RUNNING LINUX ON YOUR SMARTPHONE

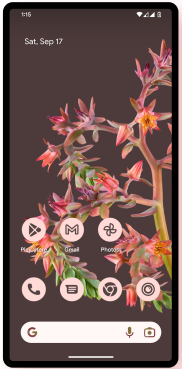
REG Lunchtime Tech Talk

David Llewellyn-Jones

9th May 2023

SAILFISH OS

SMARTPHONE OPERATING SYSTEMS



android

iOS

SAILFISH OS

ANDROID

1. AOSP (Android Open Source Project)
2. Linux kernel
3. GMS (Google Mobile Services)
4. Apps written in Java, Kotlin, others
5. Linux, but not as we know it

SAILFISH OS

IOS

1. Darwin (BSD) kernel
2. Cocoa Touch user interface
3. Apps written in Objective-C, Swift
4. Closed source, closed ecosystem

iOS

SAILFISH OS

LINUX

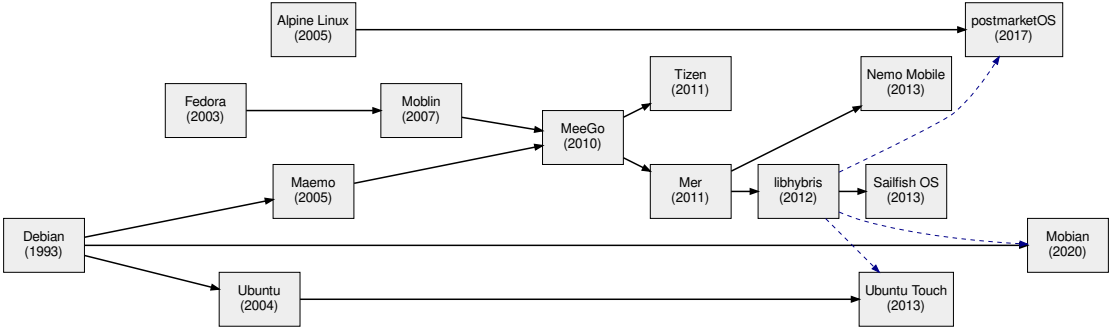
1. Mobian (Debian + Phosh)
2. postmarketOS (Alpine Linux + Plasma Mobile)
3. Ubuntu Touch (Ubuntu + Lomiri)
4. Nemo Mobile (Manjaro + Nemo)
5. Sailfish OS (Mer + Silica)

WHY ARE THEY INTERESTING?

1. Linux, glibc, GNU Core Utils, systemd
2. Open Source
3. Open development models
4. Open ecosystems
5. Community-oriented business models
6. Most Linux software just compiles and runs

SAILFISH OS

LINUX ON MOBILE HISTORY



SAILFISH OS

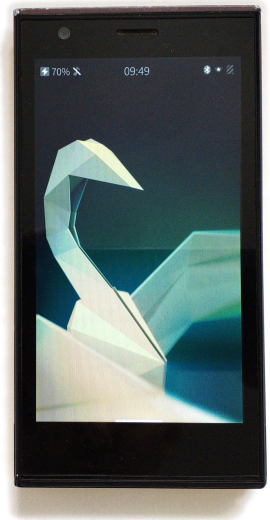
MAEMO 4.1

SAILFISH OS

MAEMO 5

SAILFISH OS

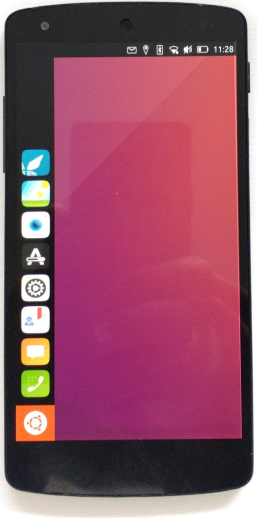
SALFISH OS 2.2



SAILFISH OS


SØKOS
PARTIO KUULUU KAIKILLE
PARTIO KOULU
EUROPARTI
WWW.EUROPARTI.FI
HENNY'S PUB

UBUNTU TOUCH 16.04

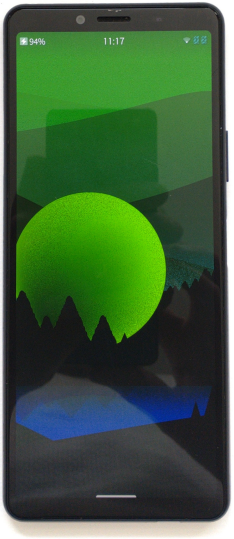


SAILFISH OS

POSTMARKETOS 22.12.2

SAILFISH OS

SAILFISH OS 4.5



SAILFISH OS

SAILFISH OS STACK

1. Android drivers
2. Libhybris and Libgbinder
3. Linux kernel 4.19.248
4. glibc
5. systemd
6. busybox

SAILFISH OS

SAILFISH OS STACK

7. Qt middleware
8. Wayland + Lipstick compositor
9. Silica user interface
10. Lipstick launcher
11. Gecko-based Web browser
12. Android App Support

SAILFISH OS

LIBHYBRIS AND LIBGBINDER

Libhybris

1. Allows AOSP drivers to be used with glibc Linux
2. Dynamic loading of Android libraries
3. Overrides bionic symbols with glibc symbols

Libgbinder

1. Android Binder Protocol
2. HAL Interface definition language
3. Switching from linking to Binder
4. Modems switched from socket to Binder in Android 8

SAILFISH OS

DBUS

Inter-process communication

1. Object-oriented and typed
2. System bus, per-login session bus
3. Peer-to-peer or bus-oriented
4. Properties, method calls, signals, introspection

```
class FlashlightDBusAdaptor: public QDBusAbstractAdaptor
{
    Q_OBJECT
    Q_CLASSINFO("D-Bus Interface",
        "com.jolla.settings.system.flashlight")

public:
    Q_PROPERTY(bool flashlightOn READ flashlightOn)

    FlashlightDBusAdaptor(QObject *parent);
    bool flashlightOn() const;

public slots:
    bool toggleFlashlight();

signals:
    void flashlightOnChanged();
};
```

```
dbus-send --session --type="method_call" --print-reply \
    --dest="com.jolla.settings.system.flashlight" \
    "/com/jolla/settings/system/flashlight" \
    "com.jolla.settings.system.flashlight.toggleFlashlight"
```

SAILFISH OS

QT

1. Middleware libraries
2. User Interface toolkit
3. Cross-platform C++, PyOtherSide
4. Meta-Object Compiler
5. QObject model

```cpp
#include <QObject>

class Example : public QObject
{
    Q_OBJECT

    Q_PROPERTY(bool selected READ selected WRITE setSelected
            NOTIFY selectedChanged)
public:
    explicit Example(QObject *parent = nullptr)
        : QObject(parent)
        , m_selected(false)
    {}

    bool selected() const { return m_selected; }

    void setSelected(bool selected) {
        if (m_selected != selected) {
            m_selected = selected;
            emit selectedChanged();
        }
    }

signals:
    void selectedChanged();

private:
    bool m_selected;
};
```

SAILFISH OS

QML

1. Declarative user interface language
2. *Components*, *properties* and *functions*
3. Property changes trigger recalculation
4. Imperative JavaScript functions
5. Rendered components positioned using anchors

```
Page {
    property int count: 0

    Example {
        id: example
        selected: toggle.checked
    }

    Timer {
        id: countdown; interval: 1000; repeat: true
        onTriggered: if (count > 0) count -= 1
        running: example.selected == true
        onRunningChanged: count = 10
    }

    Column {
        id: column; anchors.fill: parent
        PageHeader { title: qsTr("REG Tech Talk Example") }
        TextSwitch {
            id: toggle
            text: qsTr("Activated")
        }

        Label {
            x: Theme.horizontalPageMargin
            text: example.selected
                  ? qsTr("This message will self destruct in
                         %1 seconds".arg(count))
                  : qsTr("Disarmed")
        }
    }
}
```

SAILFISH OS

HACKING THE HOMESCREEN

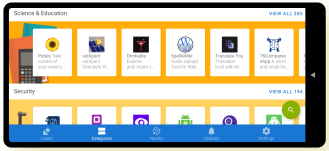
1. QML interface is highly hackable
2. A simple example:

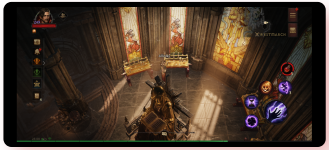
/usr/share/lipstick-jolla-home-qt5/statusarea/StatusArea.qml

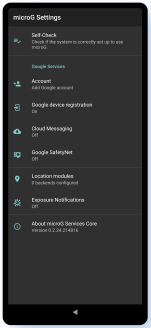
```
            Label {
                id: name
                text: "David's phone"
                anchors.verticalCenter: parent.verticalCenter
            }
```

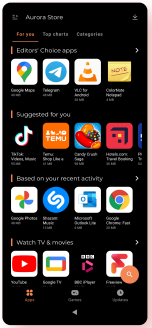
SAILFISH OS

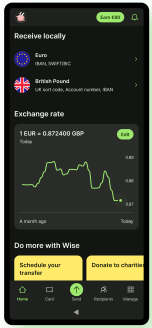
ANDROID APP SUPPORT











SAILFISH OS

FURTHER INFO

Sailfish OS	https://sailfishos.org
Ubuntu Touch	https://ubports.com
postmarketOS	https://postmarketos.org
Mobian	https://mobian-project.org
Nemo Mobile	https://nemomobile.net
Slides source	https://github.com/llewelld/reg-tech-talk-linux

SAILFISH OS

IS IT OPEN SOURCE?

Closed drivers, everything else open

1. Mobian
2. postmarketOS
3. Ubuntu Touch
4. Nemo Mobile

Sailfish OS

1. Closed drivers
2. Linux kernel open
3. Middleware, Qt open
4. User interface closed
5. Jolla apps closed
6. Android App Support closed

SAILFISH OS

EXCITING COMMUNITY PROJECTS

Lots of neat stuff...

1. Flatpack support
2. Sailfish on x86
3. AsteroidOS
4. SDK Rust support

SAILFISH OS

SUPPORTED DEVICES

Distribution	Official	Community	More info
Sailfish OS	Xperia X, XA2, 10, 10 II, 10 III, Gemini	PinePhone, Fairphone 2, Galaxy Note 4, F(x)Tec Pro, Volla, plus at least 20 others	forum.sailfishos.org/t/14081
Mobian	PinePhone, Librem 5, OnePlus 6/6T, Pocophone F1	Fairphone 4, SHIFT6mq, Mi Mix 2S	wiki.mobian.org/?id=devices
postmarketOS	PinePhone, Librem 5	Fairphone 4, OnePlus 6/6T, Galaxy A3/A5/E7/Tab, Pocophone F1, Mi Note 2, plus at least 20 more	postmarketos.org/download
Ubuntu Touch	Volla, Fairphone 2, Nexus 5, OnePlus One, PinePhone	Pixel 3a, Poco X3, Mi A2, BQ Aquaris M10, Asus Zenfone Max Pro M1, Xperia X	devices.ubuntu-touch.io
Nemo Mobile	PinePhone, Volla	–	nemomobile.net/devices

SAILFISH OS

PYTORCH LIGHTNING

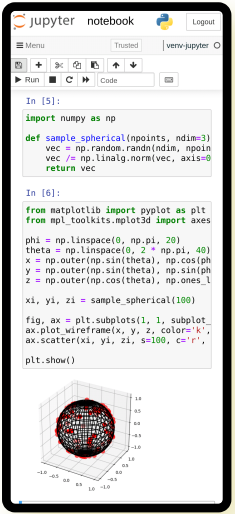
```
zypper install gcc python3-devel
python3 -m venv venv
. ./venv/bin/activate
python3 -m pip install torch lightning torchvision
python3 example.py
```

```python
# See: https://lightning.ai/docs/pytorch/stable/starter/introduction.html

import os
from torch import optim, nn, utils, Tensor
from torchvision.datasets import MNIST
from torchvision.transforms import ToTensor
import lightning.pytorch as pl

# define any number of nn.Modules (or use your current ones)
encoder = nn.Sequential(nn.Linear(28 * 28, 64), nn.ReLU(), nn.Linear(64, 3))
decoder = nn.Sequential(nn.Linear(3, 64), nn.ReLU(), nn.Linear(64, 28 * 28))

# define the LightningModule
class LitAutoEncoder(pl.LightningModule):
    def __init__(self, encoder, decoder):
        super().__init__()
        self.encoder = encoder
        self.decoder = decoder

    def training_step(self, batch, batch_idx):
        # training_step defines the train loop.
        # it is independent of forward
        x, y = batch
        x = x.view(x.size(0), -1)
        z = self.encoder(x)
        x_hat = self.decoder(z)
        loss = nn.functional.mse_loss(x_hat, x)
        # Logging to TensorBoard (if installed) by default
        self.log("train_loss", loss)
        return loss

    def configure_optimizers(self):
        optimizer = optim.Adam(self.parameters(), lr=1e-3)
        return optimizer

# init the autoencoder
autoencoder = LitAutoEncoder(encoder, decoder)

# setup data
dataset = MNIST(os.getcwd(), download=True, transform=ToTensor())
train_loader = utils.data.DataLoader(dataset, num_workers=4)

# train the model (hint: here are some helpful Trainer arguments for rapid idea iteration)
trainer = pl.Trainer(limit_train_batches=100, max_epochs=1)
trainer.fit(model=autoencoder, train_dataloaders=train_loader)
```

SAILFISH OS

JUPYTER LAB

```
python3 -m venv venv-jupyter
. ./venv-jupyter/bin/activate
pip install jupyter ipykernel matplotlib
python3 -m ipykernel install --user --name=venv-jupyter
jupyter notebook password
jupyter notebook --ip 10.0.0.42 --port 8888
```



SAILFISH OS