Don’t let your motivation go, save time with kworkflow

Melissa Wen

kernel GPU driver developer @ Igalia

Spending your life compiling the kernel

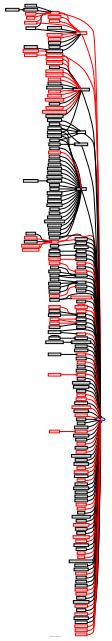
So, you are a kernel developer…
… or wanna be a kernel developer…
… or don’t wanna be a kernel developer…

United by a single need: Validate a custom kernel with a given change.

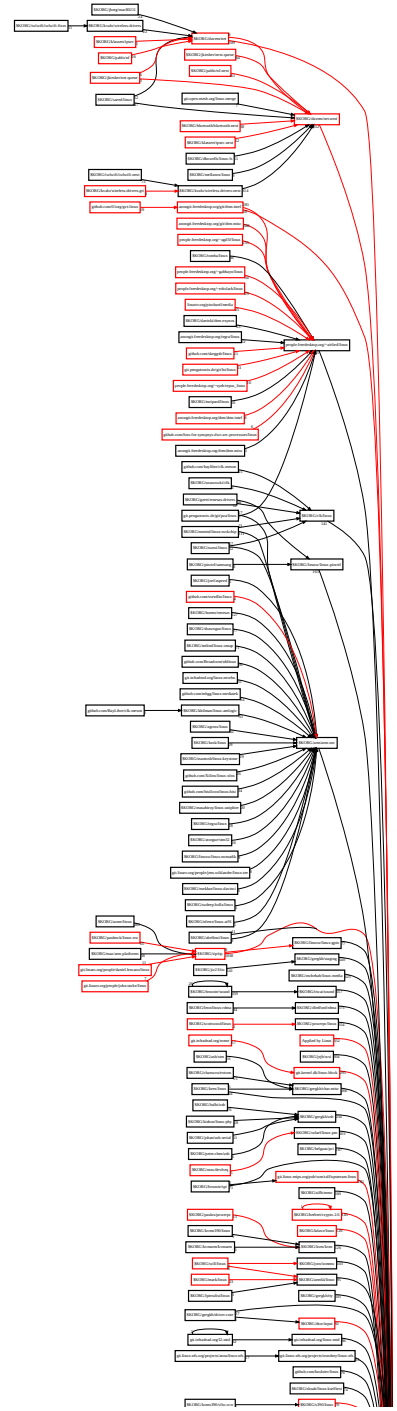
For a given distro:

For a given device:

For a given subsystem:



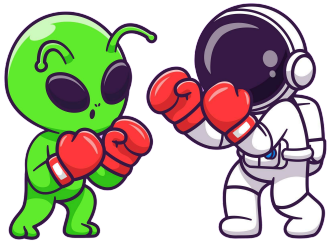
igalia



igalia

Being or not being a kernel developer

Issue tracker: reporter vs kernel developer



<Source: freepik/catalyststuff>

igalia

Once upon a time there was a userspace developer who reported a kernel issue...

- *Reporter:* “There is an issue in your driver only reproducible when running this distribution.”
  - *Kernel developer:* “Can you check if this issue still happens using this kernel branch?”
  - *Reporter* **never compiled and installed a custom kernel** before. Read many kernel tutorials to **create a build & deploy kernel script.**
- *Reporter:* “Sorry for delaying, it’s my first time deploying a custom kernel. I’m not sure I did it right, but the issue is still present in this kernel branch”
  - *Kernel developer* needs to reproduce the issue on their side, but **never used this distribution** so also **created almost the same script created by the reporter.**

The Problem: You keep creating new scripts!

- Every time you:
  - Change distro
  - Change architecture
  - Change hardware
  - Change project
- You create **another** script for your new kernel development workflow!

igalia

My precious script

igalia

The Repetitive Developer Cycle

- Instead of creating and accumulating scripts, **save your life time** with kworkflow!

```
# Your script

$ make ARCH=$ARCH_64 CROSS_COMPILE=$CROSS_COMPILE_64 $DEFCONFIG
$ make -j$(nproc) ARCH=$ARCH_64 CROSS_COMPILE=$CROSS_COMPILE_64
$ make ARCH=$ARCH_64 CROSS_COMPILE=$CROSS_COMPILE_64 INSTALL_MOD_PATH=$TMP modules_install
$ ssh $RPI4 mkdir -p /tmp/new_modules /tmp/new_kernel /tmp/new_kernel/overlays
$ rsync -av $TMP/ $RPI4:/tmp/new_modules/
$ scp arch/$ARCH_64/boot/Image $RPI4:/tmp/new_kernel/Image-$KERNEL.img
$ scp arch/$ARCH_64/boot/$DTB_PATH/*.dtb $RPI4:/tmp/new_kernel
$ scp arch/$ARCH_64/boot/dts/overlays/*.dtb* $RPI4:/tmp/new_kernel/overlays
$ ssh $RPI4 sudo rsync -av /tmp/new_modules/lib/modules/ /lib/modules/
$ ssh $RPI4 sudo rsync -av /tmp/new_kernel/ /boot/
$ rm -rf $TMP
```

```
# With kworkflow

$ kw k --fetch --remote root@localhost:2222
$ kw bd
```

igalia

What is kworkflow? (kw)

- A collection of tools and software combined to:
  - Optimizes Linux kernel development workflow.
  - Reduces time spent on repetitive tasks.
  - Standardizes best practices.
  - Ensures reliable data exchange across kernel workflows.

igalia

What is kworkflow? (kw)

<Source: **powerrangers.fandom.com**>

igalia

Key Features of kworkflow

- Build & deploy custom kernels **across devices & distros**.
- Handle **cross-compilation seamlessly**.
- Manage **multiple architecture, settings and target devices** in the same work tree.
- Organize **kernel configuration files**.
- Facilitate **remote debugging & code inspection**.
- **Standardize Linux kernel patch submission guidelines**.
- **Upcoming:** Interface to bookmark, apply and “reviewed-by” patches from mailing lists (**lore.kernel.org**).

igalia

kworkflow Command Overview

```
# Manage kw and kw configurations
kw init             - Initialize kw config file
kw self-update (u)  - Update kw
kw config (g)       - Manage kernel .config files
[...]

# Build & Deploy custom kernels
kw kernel-config-manager (k) - Manage kernel .config files
kw build (b)        - Build kernel
kw deploy (d)       - Deploy kernel image (local/remote)
kw bd               - Build and deploy kernel

# Manage and interact with target machines
kw ssh (s)          - SSH support
kw remote (r)       - Manage machines available via ssh
kw vm               - QEMU support

# Inspect and debug
kw device           - Show basic hardware information
kw explore (e)      - Explore string patterns in the work tree and git logs
kw debug            - Linux kernel debug utilities
kw drm              - Set of commands to work with DRM drivers

# Automatize best practices for patch submission
kw codestyle (c)    - Check code style
kw maintainers (m)  - Get maintainers/mailing list
kw send-patch       - Send patches via email

# Upcoming
kw patch-hub        - Interact with patches (lore.kernel.org)
```

igalia

Save time on: building and deploying custom kernels

- **Before:**
  - Manually extract and manage .config files from different targets
  - Sometimes renaming .config files with some descriptive suffix
  - copy&paste (ofc)
- **After: kw kernel-config-manager (k)** - Fetch and manage kernel `.config` files easily.

```
# Extract and copy .config file from a given device
kw k --fetch (--remote root@localhost:2222 | --local)

# Store and manage .config files
kw k --save <my_current_config_name> --description <perfect_for_device1>
kw k --list
kw k --get <my_current_config_name>
```

igalia

Save time on: building and deploying custom kernels

- **Before:** Memorize combinations of make commands and options
- **After: kw build (b)** - Build the kernel

```
kw b                # Build kernel with settings for cross-compilation, cflags, llvm, ccache, cpu scaling factor
kw b --menu         # Open menuconfig
kw b -i             # Show name/version and module count
kw b -w             # Enable compilation warnings
```


igalia

Save time on: building and deploying custom kernels

- **Before:**
  - SSH
  - Copy or remove files according to distro and architecture
  - Manually update bootloader by distro.
- **After: kw deploy (d)** - Deploy the custom kernel in a target machine

```
kw d                # Deploy kernel
kw d --setup        # Prepare target machine
kw d --list         # List available kernels
kw d --uninstall    # Remove kernel(s)
kw d --create-package # Create sharable kw package
kw d --reboot       # Reboot after deploy

kw bd               # Build and deploy kernel
```

Save time on: debugging kernels locally or remotely

- **Before:**
  - SSH
  - Manually setup and enable traces
  - Copy&Paste (again)
- **After: kw debug** - simplify kernel debug utilities: events, ftrace, dmesg.

```
kw debug            # Debug utilities
kw debug -c "cmd"   # Trace log for a command
kw debug -k         # Store trace logs
kw debug -f         # Follow traces in real-time
```

- Supports local & remote debugging.

Save time on: managing multiple kernel images in the same work tree

- **Before:**
  - Clone multiple times the same branch(?)
  - Lose compiled files when changing kernel config or compilation options
  - Manually manage deployment scripts.
- **After: kw env** - isolating multiple contexts in the same work tree as environments

```
kw config          # Manage kernel environment settings
kw env             # Create isolated kernel environments
```

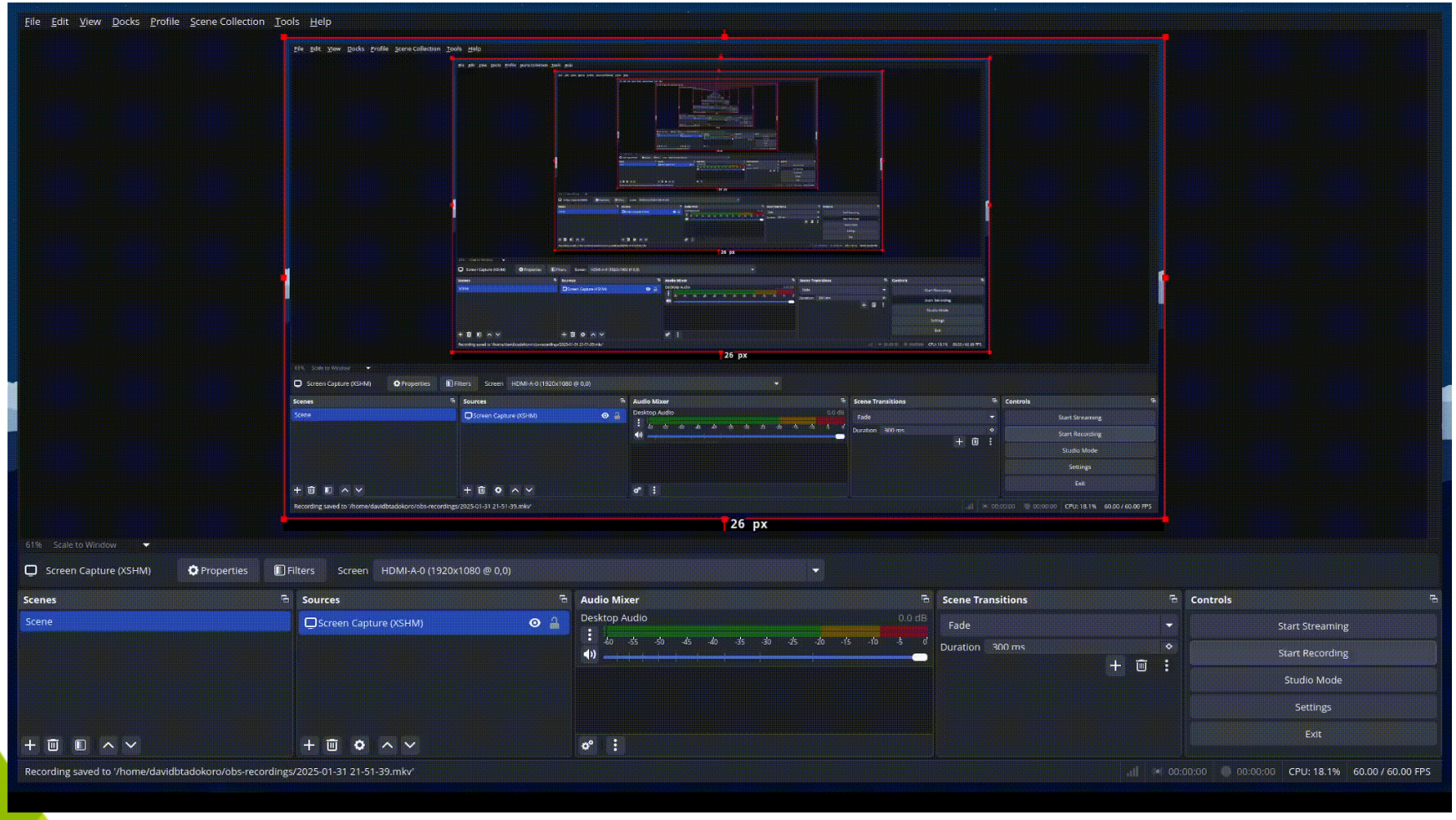
- Maintain different setups in the **same work tree**.

Save time on: submitting patches to mailing lists

```
kw code-style      # Check kernel coding style
kw maintainer      # Get maintainers & mailing lists
kw send-patch      # Automatically create the list of recipients and send patches via email
kw patch-hub       # Terminal UI for interacting with patches in lore mailing lists
```

- **Automate patch submission rules on sending patches**, via `git send-email`.
- Ensure patches **reach the right people (maintainers and ML)**.

igalia


File Edit View Docks Profile Scene Collection Tools Help
26 px
Screen Capture (XSHM)
Properties
Filters
Screen
HDMI-A-0 (1920x1080 @ 0,0)
Scenes
Scene
Sources
Screen Capture (XSHM)
Audio Mixer
Desktop Audio
0.0 dB
Scene Transitions
Fade
Duration
300 ms
Controls
Start Streaming
Start Recording
Studio Mode
Settings
Exit
Recording saved to '/home/davidbtadokoro/obs-recordings/2025-01-31 21-51-39.mkv'
00:00:00
00:00:00
CPU: 18.1%
60.00 / 60.00 FPS

Call to Action

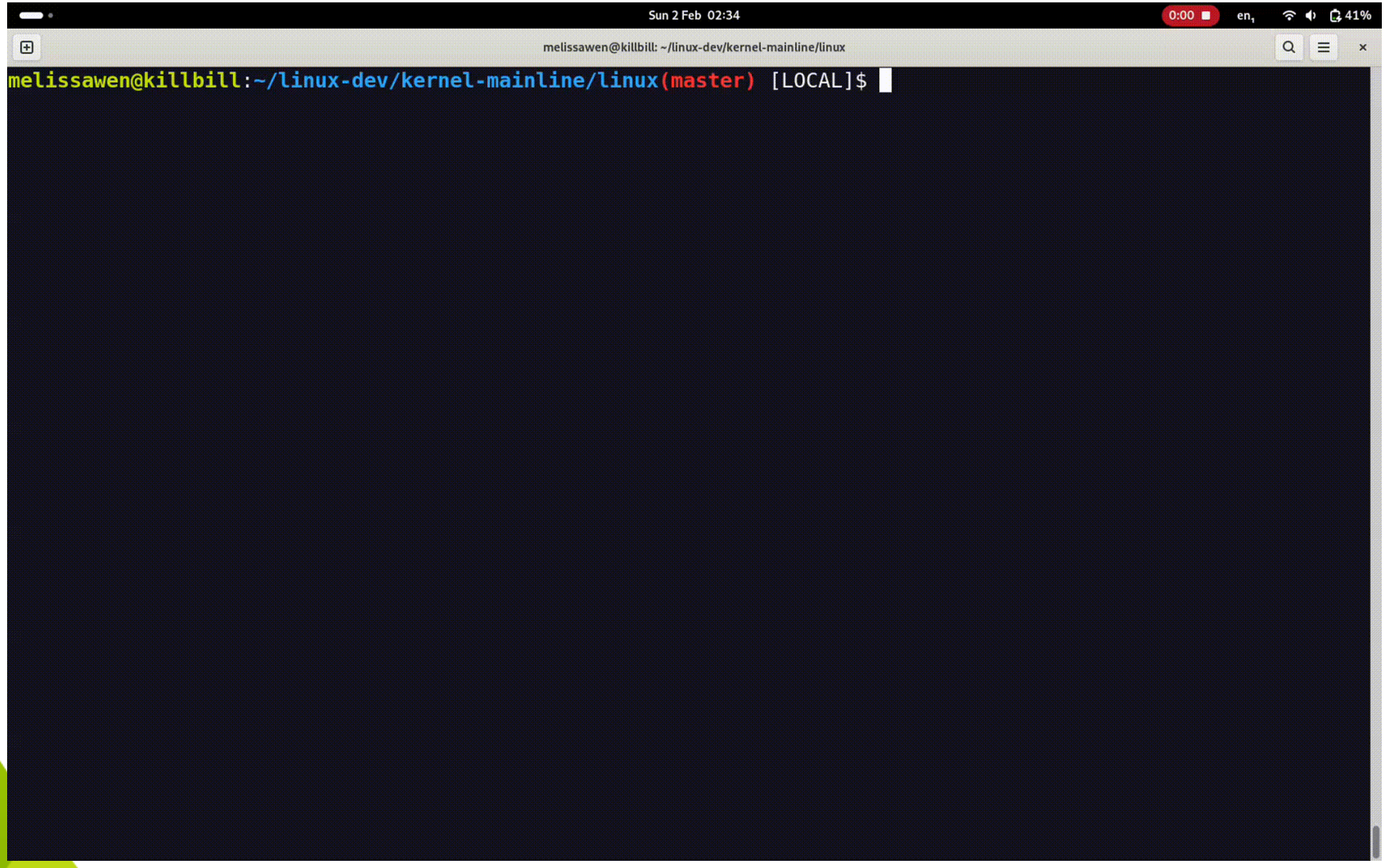
- Stop writing redundant scripts!
- Save everybody's time and effort with kworkflow.
- **Try kworkflow today: https://kworkflow.org**
- **Contribute to kworkflow: it's 100% volunteering work.**
- **Challenge:**
  - Replace one of your scripts with kworkflow this week!
  - Make a new kworkflow feature with one of your scripts.

igalia

Demo setup:

- Setup: Three devices:
  - laptop (debian|x86|intel|local)
  - SteamDeck (steamos|x86|amd|remote)
  - RaspberryPi 4 (raspbian|arm64|broadcomm|remote)
- Goal: To validate a change on DRM/VKMS using a single kernel tree.
- Kworkflow commands:
  - kw env
  - kw d
  - kw bd
  - kw device
  - kw debug
  - kw drm

igalia


Sun 2 Feb 02:34
0:00
en,
41%
melissawen@killbill: ~/linux-dev/kernel-mainline/linux
melissawen@killbill:~/linux-dev/kernel-mainline/linux(master) [LOCAL]$

igalia