Brickless Battery Saving with Ubuntu Touch

...and everyone else

Hi there, I’m Dalton Durst

- Development manager, Ubuntu Touch
  - devices.ubuntu-touch.io
  - Telegram: @ubports
  - Matrix: #ubports:matrix.org
- Technology researcher, observer, explain-er

This talk

- Impossible situations users request
- Eternal Truths of Power Saving
- Fooling users into thinking we did the impossible
- Learning from those who came before
- High-level, start a discussion
- “Why is Ubuntu Touch so weird?”

People want impossible things

- Tasks run until stopped
  - Email
  - Telegram
  - Smartwatch
- “Constant Connectivity”
- Long battery life

Physics

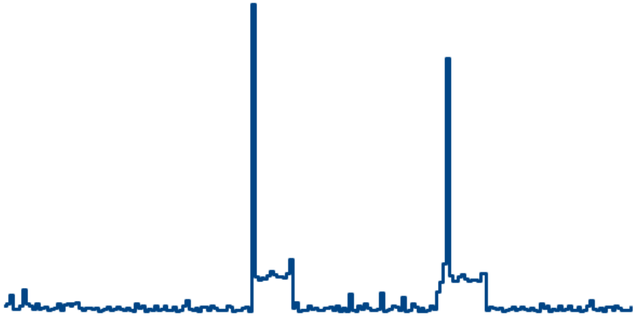
- Can’t do things forever
  - Running tasks takes energy
  - Devices have very limited energy
- Compromise
  - Old: Run the tasks you want, or don’t
  - New: Somehow do the impossible?

Don’t Compromise

- Opportunistic Idle
- XPS 13 9370 screen-off battery life:
  - 65h idle
  - 26h w/ my required apps
  - 9 hours w/ all of those and cnn.com

Eternal Power Management Truths

- Idle = Good
- “Idle” means “Idle”
- “Idle → Running” takes more energy than “Running → Busy”
  - Idling less for longer better than idling more for shorter.



Illusion of Constant Connectivity

“I want to be up-to-date” != “Always-on daemons”

Compromise: the other way

- Apps don’t know when it’s safe to process
- Suspend apps when not in use
  - Focused = Not Paused = I can process
  - Not Focused = Paused = No worries?
- Makes aggressive power management possible

“I wanted REAL Linux on my phone!”

Walk back restrictions

- Trade battery life for developer convenience
- Walk back in phases
  - Required Background Services
  - Push Services
  - Background Extensions
  - Poll Services
  - Maintenance tasks
  - Daemons
- Implementing in this order is advised

Required Background Services

- Targeted, most common tasks
  - Media Playback
  - HTTP Download and Upload
  - Location updates
  - Alarm clocks

Push services

- Flip the script
  - App → Server
  - Server → App

Background Tasks

- Actually executing app code in the background
- Split common “Background work” into categories
  - Now
  - Sometime later
  - Exact time later
- More flexible options give worse battery life

Background Tasks: Now

- Limited uses
  - Compiling software
  - Rendering 3D models
  - Transcoding video
- iOS, Android give you a few minutes
- Notify the user if you run out of time

Background Tasks: Sometime Later

- Good scheduler gives good battery life
- Inflexible
- Giving your task to the God King System to schedule
  - Minimize wakeups
  - Safe to use data
  - User isn’t present
- iOS Background Tasks; Android WorkManager constrained tasks

Background Tasks: Exact Time Later

- Very flexible; Very poor battery life
- iOS: only if you’re Apple
- Android
  - WorkManager Scheduled Tasks (approximate)
  - AlarmManager (exact)

All else has failed: Daemons

- Launch issues
  - Notifying apps of file changes
- Decentralized, latency-sensitive notifications
  - VoIP
- Device expecting incoming connections
  - If your phone is a server, it gets server battery life...

Recap

- People want their computers to run their tasks forever
- People want good battery life

Recap

- Eternal Power Management Truths
  - “Idle” is good
  - “Idle” means “Idle”
  - “Idle → Running” uses more power than “Running → Busy”

Recap

- New Compromise: Illusion of Constant Connectivity
  - Suspend all apps
  - Give developers the tools they need to work in the background

Let’s do this together!

- brickless@daltondur.st
- daltondur.st/las2020
- forums.ubports.com @UniSuperBox
- Twitter: @UnivrslSuprBox