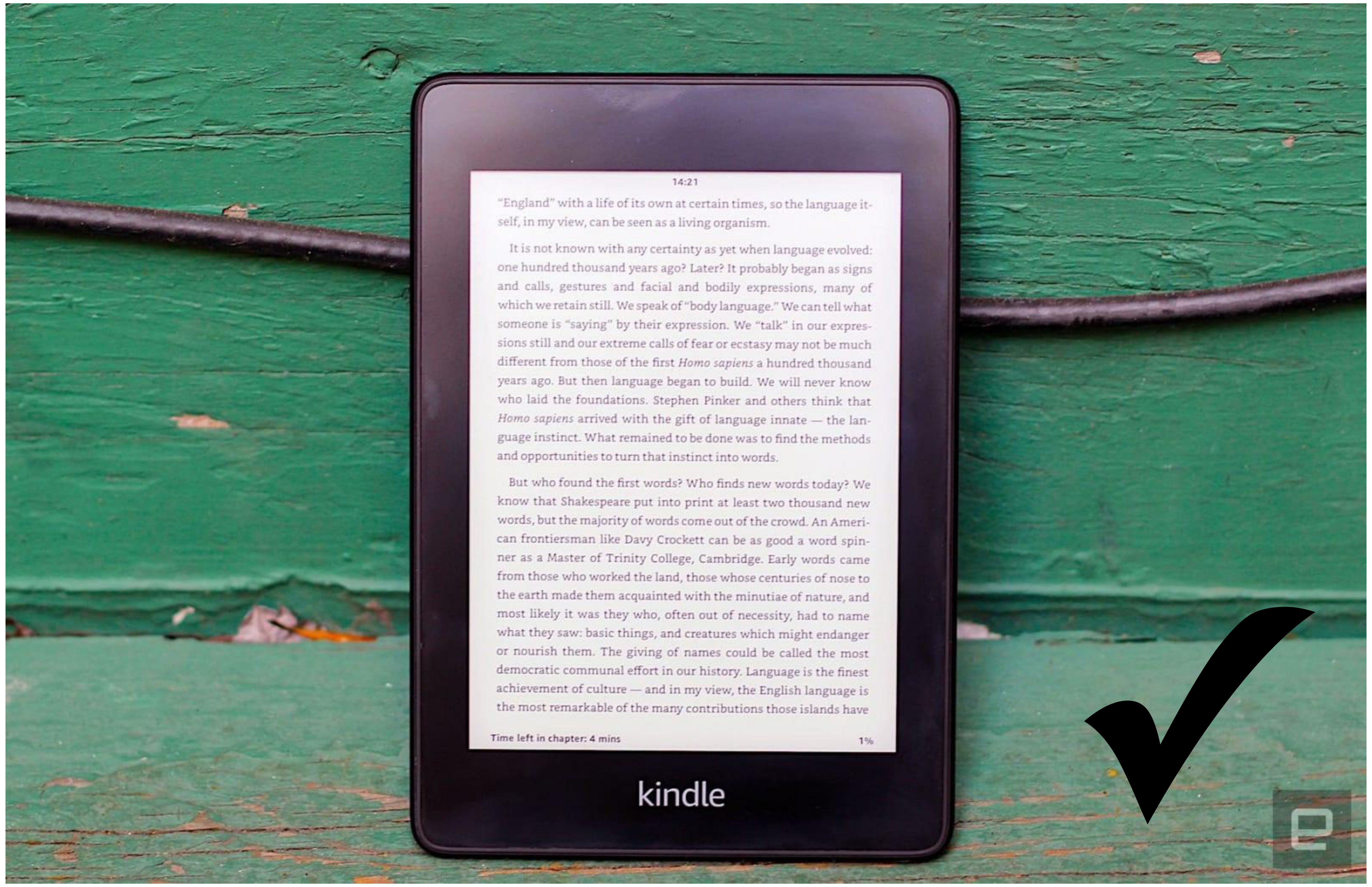
Kindle, Licenses, and Jailbreaking

Owen Hiland

Kindle Fire:

- Kernel Based on AOSP (modified Linux kernel — effectively Android)
- Install applications from Amazon App Store
- Basically just reskinned Android. Can install other OSes, if you're willing to void your warranty (or just sideload Android apps)


14:21

"England" with a life of its own at certain times, so the language itself, in my view, can be seen as a living organism.

It is not known with any certainty as yet when language evolved: one hundred thousand years ago? Later? It probably began as signs and calls, gestures and facial and bodily expressions, many of which we retain still. We speak of "body language." We can tell what someone is "saying" by their expression. We "talk" in our expressions still and our extreme calls of fear or ecstasy may not be much different from those of the first *Homo sapiens* a hundred thousand years ago. But then language began to build. We will never know who laid the foundations. Stephen Pinker and others think that *Homo sapiens* arrived with the gift of language innate — the language instinct. What remained to be done was to find the methods and opportunities to turn that instinct into words.

But who found the first words? Who finds new words today? We know that Shakespeare put into print at least two thousand new words, but the majority of words come out of the crowd. An American frontiersman like Davy Crockett can be as good a word spinner as a Master of Trinity College, Cambridge. Early words came from those who worked the land, those whose centuries of nose to the earth made them acquainted with the minutiae of nature, and most likely it was they who, often out of necessity, had to name what they saw: basic things, and creatures which might endanger or nourish them. The giving of names could be called the most democratic communal effort in our history. Language is the finest achievement of culture — and in my view, the English language is the most remarkable of the many contributions those islands have

Time left in chapter: 4 mins 1%

E-ink Kindle:

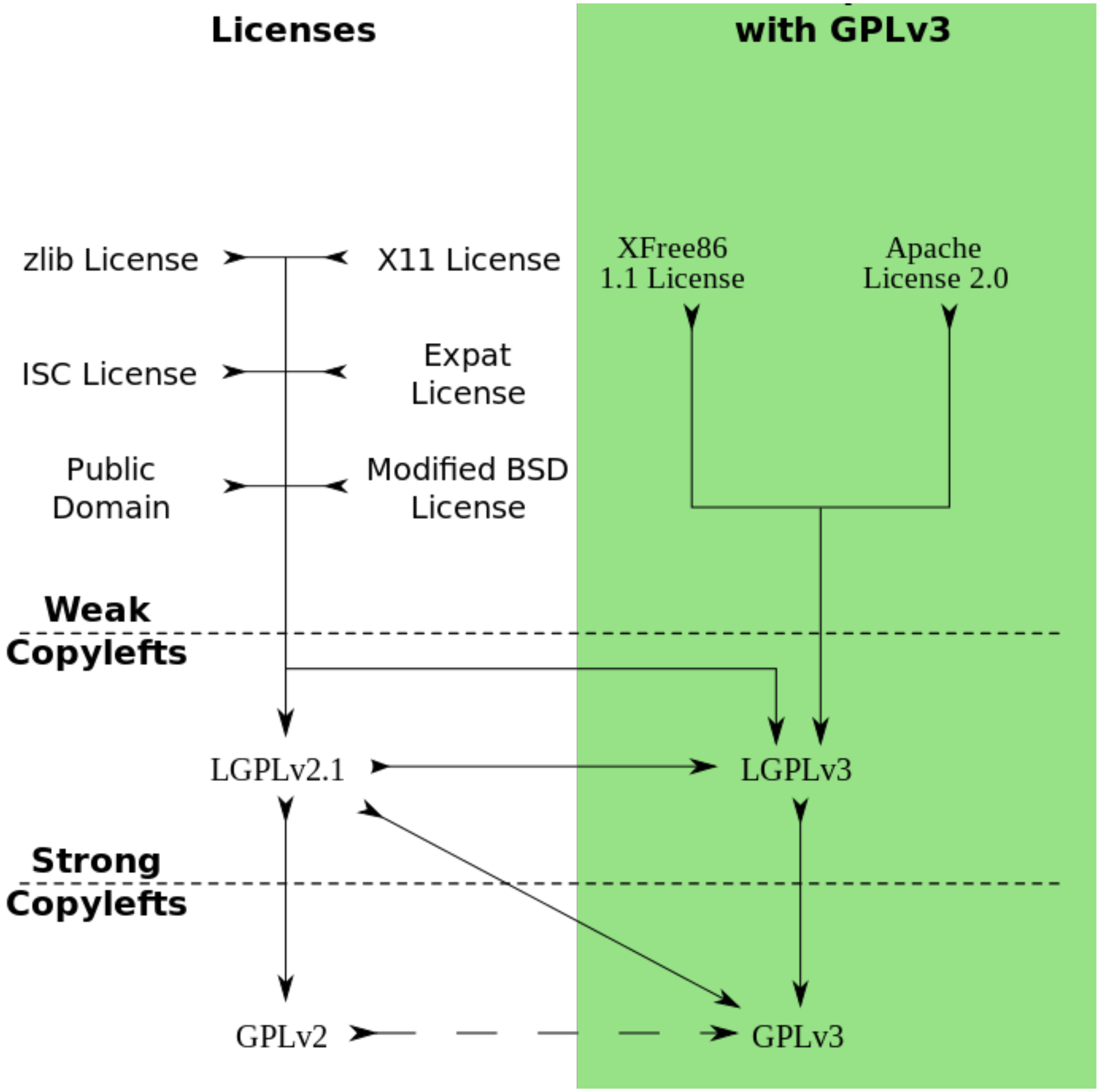
- Based on Linux kernels as well
- Limited by small form factor and low screen refresh rates.
- Connection to Amazon e-book store, but no support for installing applications since Amazon closed its Active Content Store

The Kindle’s Kernel (and the Surrounding OS)

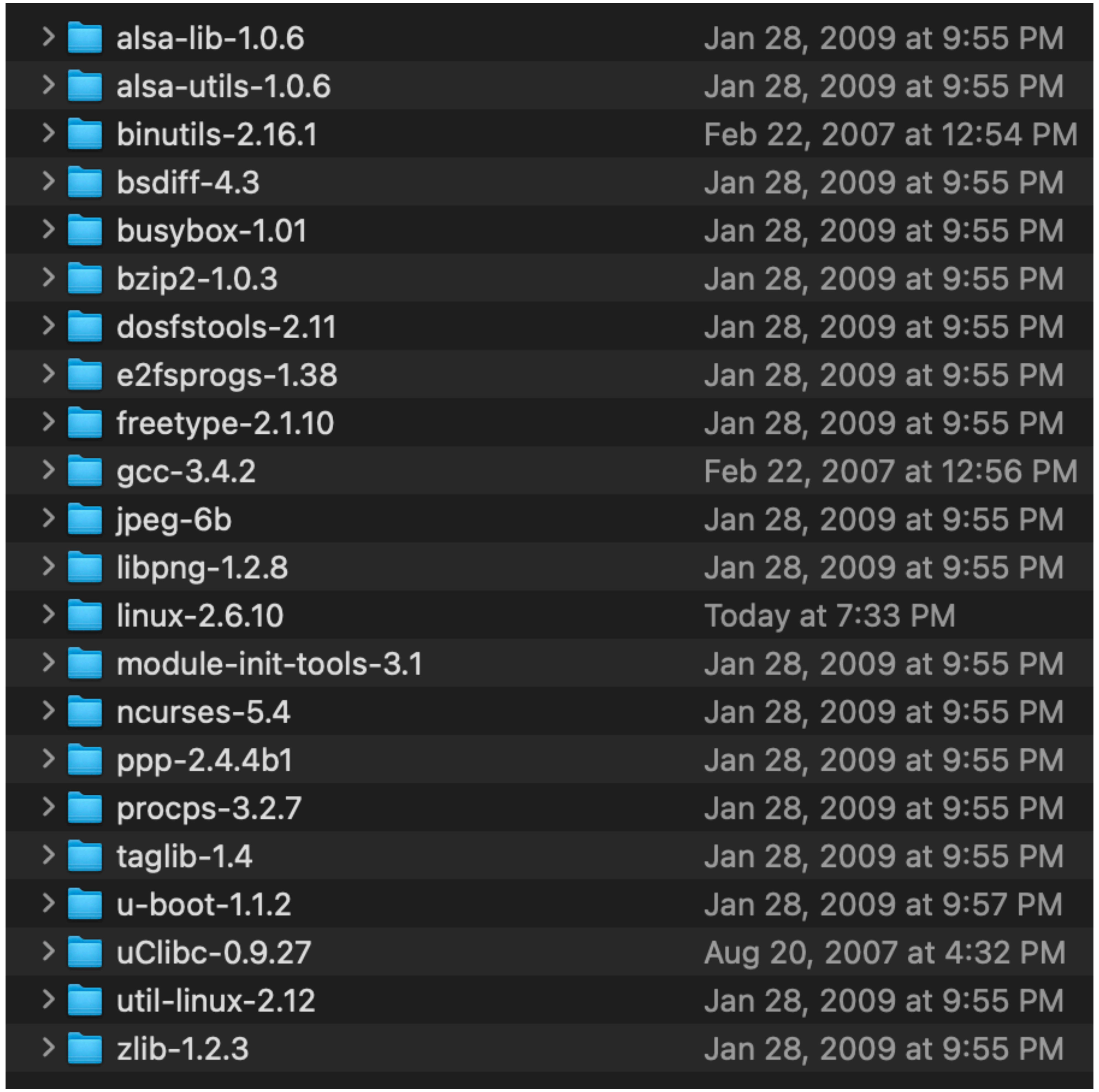
- The Kindle’s code and Amazon’s modifications to it are available because of the Linux Kernel’s use of the GNU General Public License v2, which specifies that derivative products make modifications to the licensed content used available publicly

- Companies have been taken to court over failure to make source code used under GPL and modifications made available publicly (i.e. Cisco in 2009 v.s. the FSF)

- All Kindle source code (modified from GPL sources) since the original Kindle in 2007 are available to download.

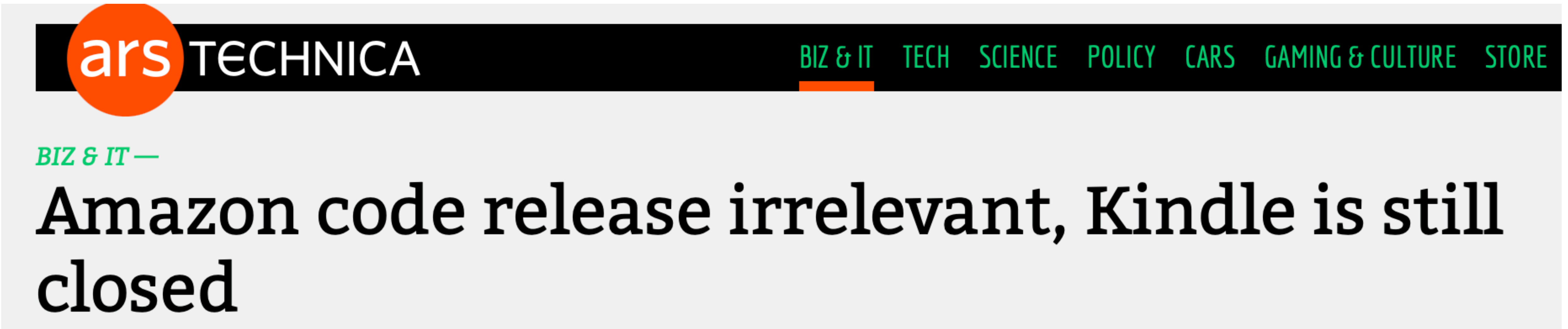


Kindle Source Code (For K1)



alsa-lib-1.0.6	Jan 28, 2009 at 9:55 PM
alsa-utils-1.0.6	Jan 28, 2009 at 9:55 PM
binutils-2.16.1	Feb 22, 2007 at 12:54 PM
bsdiff-4.3	Jan 28, 2009 at 9:55 PM
busybox-1.01	Jan 28, 2009 at 9:55 PM
bzip2-1.0.3	Jan 28, 2009 at 9:55 PM
dosfstools-2.11	Jan 28, 2009 at 9:55 PM
e2fsprogs-1.38	Jan 28, 2009 at 9:55 PM
freetype-2.1.10	Jan 28, 2009 at 9:55 PM
gcc-3.4.2	Feb 22, 2007 at 12:56 PM
jpeg-6b	Jan 28, 2009 at 9:55 PM
libpng-1.2.8	Jan 28, 2009 at 9:55 PM
linux-2.6.10	Today at 7:33 PM
module-init-tools-3.1	Jan 28, 2009 at 9:55 PM
ncurses-5.4	Jan 28, 2009 at 9:55 PM
ppp-2.4.4b1	Jan 28, 2009 at 9:55 PM
procps-3.2.7	Jan 28, 2009 at 9:55 PM
taglib-1.4	Jan 28, 2009 at 9:55 PM
u-boot-1.1.2	Jan 28, 2009 at 9:57 PM
uClibc-0.9.27	Aug 20, 2007 at 4:32 PM
util-linux-2.12	Jan 28, 2009 at 9:55 PM
zlib-1.2.3	Jan 28, 2009 at 9:55 PM

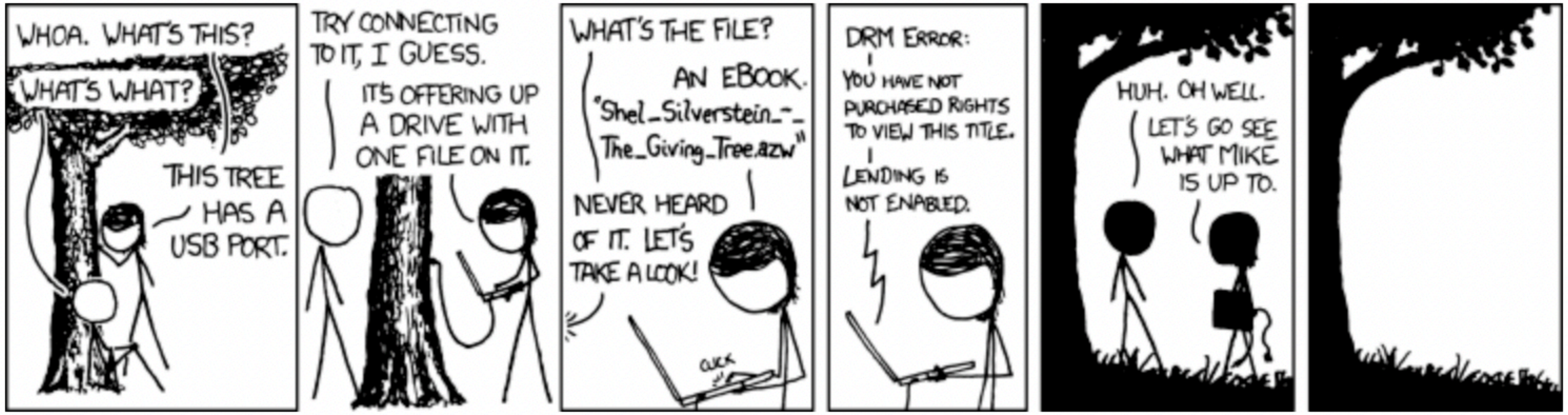
Is the Kindle a “Closed System”?

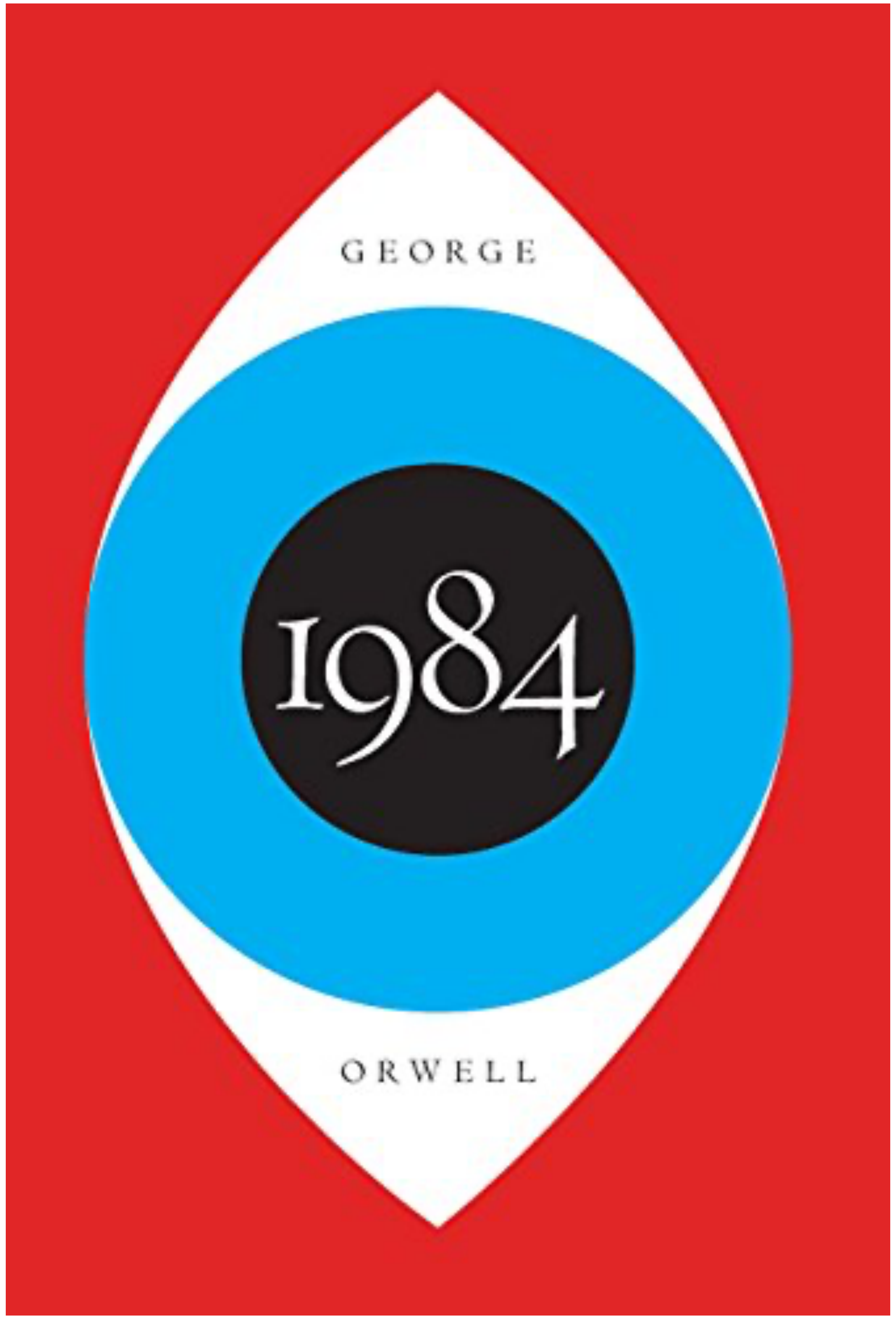


BIZ & IT —

Amazon code release irrelevant, Kindle is still closed

“Tragically, a growing number of hardware makers are using code signing and other technical measures to prevent users from swapping out the software that comes with a device. The Kindle is sadly one such device. Even though the source code of the Kindle's kernel is available, it's useless as far as Kindle enhancement is concerned because Amazon blocks installation of custom versions on the product. This means that the Kindle is a fundamentally closed device despite its use of some open source software.“


WHOA. WHAT'S THIS?
WHAT'S WHAT?
THIS TREE HAS A USB PORT.
TRY CONNECTING TO IT, I GUESS.
IT'S OFFERING UP A DRIVE WITH ONE FILE ON IT.
WHAT'S THE FILE?
AN EBOOK. "Shel_Silverstein_-_The_Giving_Tree.azw"
NEVER HEARD OF IT. LET'S TAKE A LOOK!
CLICK
DRM ERROR: YOU HAVE NOT PURCHASED RIGHTS TO VIEW THIS TITLE. LENDING IS NOT ENABLED.
HUH. OH WELL.
LET'S GO SEE WHAT MIKE IS UP TO.



Jailbreaking the Kindle

- We could certainly install our own applications, right? At one point, the e-ink Kindles had an “Active content” section — what if we could somehow reinstate it and run our own applications?

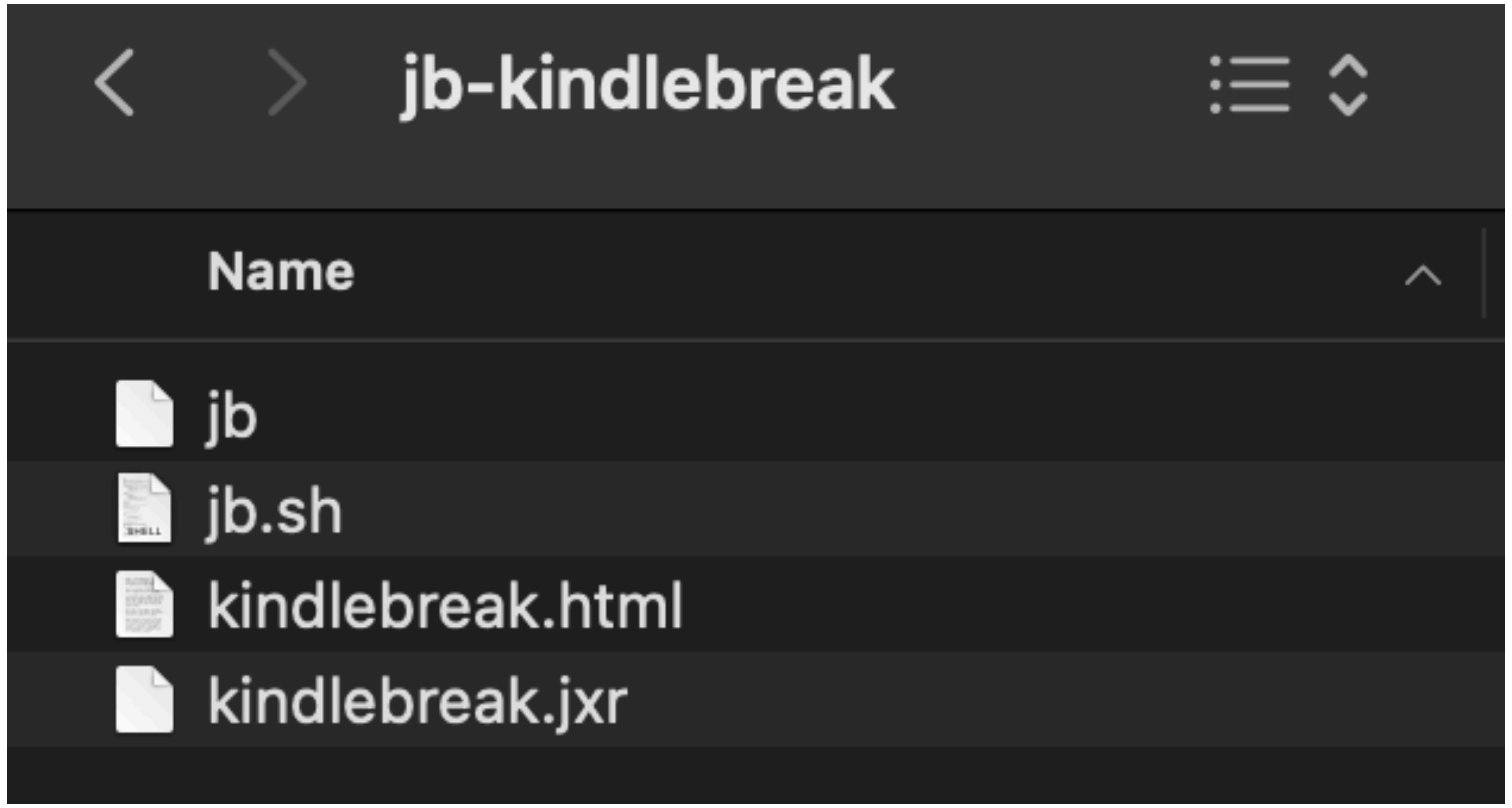
- This process is called “Jailbreaking” — effectively, it allows for the installation of applications from sources other than the “official channels” — which, for e-ink Kindles, are all officially closed.

- This requires some way of getting new software onto the Kindle, and also adding support to let the Kindle open and run applications and to display them to the user.

Performing a jailbreak

- Achieved by a few methods:
    - Imitating an update from Amazon — a .bin or similar file is placed in the root directory, where software updates go after they’ve been downloaded on the Kindle OS. The installing user updates the Kindle, and this dummy update is installed.
    - Typically, the “update” registers a developer key which specifies to the system and associates it with a user who is then granted root access to the machine. Software added to the device will be signed by this key.
    - The user must be careful to prevent future updates from Amazon from automatically downloading, which can be done by installing another bin file through a similar process.
    - The jailbreaking user can now install an application launcher such as KUAL, which will subsequently allow the user to run others’ code or their own by the use of “buttons”.

Jailbreak example code: KindleBreak



jb-kindlebreak

Name

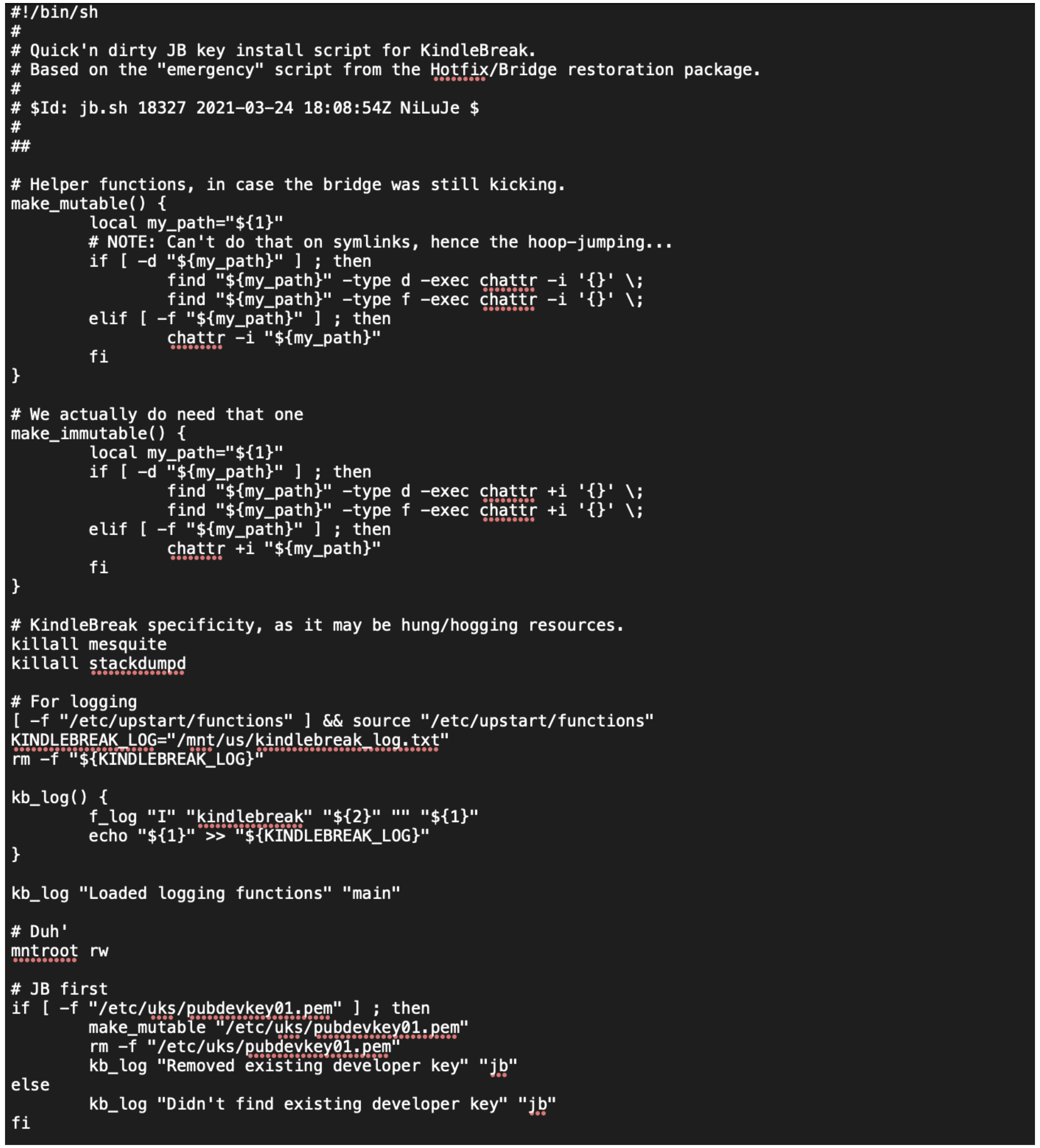
jb

jb.sh

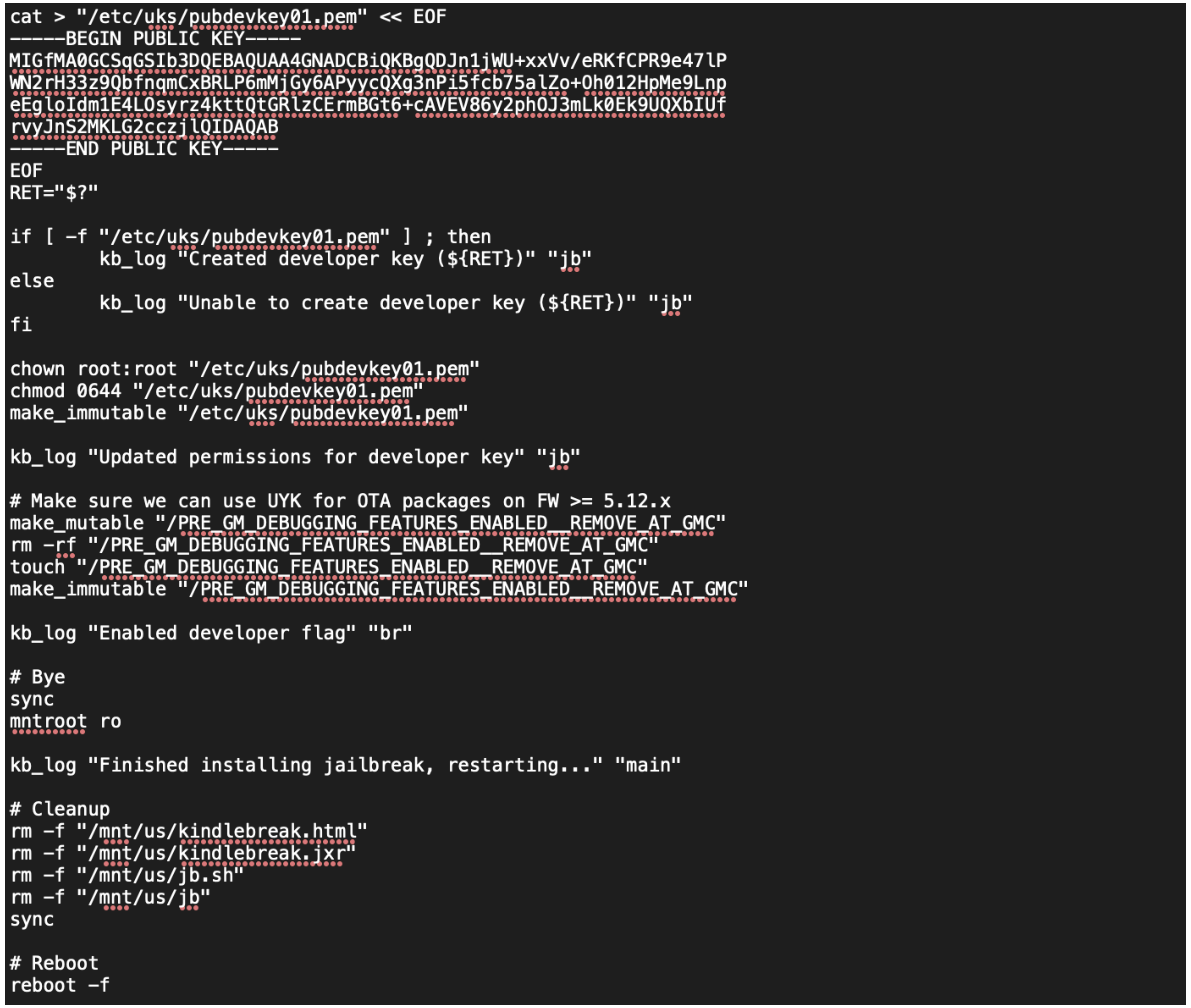
kindlebreak.html

kindlebreak.jxr

Jailbreak example code: KindleBreak



```
#!/bin/sh
#
# Quick'n dirty JB key install script for KindleBreak.
# Based on the "emergency" script from the Hotfix/Bridge restoration package.
#
# $Id: jb.sh 18327 2021-03-24 18:08:54Z NiLuJe $
#
##

# Helper functions, in case the bridge was still kicking.
make_mutable() {
        local my_path="${1}"
        # NOTE: Can't do that on symlinks, hence the hoop-jumping...
        if [ -d "${my_path}" ] ; then
                find "${my_path}" -type d -exec chattr -i '{}' \;
                find "${my_path}" -type f -exec chattr -i '{}' \;
        elif [ -f "${my_path}" ] ; then
                chattr -i "${my_path}"
        fi
}

# We actually do need that one
make_immutable() {
        local my_path="${1}"
        if [ -d "${my_path}" ] ; then
                find "${my_path}" -type d -exec chattr +i '{}' \;
                find "${my_path}" -type f -exec chattr +i '{}' \;
        elif [ -f "${my_path}" ] ; then
                chattr +i "${my_path}"
        fi
}

# KindleBreak specificity, as it may be hung/hogging resources.
killall mesquite
killall stackdumpd

# For logging
[ -f "/etc/upstart/functions" ] && source "/etc/upstart/functions"
KINDLEBREAK_LOG="/mnt/us/kindlebreak_log.txt"
rm -f "${KINDLEBREAK_LOG}"

kb_log() {
        f_log "I" "kindlebreak" "${2}" "" "${1}"
        echo "${1}" >> "${KINDLEBREAK_LOG}"
}

kb_log "Loaded logging functions" "main"

# Duh'
mntroot rw

# JB first
if [ -f "/etc/uks/pubdevkey01.pem" ] ; then
        make_mutable "/etc/uks/pubdevkey01.pem"
        rm -f "/etc/uks/pubdevkey01.pem"
        kb_log "Removed existing developer key" "jb"
else
        kb_log "Didn't find existing developer key" "jb"
fi
```



```
cat > "/etc/uks/pubdevkey01.pem" << EOF
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDJn1jWU+xxVv/eRKfCPR9e47lP
WN2rH33z9QbfnqmCxBRLP6mMjGy6APyycQXg3nPi5fcb75alZo+Oh012HpMe9Lnp
eEgloIdm1E4LOsyrz4kttQtGRlzCErmBGt6+cAVEV86y2phOJ3mLk0Ek9UQXbIUf
rvyJnS2MKLG2cczjlQIDAQAB
-----END PUBLIC KEY-----
EOF
RET="$?"

if [ -f "/etc/uks/pubdevkey01.pem" ] ; then
        kb_log "Created developer key (${RET})" "jb"
else
        kb_log "Unable to create developer key (${RET})" "jb"
fi

chown root:root "/etc/uks/pubdevkey01.pem"
chmod 0644 "/etc/uks/pubdevkey01.pem"
make_immutable "/etc/uks/pubdevkey01.pem"

kb_log "Updated permissions for developer key" "jb"

# Make sure we can use UYK for OTA packages on FW >= 5.12.x
make_mutable "/PRE_GM_DEBUGGING_FEATURES_ENABLED__REMOVE_AT_GMC"
rm -rf "/PRE_GM_DEBUGGING_FEATURES_ENABLED__REMOVE_AT_GMC"
touch "/PRE_GM_DEBUGGING_FEATURES_ENABLED__REMOVE_AT_GMC"
make_immutable "/PRE_GM_DEBUGGING_FEATURES_ENABLED__REMOVE_AT_GMC"

kb_log "Enabled developer flag" "br"

# Bye
sync
mntroot ro

kb_log "Finished installing jailbreak, restarting..." "main"

# Cleanup
rm -f "/mnt/us/kindlebreak.html"
rm -f "/mnt/us/kindlebreak.jxr"
rm -f "/mnt/us/jb.sh"
rm -f "/mnt/us/jb"
sync

# Reboot
reboot -f
```

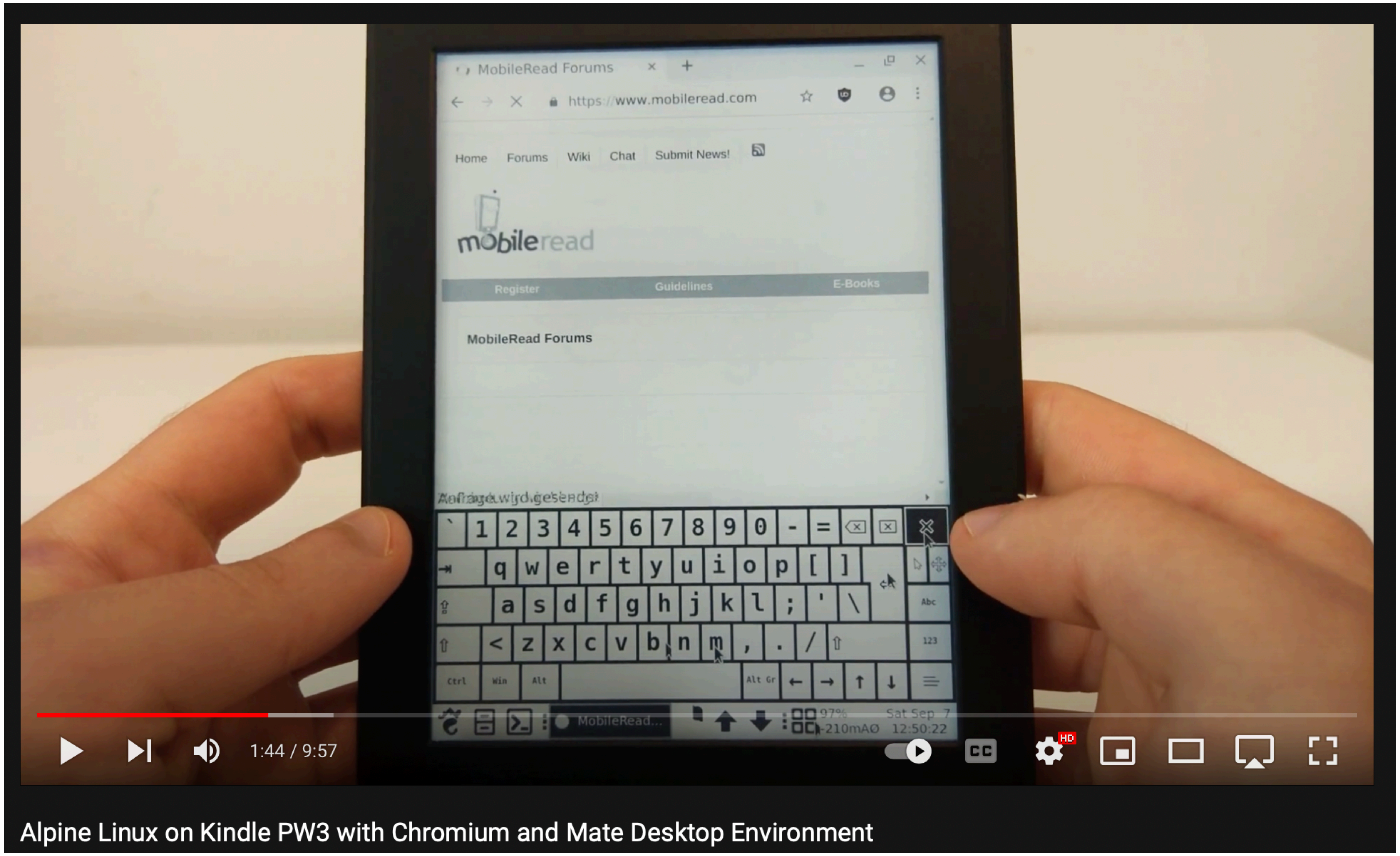
The KUAL Launcher

- Because the Kindle O/S is not replaced by this process — it only adds a user — we are still constrained by the interface offered by the kindle; also, writing a new OS would be difficult because some kindle components use proprietary drivers or are otherwise not open-source.

- Thus, a happy middle ground between writing an OS from scratch and doing nothing at all would be something like KUAL; KUAL is effectively an app for launching and displaying other apps. KUAL is written in the AZW2 format, which is the format for “Active Content” that Amazon once supported.

- The app makes uses of buttons to trigger shell scripts, which can subsequently run other programs available to the user.

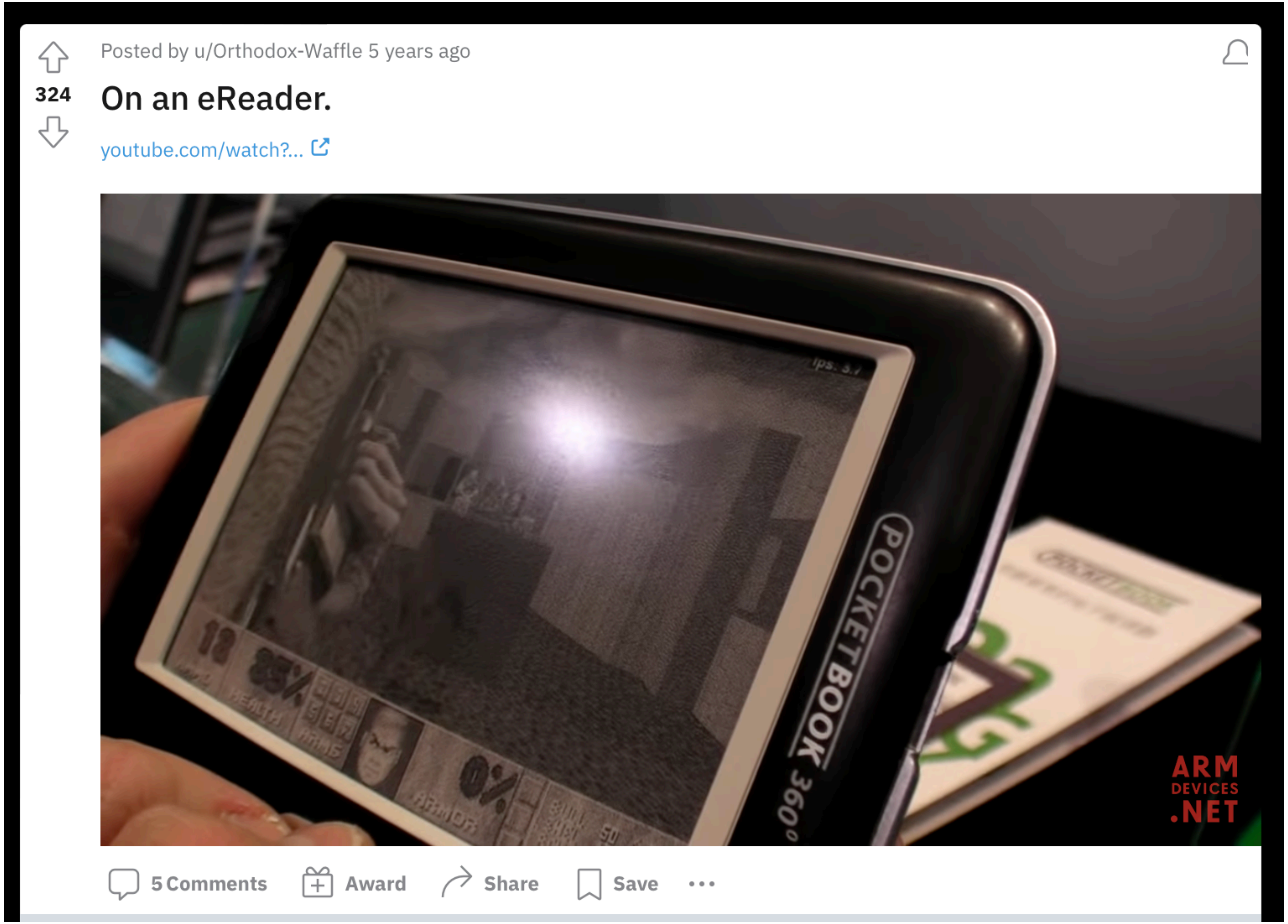
Can It Run Doom?



Alpine Linux on Kindle PW3 with Chromium and Mate Desktop Environment

Can It Run Doom?

No, but it can run Alpine Linux!



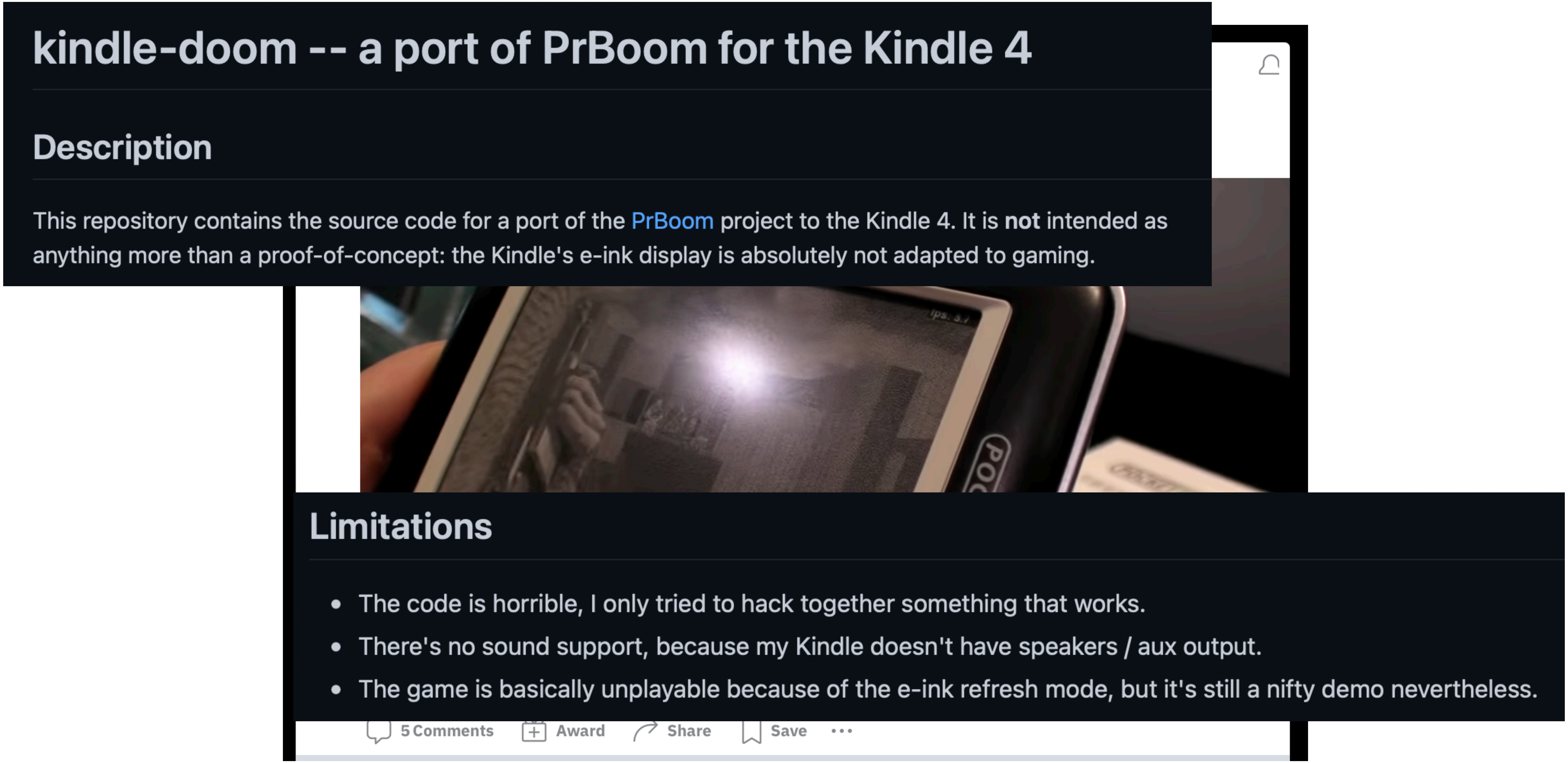
Posted by u/Orthodox-Waffle 5 years ago

324

On an eReader.

youtube.com/watch?...

5 Comments Award Share Save ...



kindle-doom -- a port of PrBoom for the Kindle 4

Description

This repository contains the source code for a port of the PrBoom project to the Kindle 4. It is **not** intended as anything more than a proof-of-concept: the Kindle's e-ink display is absolutely not adapted to gaming.

Limitations

- The code is horrible, I only tried to hack together something that works.
- There's no sound support, because my Kindle doesn't have speakers / aux output.
- The game is basically unplayable because of the e-ink refresh mode, but it's still a nifty demo nevertheless.

Final Thoughts

- Despite a lack of maker support from Amazon beyond the bare legal requirement, the Kindle ecosystem is the site of a vibrant and dedicated hacker community.

- Although the Kindle itself is not an “open” platform, it relies on a broad range of open-source projects to operate. Amazon benefits from open-source projects, but by making it difficult for users to install their own apps without relying on some degree of technical knowledge, they reduce the power of the user to use their device as they see fit.

- Although in this case Amazon tolerates the jailbreaking community, other companies — notably Apple — have taken legal action in order to classify jailbreaking as a copyright issue.

Sources

- https://arstechnica.com/information-technology/2009/06/amazon-code-release-irrelevant-kindle-is-still-closed/
- https://www.cnet.com/culture/what-will-you-do-with-amazons-kindle-source-code/
- https://www.mobileread.com/forums/showthread.php?t=203326 (KULA)
- https://www.mobileread.com/forums/showthread.php?t=338268 (KindleBreak)
- https://www.amazon.com/gp/help/customer/display.html?nodeId=200203720 (Kindle Source Code)
- https://www.wired.com/2009/02/amazons-e-books/
- https://the-digital-reader.com/2020/07/05/amazon-removes-active-content-section-of-the-kindle-store/
- https://github.com/schuhumi/alpine_kindle (Alpine Linux For Kindle)
- https://github.com/simsor/kindle-doom (Doom for Kindle)