28-resistive-raspberr...	pitft28-resistive-raspberr...	pitft35-resistive.dtb	pitft35-resistive-raspberr...	pitft35-resistive-raspberr...
pps-gpio.dtb	pps-gpio-raspberrypi4.dtb	pps-gpio-raspberrypi4.dtb	rpi-ft5406.dtb	rpi-ft5406-raspberrypi4.dtb	rpi-poe.dtb	rpi-poe-raspberrypi4.dtb	rpi-poe-raspberrypi4.dtb	u-boot	u-boot-raspberrypi4.bin	u-boot-raspberrypi4-2019.01...	ulimage	ulimage-raspberrypi4.bin	ulimage-raspberrypi4-master...	vc4-fkms-v3d.dtb	vc4-fkms-v3d.raspberr...	vc4-fkms-v3d.raspberr...
vc4-fkms-v3d.raspberr...	vc4-kms-v3d.dtb	vc4-kms-v3d.raspberr...	vc4-kms-v3d.raspberr...	w1-gpio.dtb	w1-gpio-pullup.dtb	w1-gpio-pullup-raspberr...	w1-gpio-pullup-raspberr...	w1-gpio-raspberrypi4.dtb	w1-gpio-raspberrypi4.dtb	webos-image-raspberr...	webos-image-raspberr...	webos-image-raspberr...	webos-image-raspberr...	webos-image-raspberr...	webos-image-raspberr...	webos-image-raspberr...
webos-image-raspberrypi4-master-20200712104328.manifest	webos-image-raspberrypi4-master-20200712104328.manifest	webos-image-raspberrypi4-master-20200713145524.ostree	webos-image-raspberrypi4-master-20200713145524.ostree	webos-image-raspberrypi4-master-20200713145524.ota-ext4	webos-image-raspberrypi4-master-20200713145524.tar.bz2	webos-image-raspberrypi4-master-20200713145524.tar.bz2	webos-image-raspberrypi4-master-20200713145524.wic	webos-image-raspberrypi4-master-20200713153942.manifest	webos-image-raspberrypi4-master-20200713153942.tar.bz2	webos-image-raspberrypi4-master-20200713154229.manifest	webos-image-raspberrypi4-master-20200713154229.tar.bz2	webos-image-raspberrypi4-master-20200713154229.wic	webos-image-raspberrypi4-master-20200713154229.wic	webos-image-raspberrypi4-master-20200713154229.wic	webos-image-raspberrypi4-master-20200713154229.wic	webos-image-raspberrypi4-master-20200713154229.wic



- Recent
- Home
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Trash
- + Other Locations



Inbox - praveen.kulkarni X Meeting is in progress X Building webOS OSE | Source Edition - Mozilla Firefox

https://www.webosose.org/docs/guides/setup/building-webos-ose/

webOS

Open Source Edition

About Docs Blog Community Search

Guides

- Getting Started
- Hello, webOS OSE!
- Introduction to LS2 API
- Setup
- System Requirements
- Building webOS OSE**
- Flashing webOS OSE
- Network Setup
- Dual Display Setup
- Google Assistant Setup
- Firmware-Over-the-Air Setup
- GDB Debugging Setup
- Native Development Kit Setup
- Development
- Web Apps
- JS Services
- QML Apps
- Native Apps
- Native Services
- Configuration Files

To build an individual component, enter:

```
$ source oe-init-build-env  
$ bitbake <component-name>
```

Alternatively, you can enter:

```
$ make <component-name>
```

To clean a component's build artifacts under `BUILD`, enter:

```
$ source oe-init-build-env  
$ bitbake -c clean <component-name>
```

To remove the shared state for a component as well as its build artifacts to ensure it gets rebuilt afresh from its source, enter:

```
$ source oe-init-build-env  
$ bitbake -c cleansstate <component-name>
```

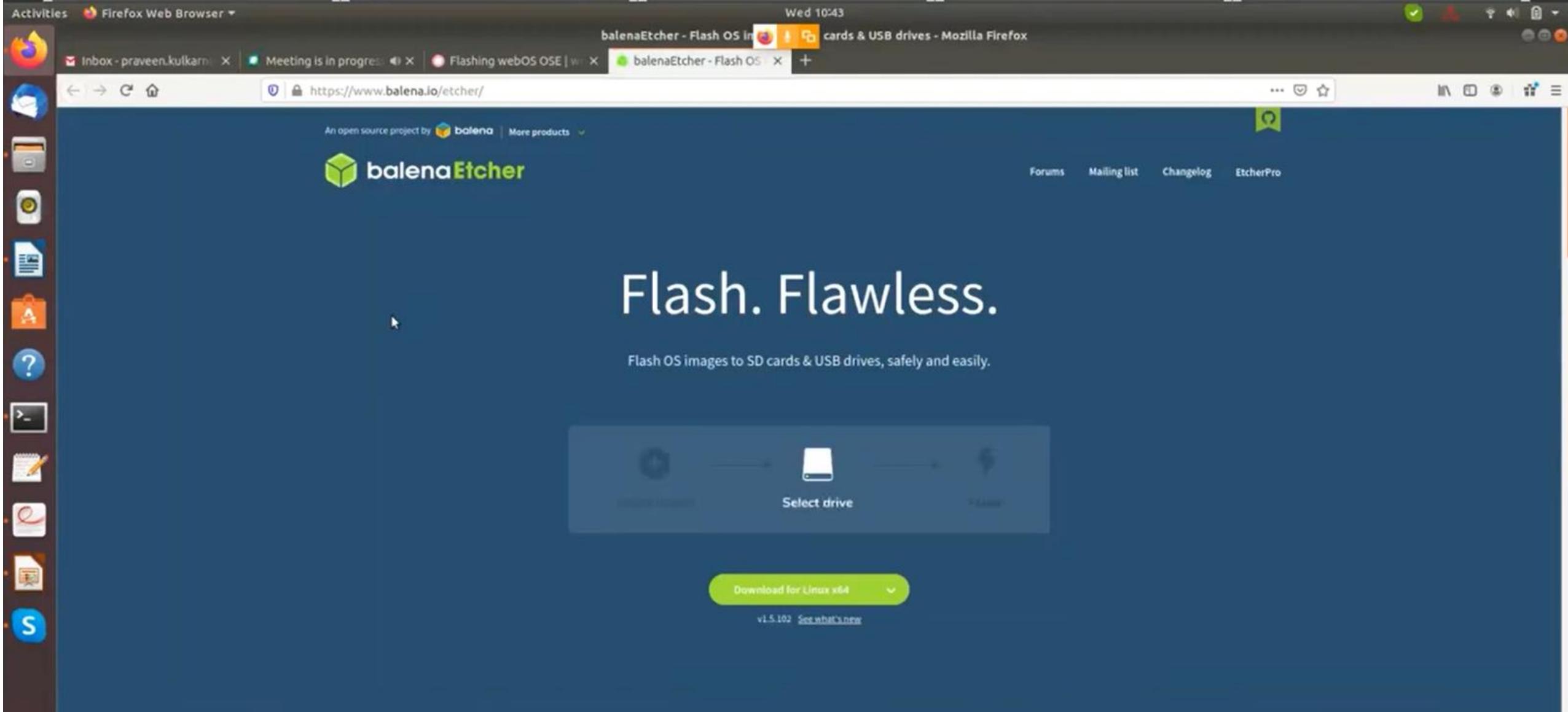
Next Steps

- If you built the image for Raspberry Pi 4 or Raspberry Pi 3, it's time to flash the image to the target device. See [Flashing webOS OSE](#).
- If you built the image for the emulator, refer to the Emulator User Guide to set up and use the emulator.

Contents
Before You Begin
Cloning the Repository
Installing the Required Tools and Libraries
Configuring the Build
Setting the Parallelism Values
Configuring the Build for the Target Device
Building the Image
Building webos-image
Building webos-image-devel
Checking the Created Image
Cleaning
Building and Cleaning Individual Components
Next Steps

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 and sample code is licensed under the Apache License 2.0.

https://www.webosose.org/docs/guides/setup/flashing-webos-ose/



Validated Flashing

No more writing images on corrupted
cards and wondering why your device

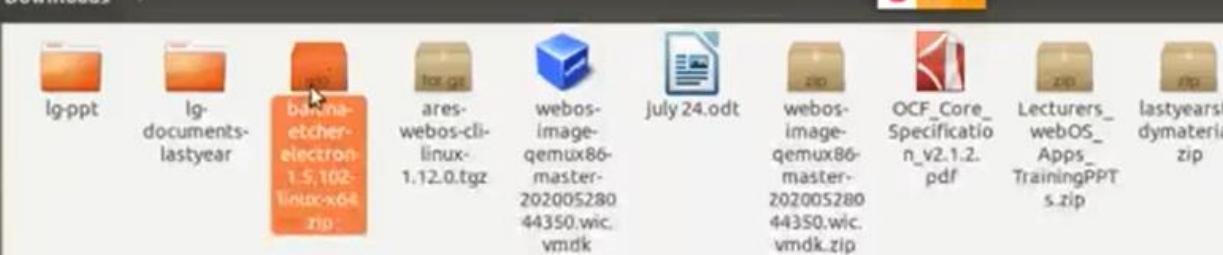
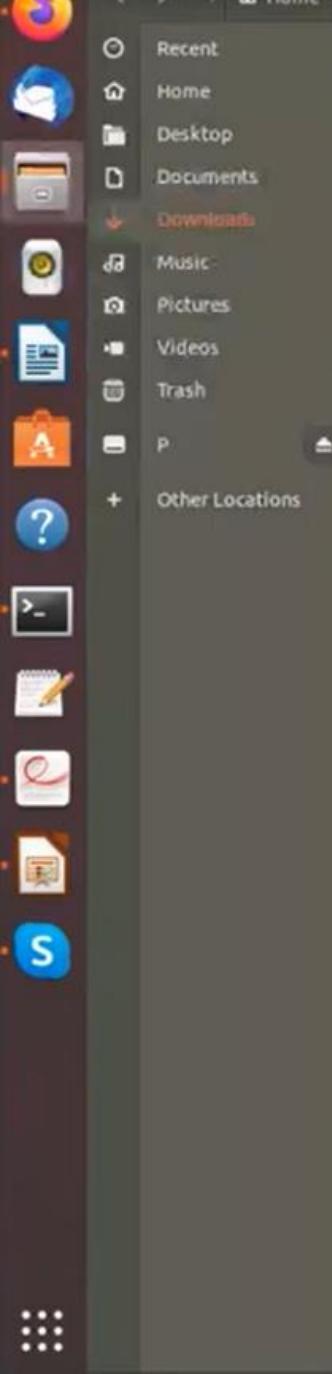


Hard Drive Friendly

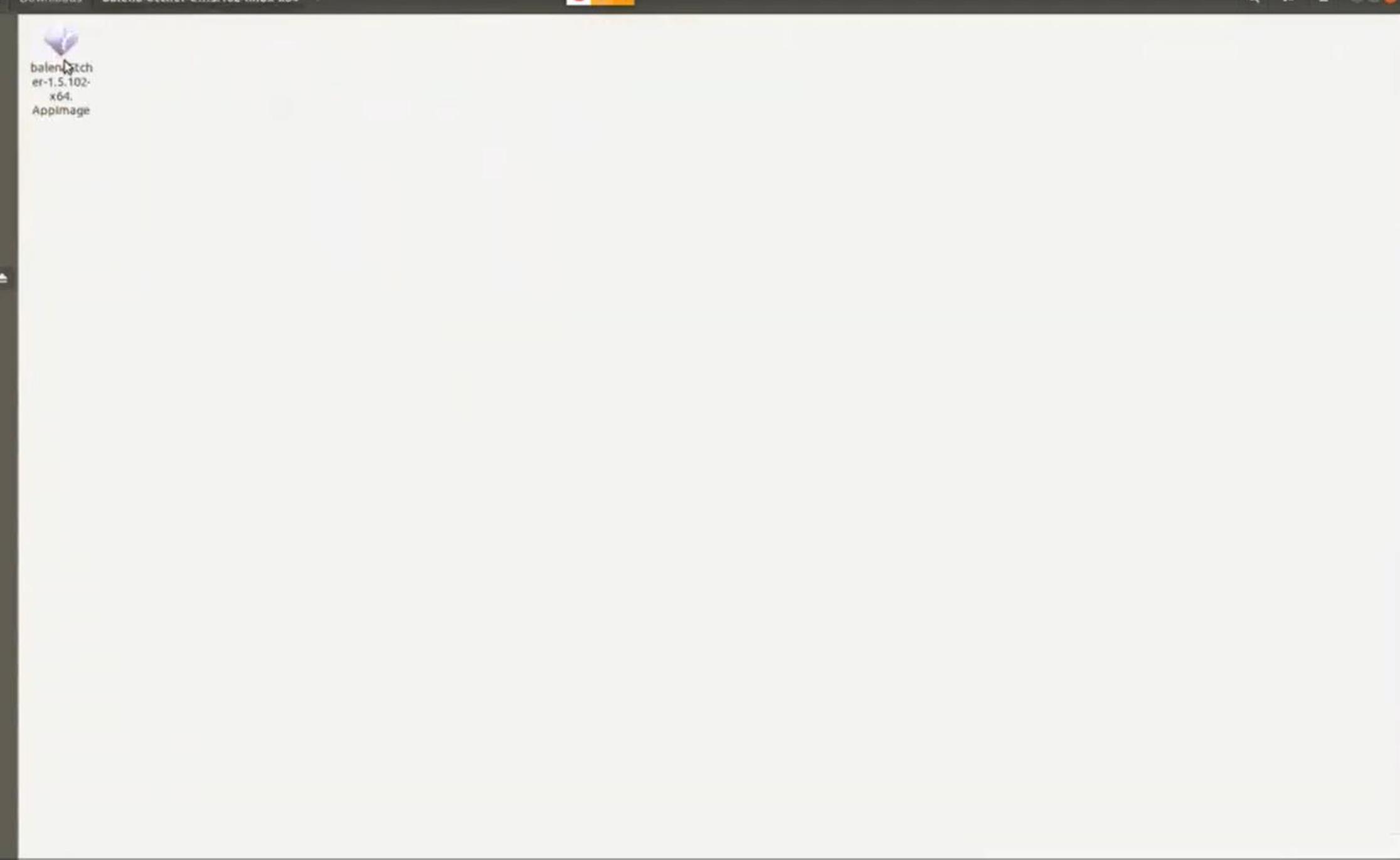
Makes drive selection obvious to avoid
wiping your entire hard-drive

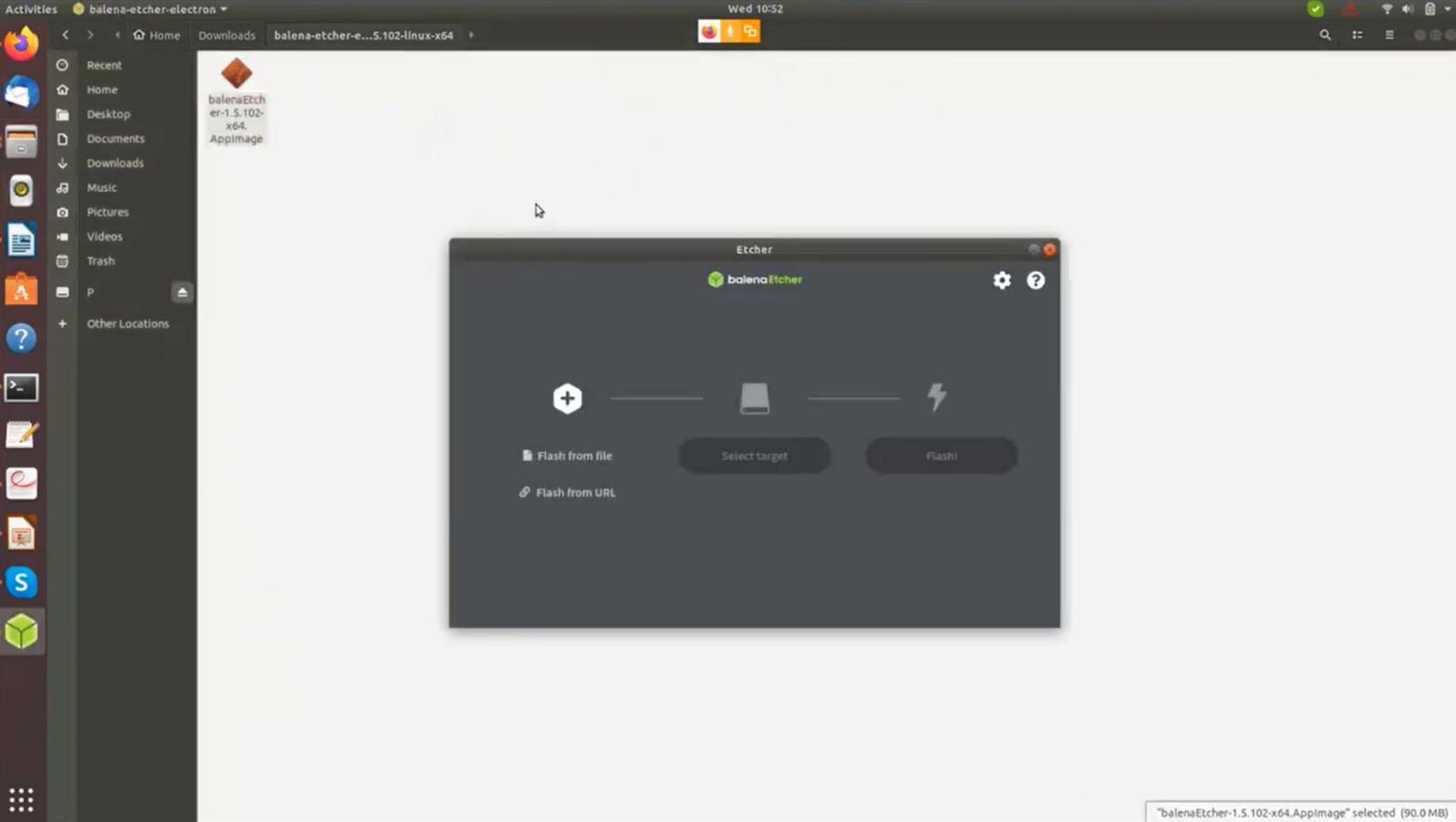
FEATURES

A better way to burn.

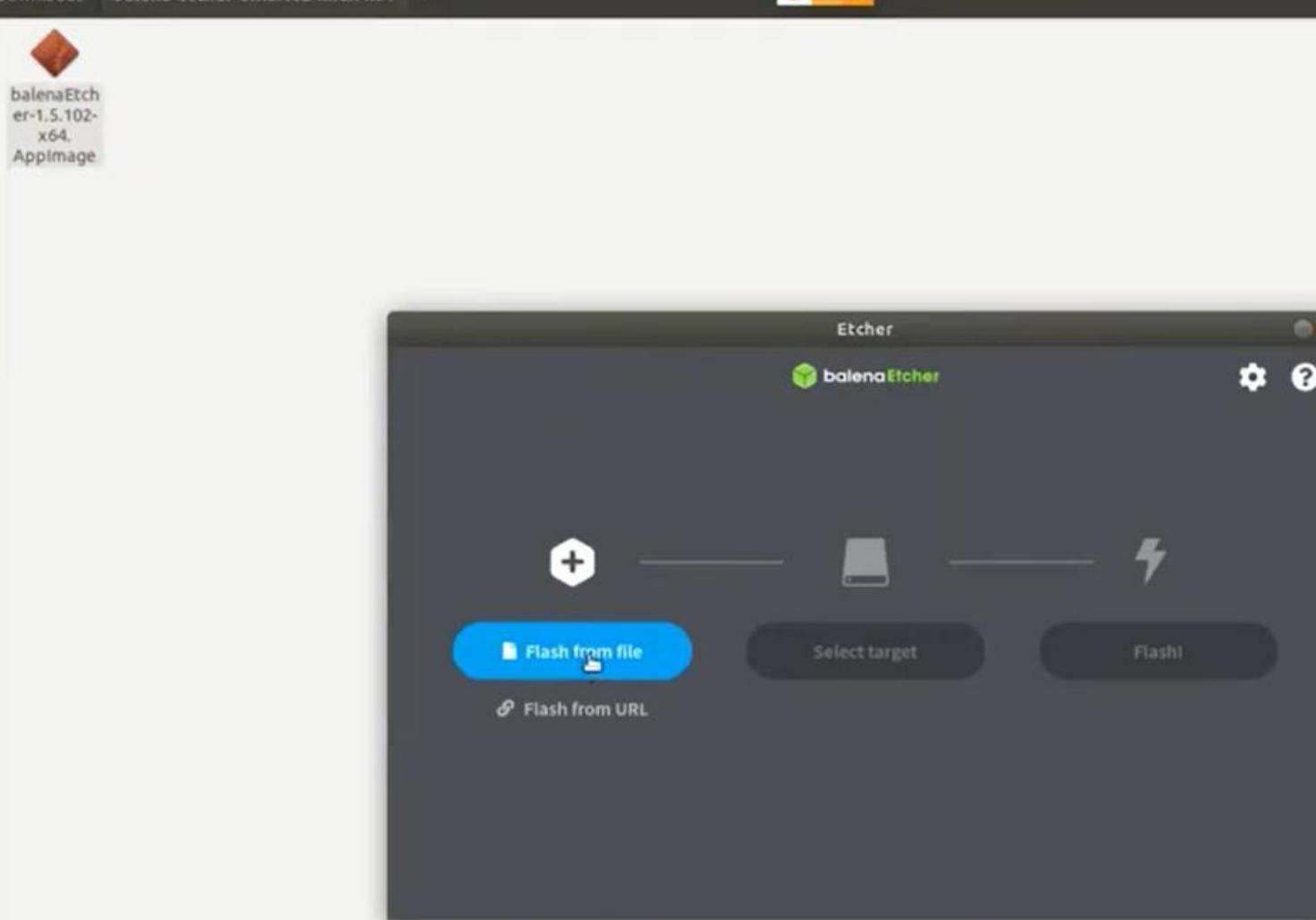


"balena-etcher-electron-1.5.102-linux-x64.zip" selected (89.4 MB)



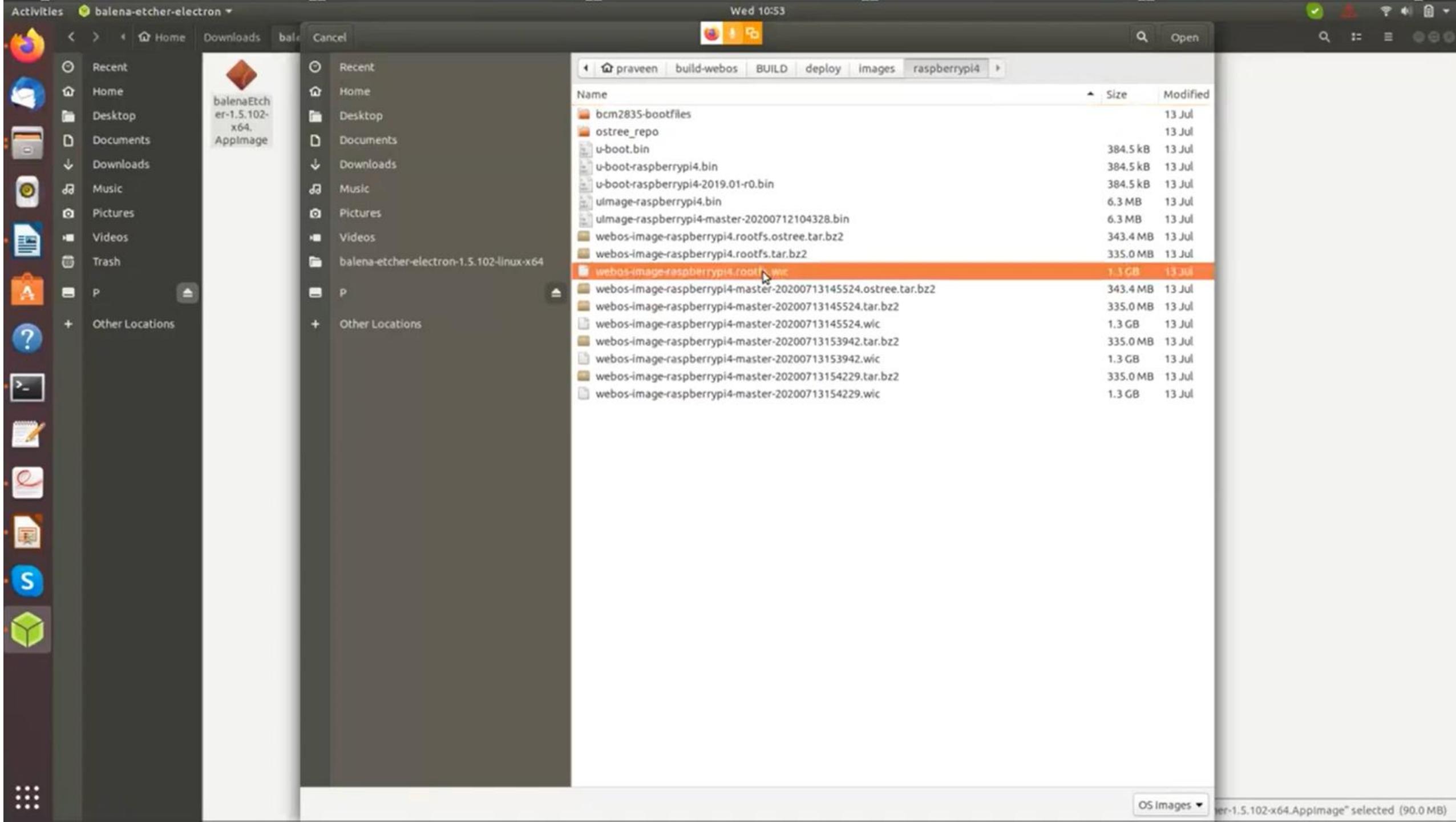


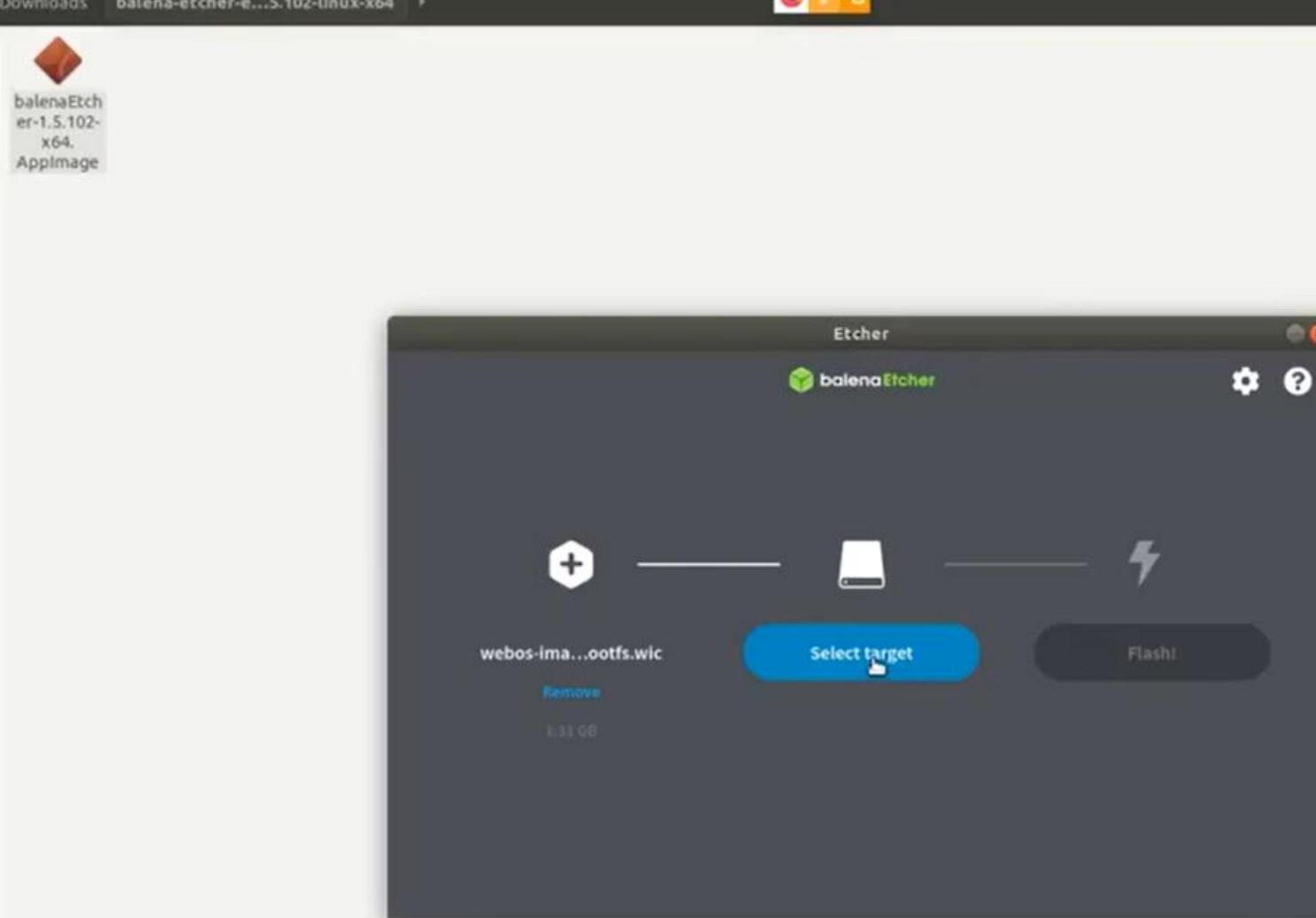
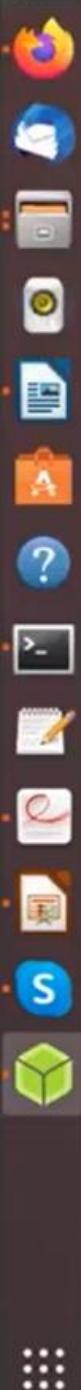
"balenaEtcher-1.5.102-x64.AppImage" selected (90.0 MB)

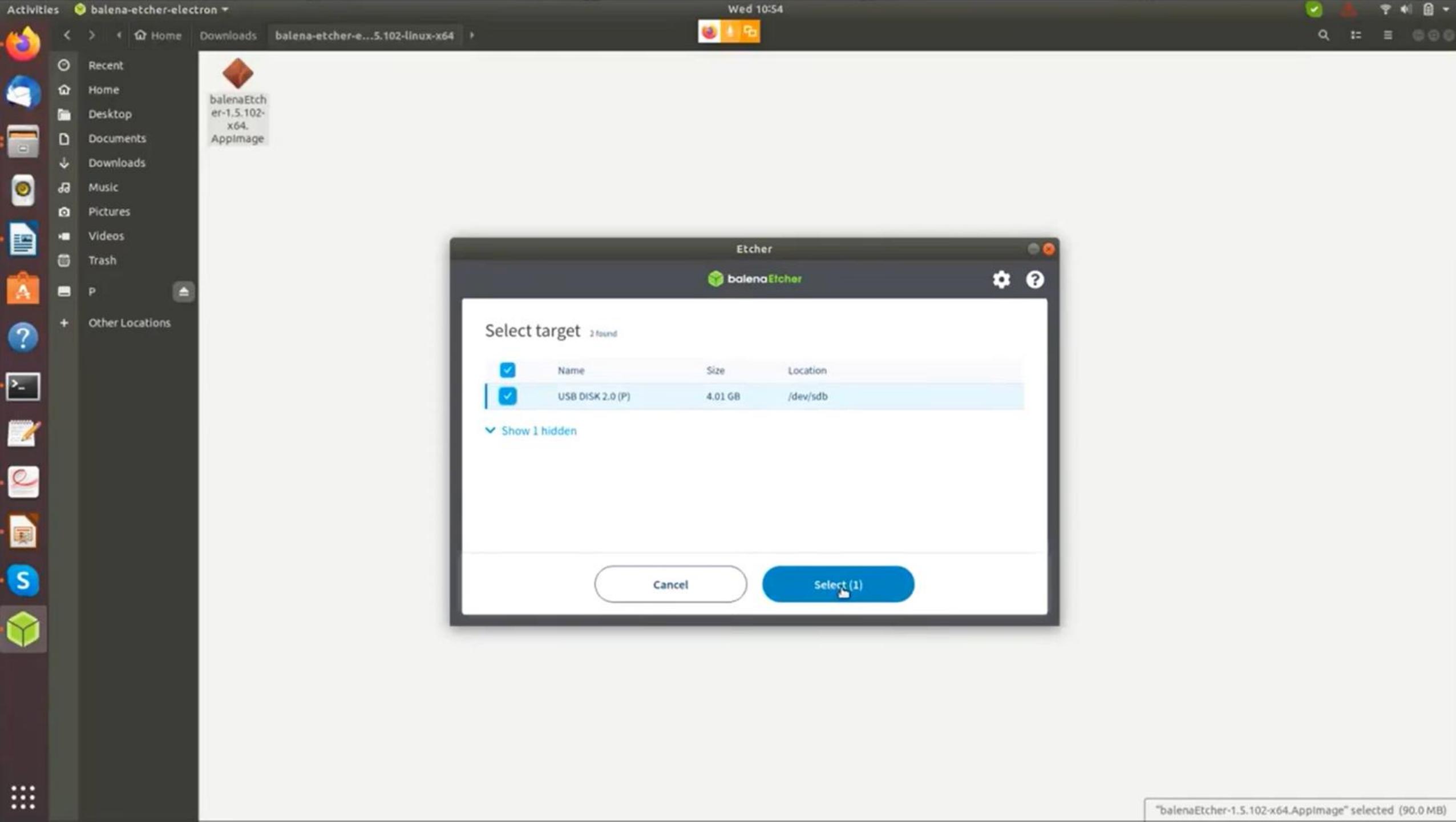


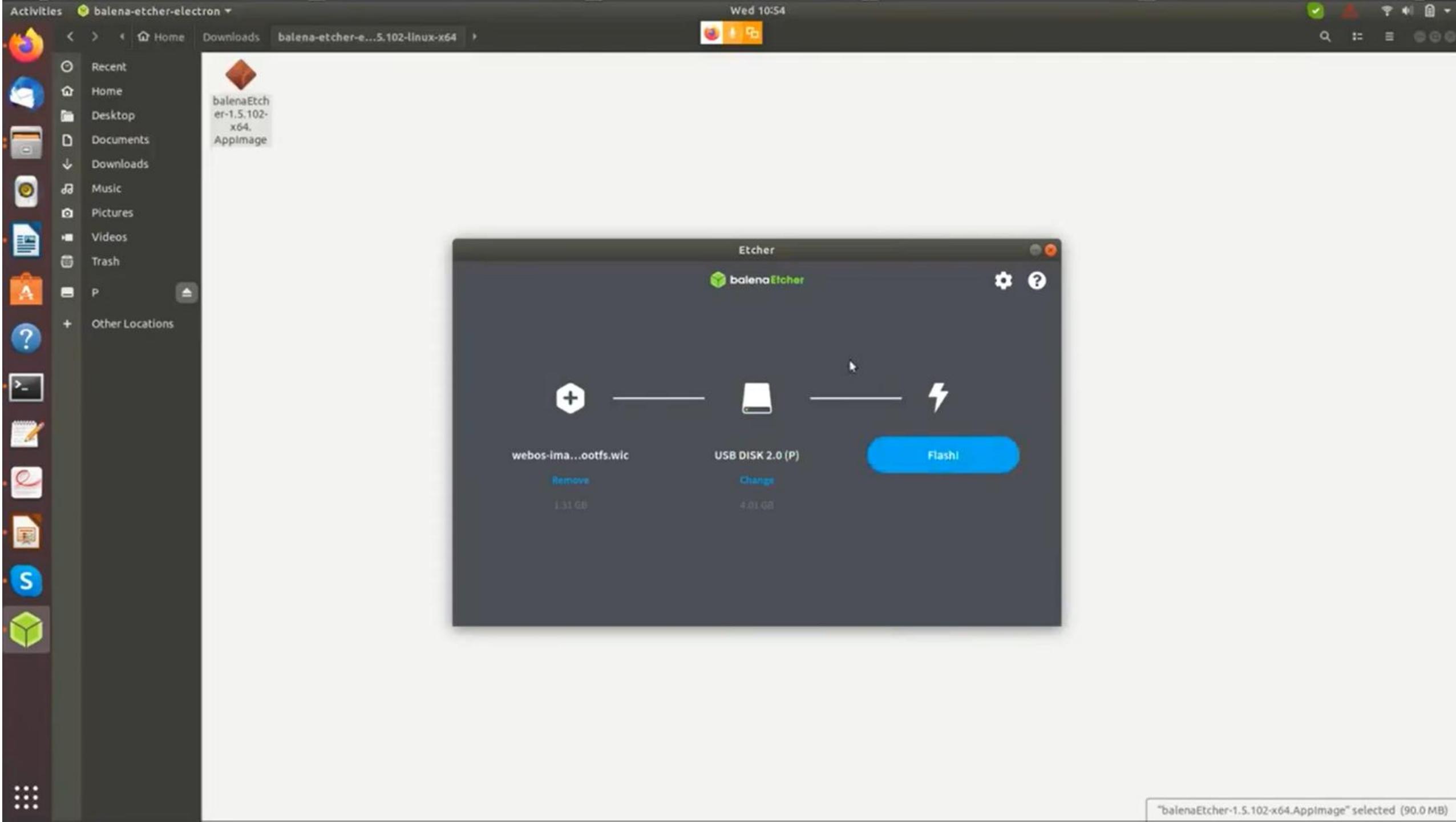
balenaEtcher-1.5.102-x64.AppImage











Activities Firefox Web Browser

Wed 11:05

User Guide | webOS Open Source Edition - Mozilla Firefox

Inbox - praveen.kulkarni X WebOS - Build Status - C Meeting is in progress X User Guide | webOS Open Source Edition - Mozilla Firefox Building webOS OSE | w balenaEtcher - Flash OS X +

https://www.webosose.org/docs/tools/sdk/emulator/virtualbox-emulator/emulator-user-guide/

webOS Open Source Edition

About Docs Blog Community

Search

Tools

SDK

- SDK Download
- Command-Line Interface
- Emulator
- VirtualBox Emulator

User Guide

- QEMUX86 Emulator for Linux (deprecated)
- Beavisier
- Workflow Designer

LS2 Commands

- luna-send Usage Guide
- ls-monitor Usage Guide

Home > Docs > Tools > SDK > Emulator > VirtualBox Emulator

VirtualBox Emulator User Guide

webOS Open Source Edition (OSE) provides an emulator that enables you to develop the webOS application and service on a virtual environment. With the emulator, you can test major features of webOS OSE on your PC without the need of a physical device such as Raspberry Pi.

The emulator runs as a virtual machine on VirtualBox and supports host platforms including Ubuntu Linux, macOS, and Windows.

Note

- The VirtualBox-based emulator is supported by webOS OSE 1.10.0 or higher.
- webOS OSE emulator requires VirtualBox version 6.0 or higher, which can be installed on 64-bit host platforms only.

Key Features

Key characteristics of the emulator are as follows:

- VirtualBox-based emulator
- Emulates major features of webOS OSE platform on PC without Raspberry Pi target device
- Provides graphics functionality with host PC's GPU H/W acceleration
- Supports webOS OSE CLI tool for application and service development

Known Issues

- Touch input is not supported.

Contents

Key Features

- Known Issues
- System Requirements
- Installing VirtualBox
 - Ubuntu Linux
 - macOS
 - Windows
- Installing VirtualBox Extension Pack
- Preparing a webOS OSE Emulator
- Virtual Machine Image
- Setting Up the Virtual Machine in VirtualBox
 - Using the GUI
 - Using the Commands
- Configuring USB Devices
- Updating the Image on the Virtual Machine
- Connecting to the Emulator
 - Connect from the Shell
 - Connect from the Web Browser

Activities Firefox Web Browser

Firefox Web Browser - User Guide | webOS Open Source Edition - Mozilla Firefox

Meeting is in progress

Inbox - praveen.kulkarni | WebOS - Build Status | Meeting is in progress | User Guide | webOS Open Source Edition - Mozilla Firefox | Building webOS OSE | balenaEtcher - Flash OS

https://www.webosose.org/docs/tools/sdk/emulator/virtualbox-emulator/emulator-user-guide/

Wed 11:05

About Docs Blog Community Search

Installing VirtualBox

If you don't have **Oracle VM VirtualBox** already installed on your computer, you need to install it.

Ubuntu Linux

You can either download a package (.deb) file and install the package, or use the `apt-get` command to install Oracle VM VirtualBox from an online package repository (recommended).

Installing a Package

Download the appropriate package from [VirtualBox](#) page and install the package.

Installing from an Online Package Repository

Follow the instructions in [Debian-based Linux distributions](#) section.

macOS

Download the binary package specified as "OS X hosts" from [VirtualBox Download](#) page and install it on your computer.

Windows

Download the binary package specified as "Windows hosts" from [VirtualBox Download](#) page and install it on your computer.

Setting the path on Windows

On Windows, the path to the VirtualBox installation directory is not automatically added to the path during installation. We strongly recommend you set the path manually for later steps, using one of the commands below on a command shell.

Setting the PATH variable in the system environment (run the shell as Administrator)

```
setx /m PATH "C:\Program Files\Oracle\VirtualBox;%PATH%"
```

Contents

- Key Features
- Known Issues
- System Requirements
- Installing VirtualBox
 - Ubuntu Linux
 - macOS
 - Windows
- Installing VirtualBox Extension Pack
- Preparing a webOS OSE Emulator
- Virtual Machine Image
- Setting Up the Virtual Machine in VirtualBox
 - Using the GUI
 - Using the Commands
- Configuring USB Devices
- Updating the Image on the Virtual Machine
- Connecting to the Emulator
 - Connect from the Shell
 - Connect from the Web Browser

User guide 따라하기

Activities Firefox Web Browser

Wed 11:06

Linux_Downloads - VirtualBox - Mozilla Firefox

Inbox - praveen.kulkarni | WebOS - Build Status | Meeting is in progress | User Guide | webOS Open | Linux_Downloads - Oracle | Building webOS OSE | balenaEtcher - Flash OS | +

https://www.virtualbox.org/wiki/Linux_Downloads

 VirtualBox

search... Login Preferences

About Screenshots Downloads Documentation End-user docs Technical docs Contribute Community

VirtualBox

Download VirtualBox for Linux Hosts

Note: The package architecture has to match the Linux kernel architecture, that is, if you are running a 64-bit kernel, install the appropriate AMD64 package (it does not matter if you have an Intel or an AMD CPU). Mixed installations (e.g. Debian/Lenny ships an AMD64 kernel with 32-bit packages) are not supported. To install VirtualBox anyway you need to setup a 64-bit chroot environment.

The VirtualBox base package binaries are released under the terms of the [GPL version 2](#).

Please choose the appropriate package for your Linux distribution.

VirtualBox 6.1.12 for Linux

- [Oracle Linux 8 / Red Hat Enterprise Linux 8 / CentOS 8](#)
- [Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7](#)
- [Oracle Linux 6 / Red Hat Enterprise Linux 6 / CentOS 6](#)
- [Ubuntu 19.10 / 20.04](#)
- [Ubuntu 18.04 / 18.10 / 19.04](#)
- [Ubuntu 16.04](#)
- [Ubuntu 14.04 / 14.10 / 15.04](#)
- [Debian 10](#)
- [Debian 9](#)
- [Debian 8](#)
- [openSUSE 15.0](#)
- [openSUSE 13.2 / Leap 42](#)
- [Fedora 32](#)
- [Fedora 31](#)
- [Fedora 29 / 30](#)
- [Fedora 26 / 27 / 28](#)
- [All distributions \(built on EL6 and therefore not requiring recent system libraries\)](#)

You might want to compare the checksums to verify the integrity of downloaded packages. The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!

- [SHA256 checksums](#), [MD5 checksums](#)

Oracle Linux

Users of Oracle Linux 6 and 7 can use the Oracle Linux yum [repository](#) and enable the [ol6_developer](#) channel for Oracle Linux 6 or the [ol7_developer](#) channel for Oracle Linux 7. After that, do

```
yum install VirtualBox-6.1
```

to get the latest maintenance release of VirtualBox 6.1.x installed.

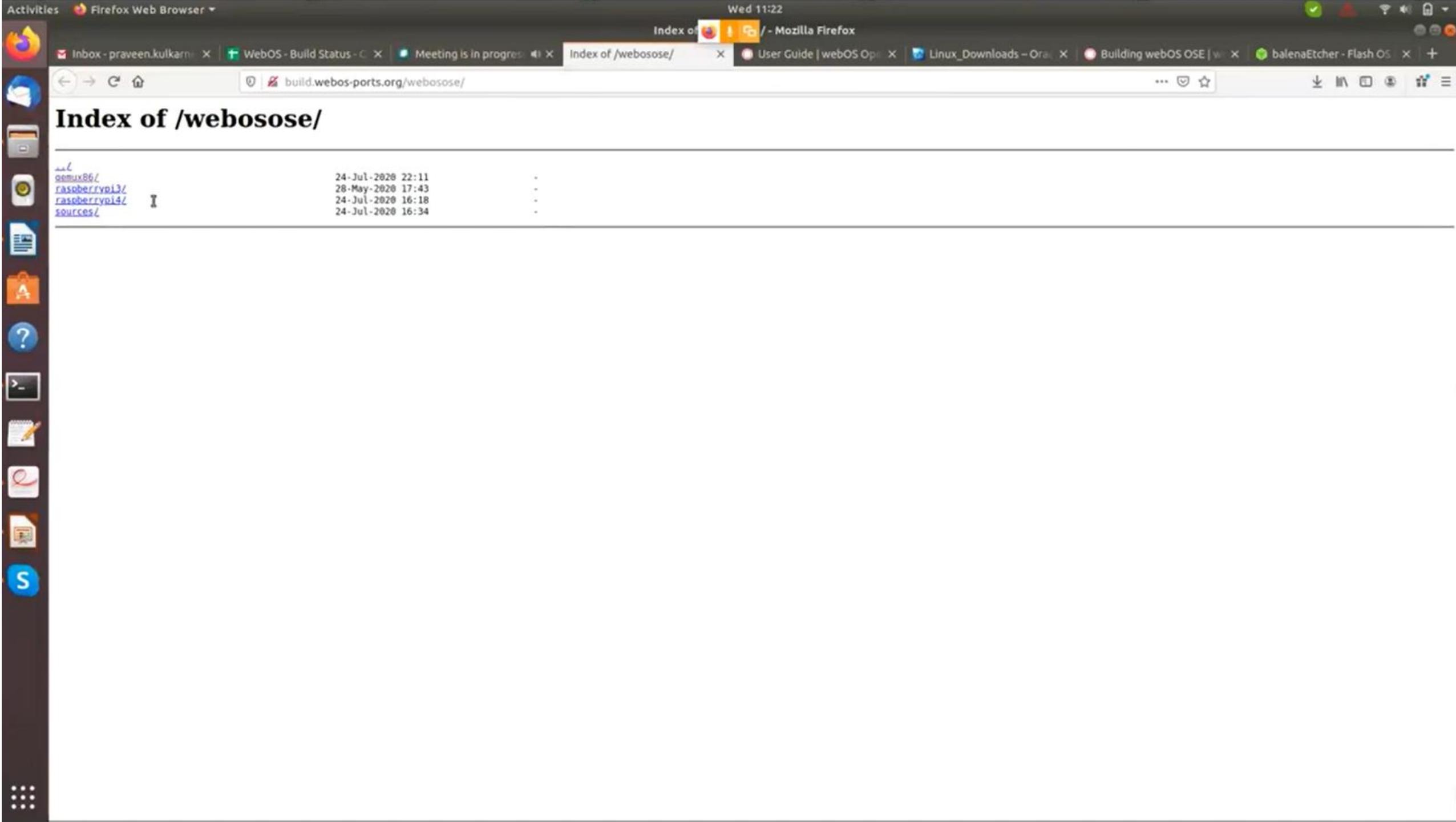
Debian-based Linux distributions

Add the following line to your `/etc/apt/sources.list`. According to your distribution, replace '`<mydist>`' with 'eoan', 'bionic', 'xenial', 'buster', 'stretch', or 'jessie' (older versions of VirtualBox supported different distributions):

```
deb [arch=amd64] https://download.virtualbox.org/virtualbox/debian <mydist> contrib
```

This Oracle software license is not available for download.

https://download.virtualbox.org/virtualbox/6.1.12/virtualbox-6.1.12-139181-Ubuntu-bionic_amd64.deb



DAY3-2

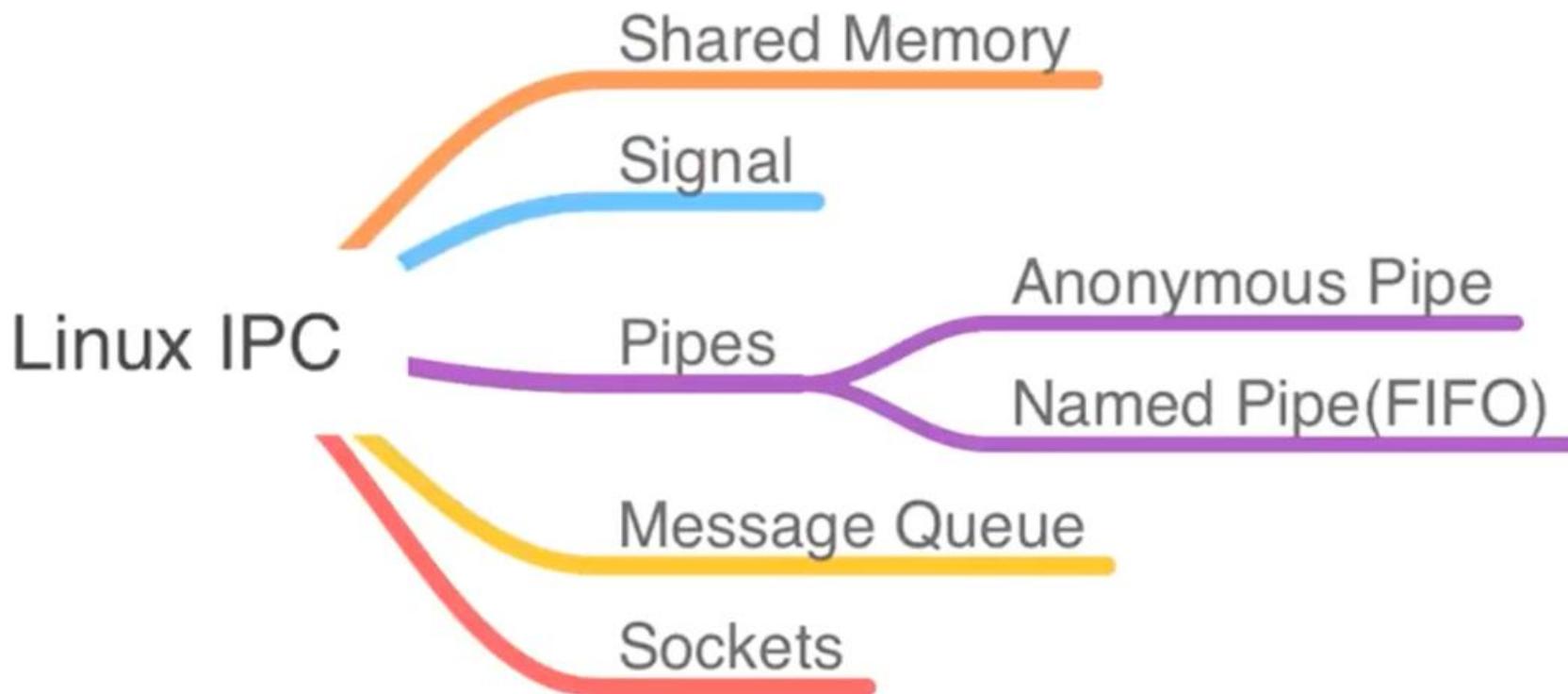


Inter Process Communication

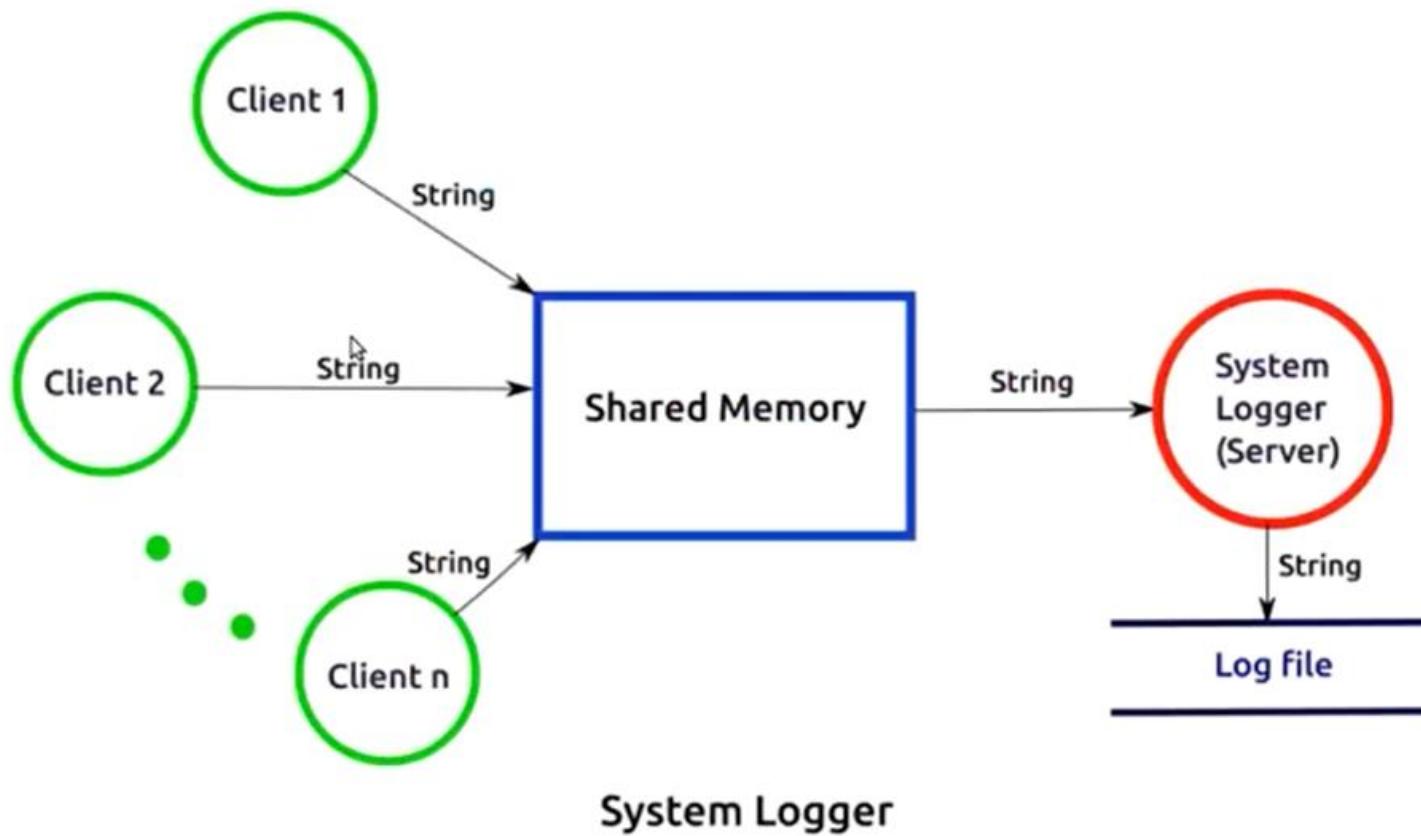
IPC



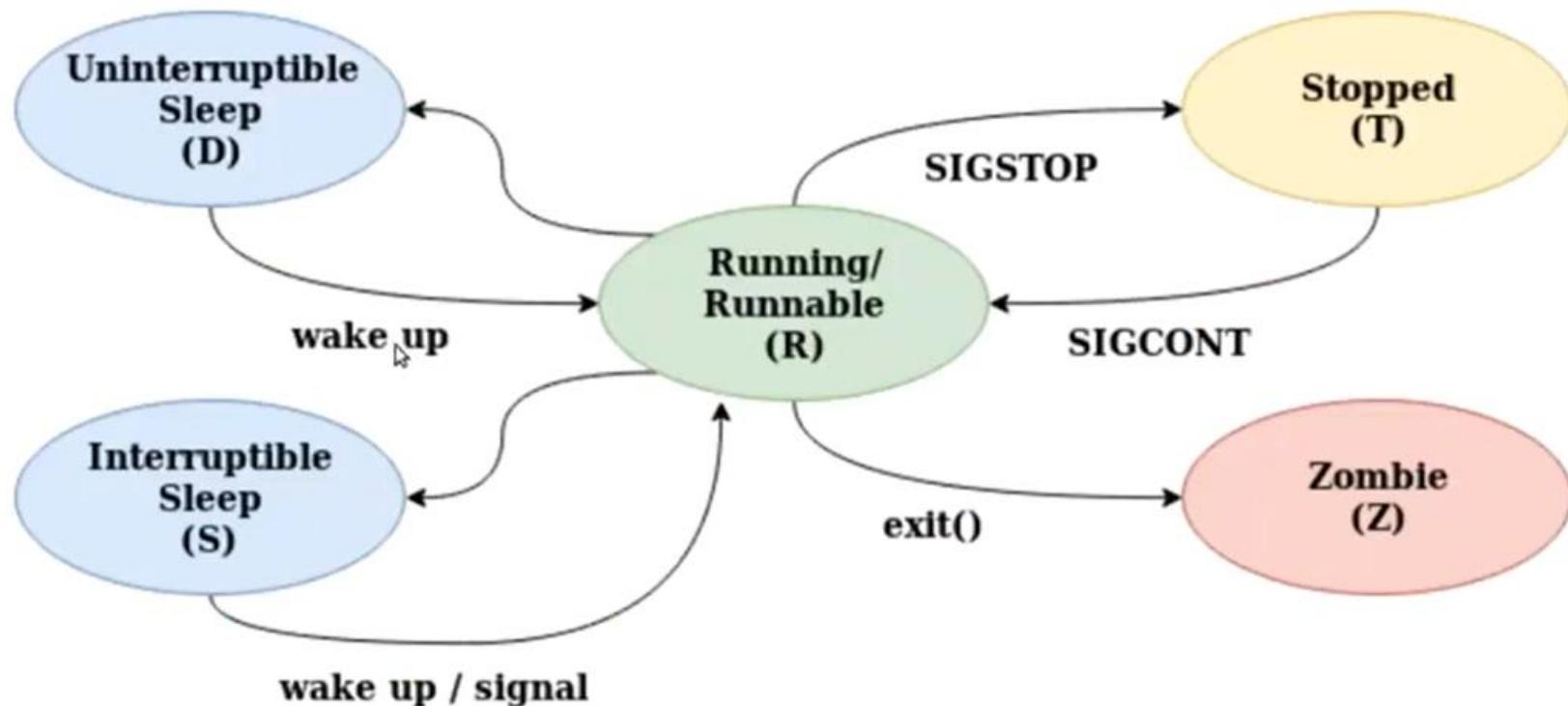
Different Types of IPC in Linux



Shared Memory



Signal



Pipes

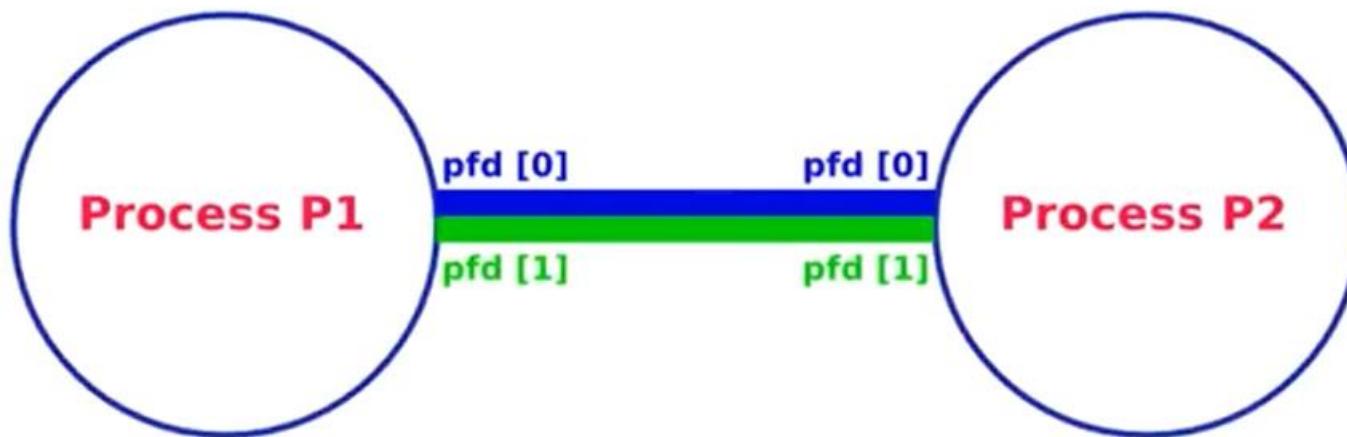
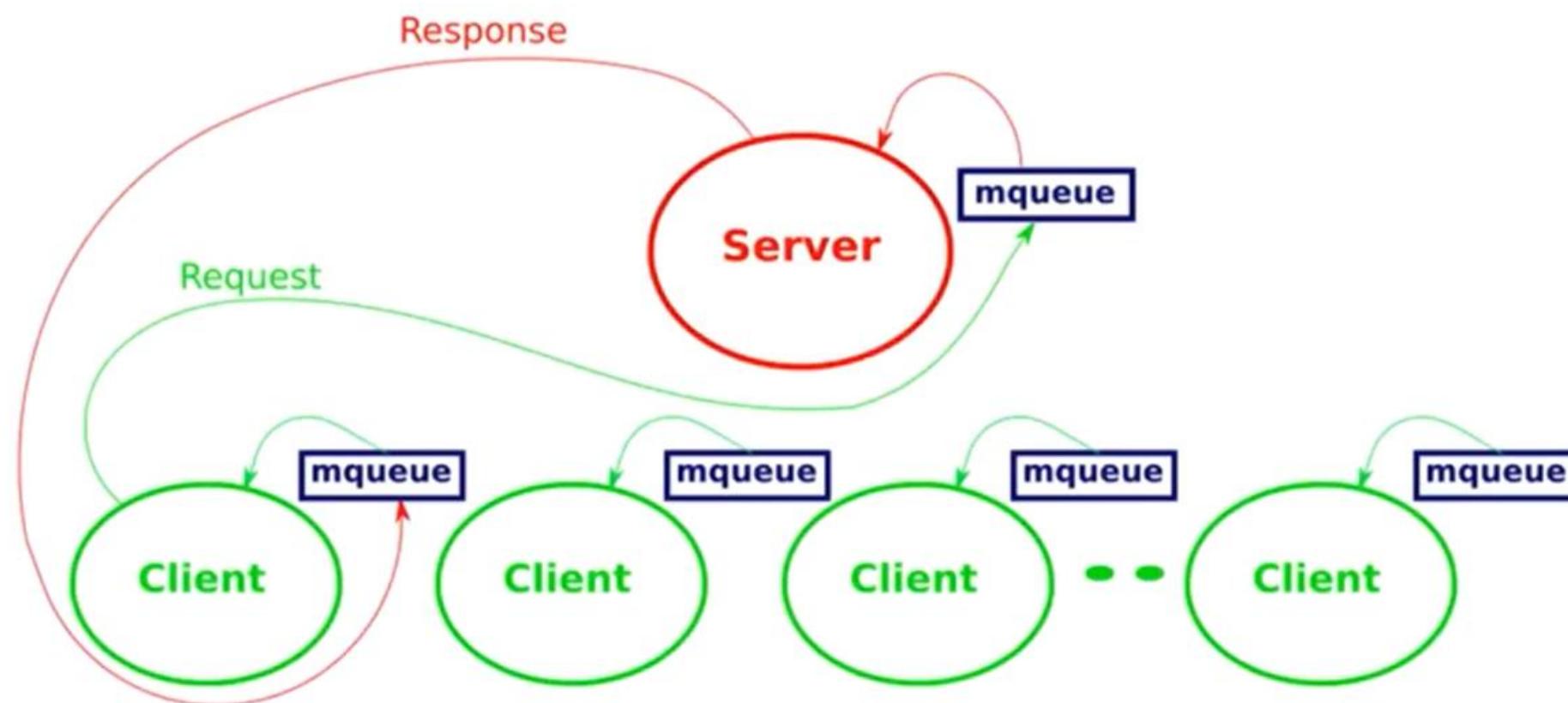


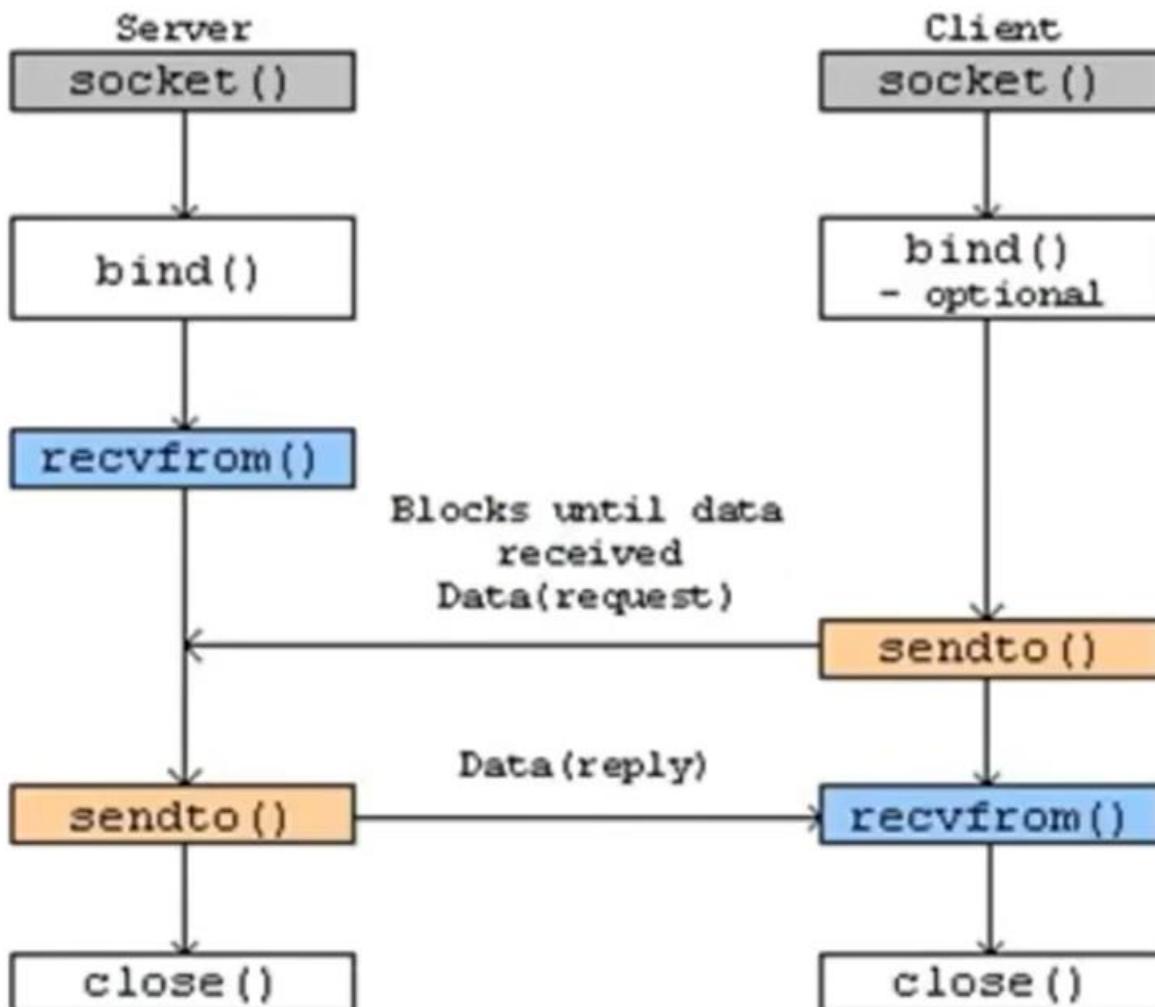
Fig 1: Two processes connected with a pipe

Message Queue



Interprocess Communication between client and server using Message Queues

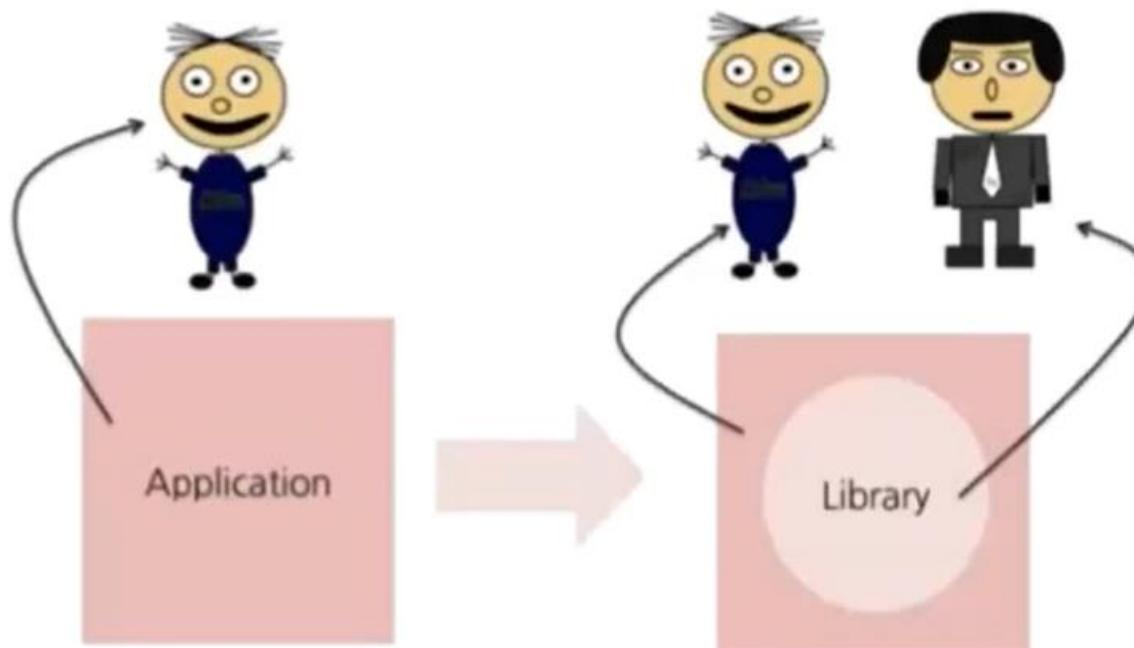
Socket



Bus

Why do we need Bus?

- ❑ Complex systems - more developers, more libraries, more services -means need for interaction
- ❑ Need to exchange data



Example

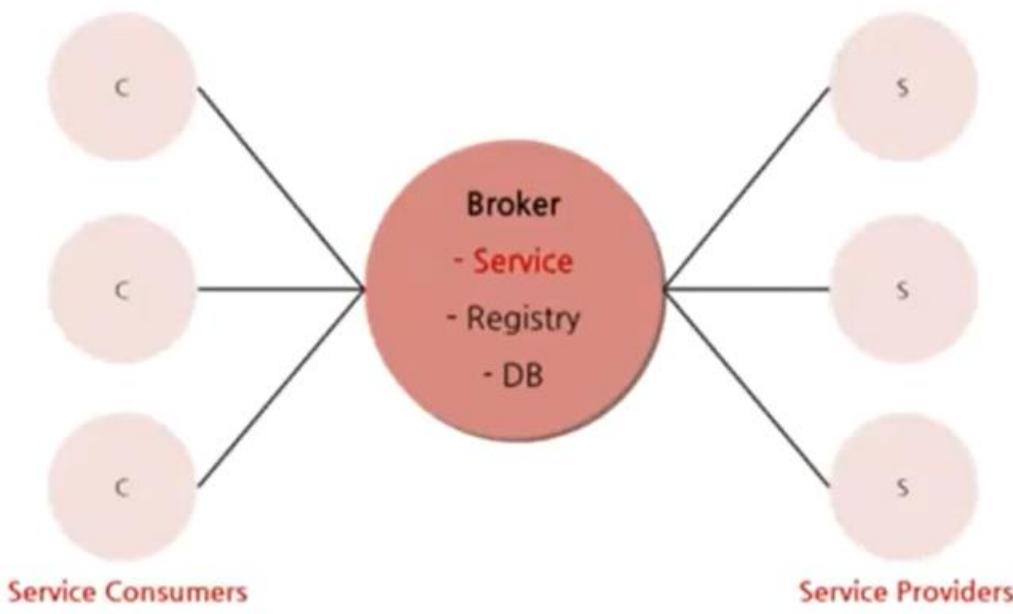
- Ensure that products from various developers can work with each other
- Allows anonymous third party programs to work together



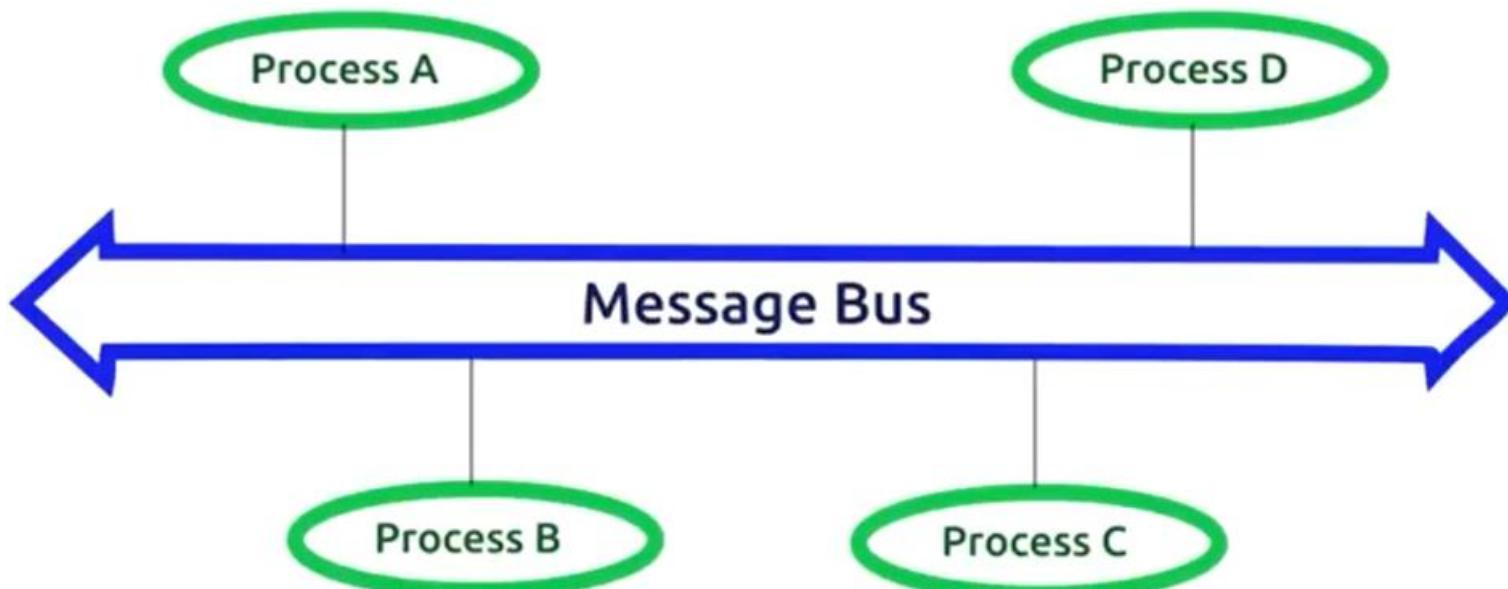


Bus System roles

- Offer service
- Broker acts as a bridge - It can be a registry, database or daemon
- In bus system broker is a daemon



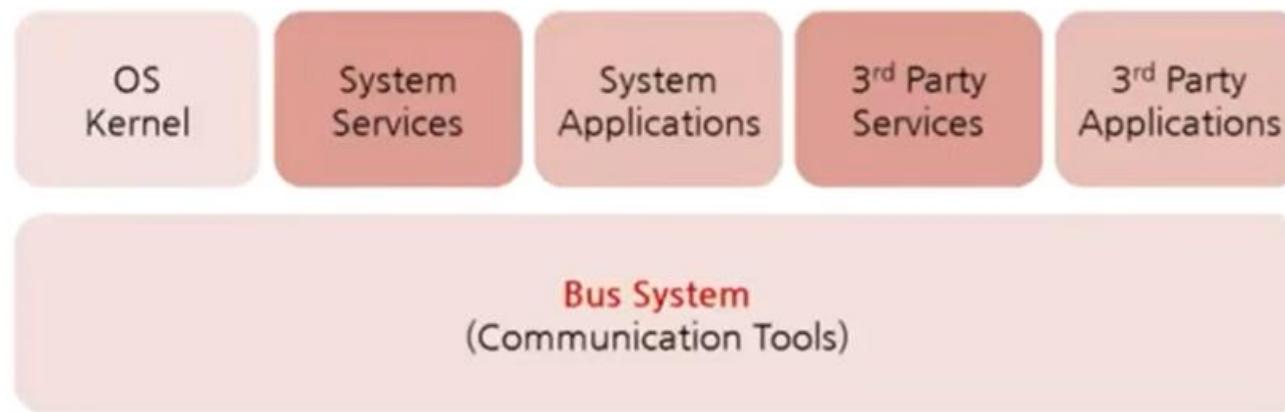
D-Bus in Linux



Interprocess communication using D-Bus

Luna Bus in webOS

- IPC mechanism in webOS is Luna-Bus
- Conceptually similar to D-BUS
- Uses protocol – ‘luna://..’



	Linux	Tizen	Meego	Android	webOS
Bus System	D-Bus	D-Bus	D-Bus	Binder	LunaBus

Luna Bus Features

- Improve performance, direct connections between services - *Peer to Peer*
- UNIX Domain sockets are used for communication.
- **Service, Object, Method Addressing** - As in *DBUS*

char*getServiceName(void)

Return service name

Char*hello(char*name)

Return "hello YOUR_NAME"

Char*echo (char*message)

Return message itself.

long getCurrentTime(void)

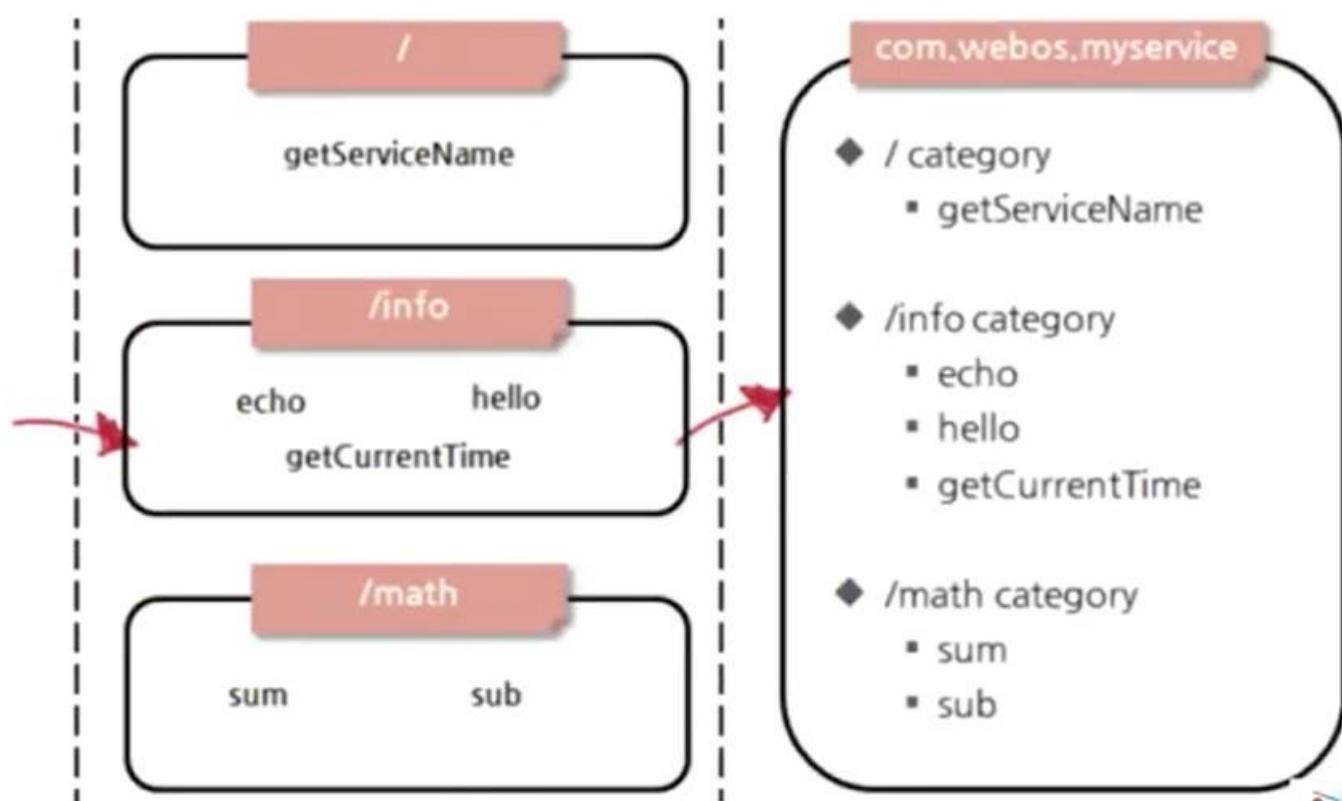
Return current system hour

int sum(int a, int b)

Return result value of a+b

int sub(int a, int b)

Return result value of a-b



Luna Bus Features...

- **Subscription API**- As in DBUS
- a client-service operation consists of a client calling a service and receiving a relevant response.
- *Example:*

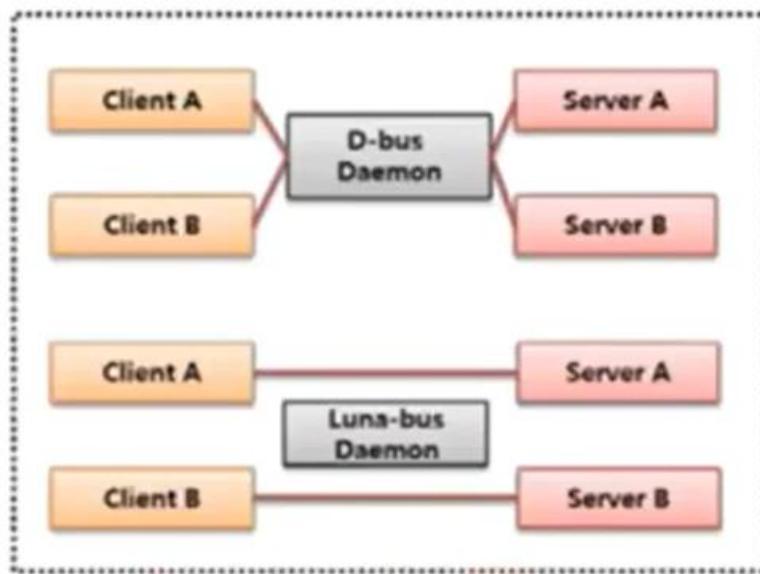


```
var curAddressList:  
  
function onTick() {  
    // fetch phonebook  
    var newList = navigator.addressbook.getList();  
  
    // compare with the existing phonebook  
    var isSame = compare(curAddressList, newList);  
    if (!isSame) {  
        curAddressList = newList;  
        drawAddressList();  
    }  
}  
  
setInterval(onTick(), 1000); // Call onTick() every one second
```

```
var curAddressList:  
  
function onSuccess(newAddressList) {  
    drawAddressList();  
}  
  
function onFailure(message) {  
    alert('Failed because: ' + message);  
}  
  
navigator.addressbook.getList(onSuccess, onFailure);
```

Luna Bus Features...

Performance- Exclusive to LUNA



Average response speed (micro second) according to the number of D request for repetition (req size = 128)

	1	10	50	100	1000	5000	10000
x86-dbus	1668	1503	1032	1521	2063	2258	2264
x86-luna	442	133	84	147	102	100	107

Luna Bus Features...

- **Multiple Replies** - *Exclusive to LUNA*

< CASE 1 >

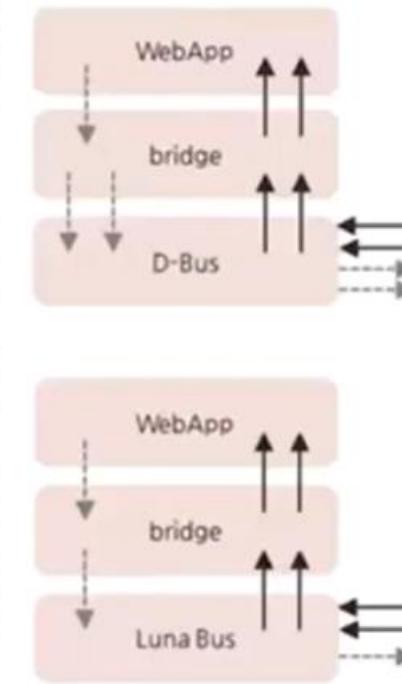
```
function tick() {
    var position = Sensor.getData();
    updateUI(position);
}
setInterval(function () {tick()}, 1000);
```

< CASE 2 >

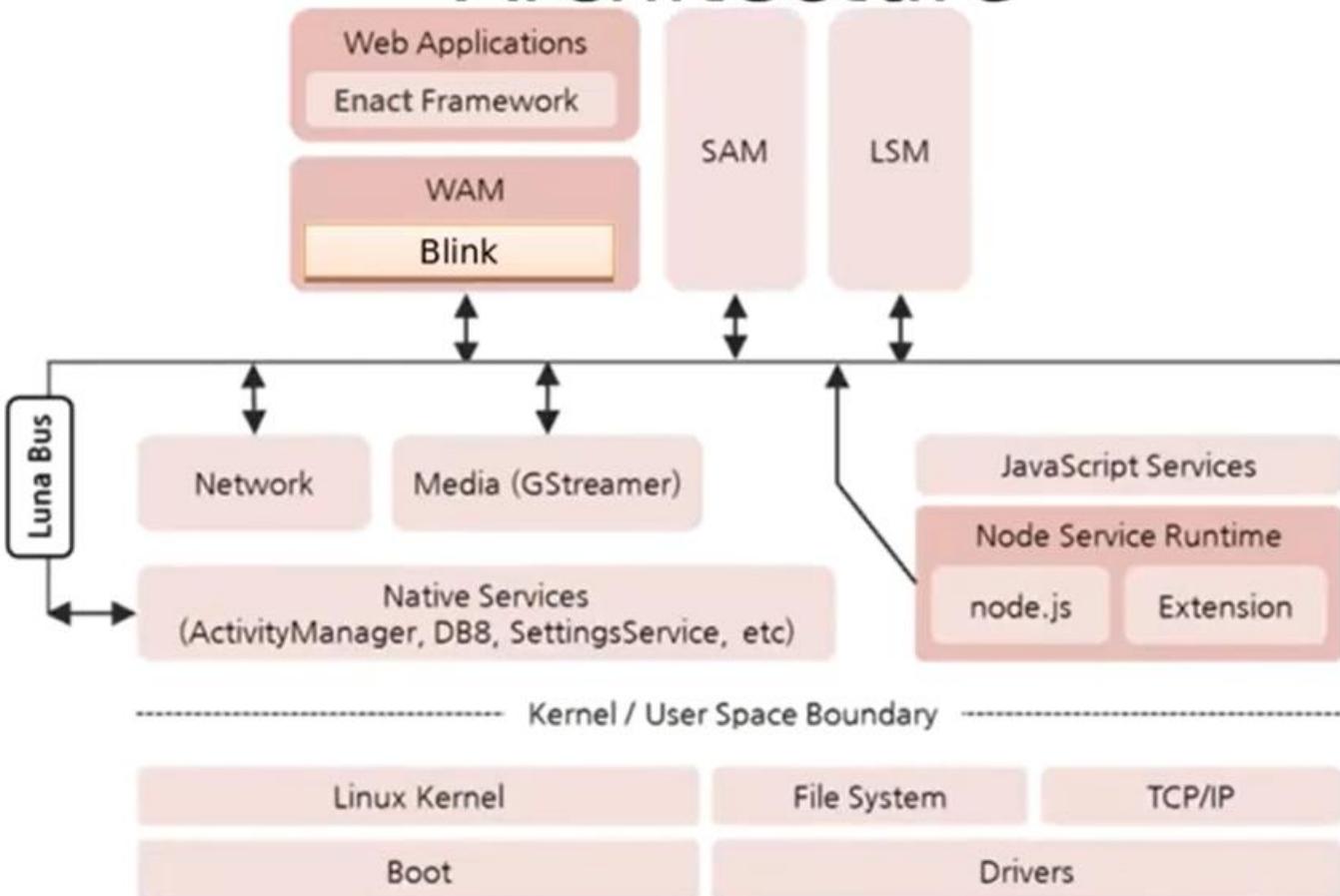
```
function onReceived(position) {
    updateUI(position); tick();
}
function tick() {
    Sensor.getData("onReceived");
}
```

< CASE 3 >

```
function onChanged(position) {
    updateUI(position);
}
Sensor.setEventHandler("onChanged", 1000);
```



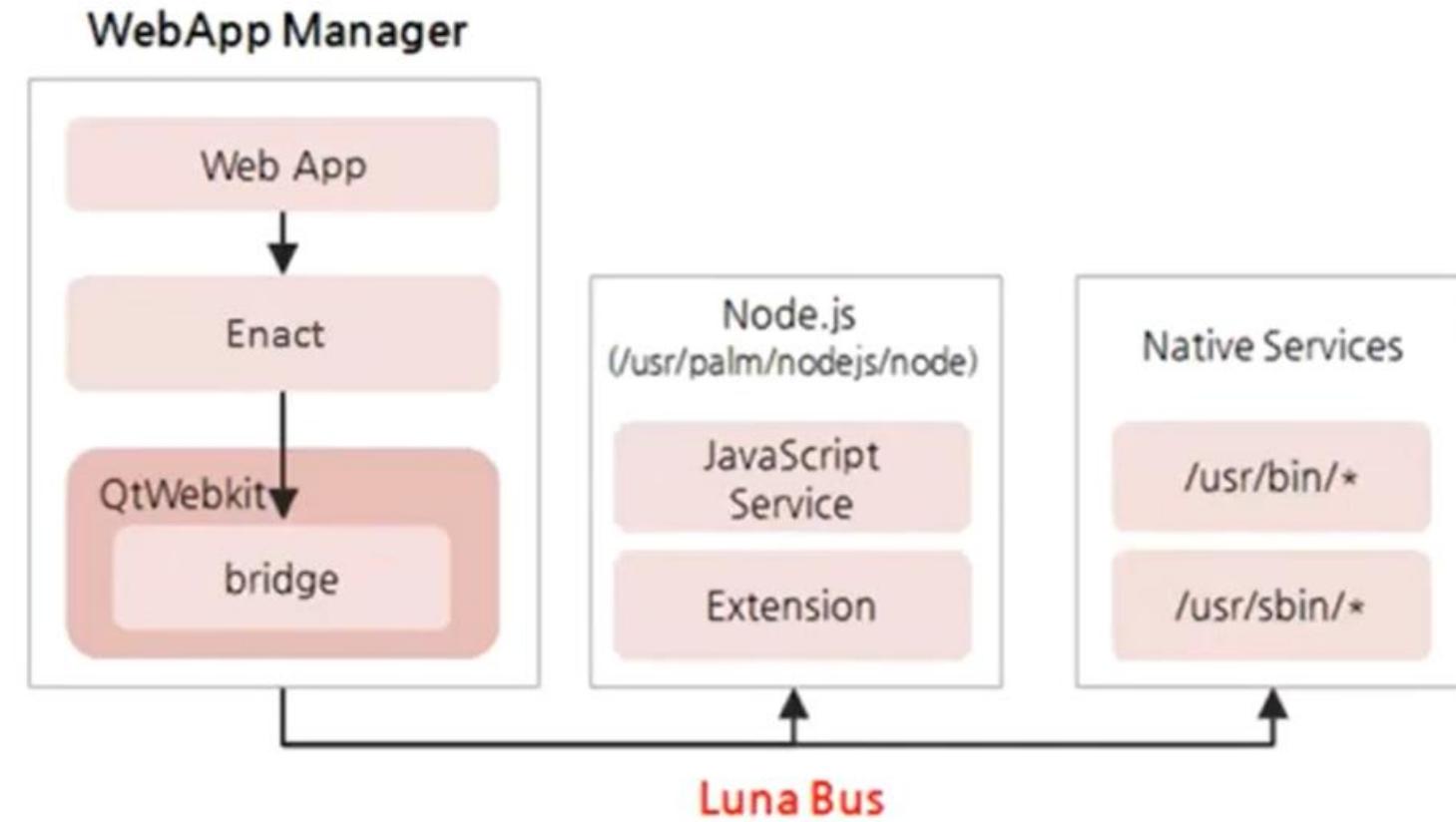
Luna Bus Position in webOS Architecture



Components and Processes

In the architecture above, most of the components can be considered independent processes. However, components that don't communicate directly with Bus, such as web apps or the Enact framework, are not processes. These can be considered threads in WebKit within the WAM.

Luna Bus Position in Web App Runtime



- Context is Web App Runtime
- Managed by Web App Manager (WAM)
- Each app has unique identifier

Access Control





What is ACG (Access Control Groups) ?

- Provides fine grained access control of security permissions
- Allows to define security group with different set of methods
- Groups which an app wants to access is placed in the **appinfo.json** file under **“requiredPermissions”**
- Achieved using 4 different types of access files - for applications
 - **AppInfo.json** - Similar to manifest.xml in Android, part of application package
 - **Role file** - Autogenerated from appinfo.json
 - **Client Permissions file** -

Open

bands.txt

Save

ssh -p 6622 root@localhost

```
1. luna-send -n 1 -f luna://com.webos.service.systemservice/time/getSystemTime '{}'  
2. luna-send -i -f luna://com.webos.service.systemservice/time/getSystemTime '{"subscribe": true}'  
3. luna-send -n 1 -f luna://com.webos.appInstallService/status '{}'  
4. luna-send -n 1 luna://com.webos.service.alarm/set '{  
    "key":"test",  
    "uri":"luna://com.webos.service.testsleepd/firealarm",  
    "params":{},  
    "at":"07/29/2020 04:40:00",  
    "wakeup":true  
}'  
5. luna-send -i -f luna://com.webos.service.bluetooth2/device/getStatus '{}'  
6. luna-send -n 1 -f luna://com.webos.service.camera2/getCameraList '{}'
```

DAY4

Career as a Software Developer at LGSI

global online internship program

30/07/2020

Braulio Fernandes
CSP / Evangelism team

Introduction

In this session, we hope to satisfy your questions on the following:

- ✓ How you benefit by working at LGSI
- ✓ How LGSI benefits from you
- ✓ Some details of the webOS Platform

How will you contribute to LG ...

You will be contributing to the growth of some very exciting LG consumer and commercial devices such as:

- TV
- Signage
- Robots
- Automobile entertainment
- Refrigerator
- Mobile
- and so on...

Significantly, these contributions will be on the **webOS** platform that is used on a number of LG products.

What is webOS ...

- It is a powerful and versatile software platform.
- It is a web-centric and usability-focused software platform.
- It is used in multiple LG products such as Smart TVs, Signage Devices, automotive solutions, Robots, and more.
- It is also available to global developer community as a open source platform called as webOS OSE (<http://webosose.org>).
- In addition, you can contribute ideas that can be further developed by LGSI.

Functional details of webOS ...

- Contains powerful features that are invaluable in the fields of multimedia, connectivity, and application frameworks.
- They are used in many industry verticals such as IoT, robotics, and automotive products.
- Core Platform
 - Encompasses service components that support file system, DB, IPC, device manager, sensors, and power management.
- Connectivity
 - Supports connectivity features with Bluetooth, NFC, Wi-Fi/networking, IoT, CAN/eAVB, Telephony, and Location based services.

Functional details of webOS ... (continued)

- Multimedia
 - Includes multimedia features such as Digital Rights Management and Direct rendering with the supporting components of audio & video.
- Apps and Framework
 - Application layer development is supported with Enact app framework, SDK tools and test automation.
- Analytics
 - Platform expansion involves AI frameworks with data analytics and context-aware systems.

Besides there are various initiative teams that work on upcoming projects and business-cases.

Skills/Knowledge that will come handy ...

- Platform development:
 - Native languages (C, C++)
- App Development:
 - JS
 - HTML5 and CSS
 - Node.js
 - EnactJS
- Other requirements:
 - Understanding of the Linux Operating System
 - Working on Open Source Software
 - GIT processes

Further growth of webOS ...

- We are expanding webOS to create new business opportunities for the local market.
- We are enhancing the webOS ecosystem based on the Open-Evolve-Connect strategy through developer summits, tech forums, university collaboration, start-up collaboration, hackathons, internships, and so on.
- The CSP division works on niche technology areas such as swarm intelligence, context-aware systems, IoT edge accelerator platform, blockchain, and cloud services.
- Going forward, we plan to further expand webOS towards automotive, IoT products, advanced robotics and vehicle automation. Our ultimate goal is to develop the webOS as a platform for all LG consumer devices.

Who will you be working with ...

Professionals who:

- Have expertise in their chosen fields of work.
- Highly motivated to get results and contribute to their growth and that of the company.
- Love experimenting and trying out new technologies.
- Are eager to share their knowledge with others.
- Are eager to learn more and thus contribute more.

What is expected of you ...

- A willingness to learn new technology and skills.
- A sense of ownership of the work assigned to you.
- An eagerness to contribute ideas to the growth of the company.
- Take part in certifications that can boost your confidence and help you become an expert in different areas.
- An energy that makes you a fun person to collaborate with.
- Become a part of initiative teams to further develop new ideas.

We invite you to join us at
LGSI
for a career full of
promise !!!