

Android タブレット、スマートウォッチで Linuxを色々と遊んでみよう

1、自己紹介

2、Androidの紹介、全体像

3、Android上でのLinux動作

- ・VNC、RDPビューワーで遠隔モニタ代わり
- ・Debian noroot、chroot環境
- ・Androidでx86 PCエミュレータ
- ・Nexus7、Androidのroot化
- ・Debian kitとComplete Linux、TightVNCserver、Xserver-SDL
- ・Androidスマートウォッチ近況

4、Nexus7でOS載せ替えテスト

- ・Android独自ビルドとMultiROM、TWRP
- ・FirefoxOS
- ・Ubuntu Touch
- ・Ubuntu13.04とその他Linuxディストリビューション
- ・その他カスタムROM

5、Nexus7の改造

- ・ブートローダー
- ・Linuxディストリビューション
- ・Linuxカーネル
- ・PKGSRC on Android

6、次回出展



講演資料は電子データと
紙面を配布しております。
詳しくはブースまで。

<http://kapper1224.sakura.ne.jp/>

講演：Kapper

OSC 東京2015/10/24 14:00 ~

会場：明星大学 303室

自己紹介 Self Introduction



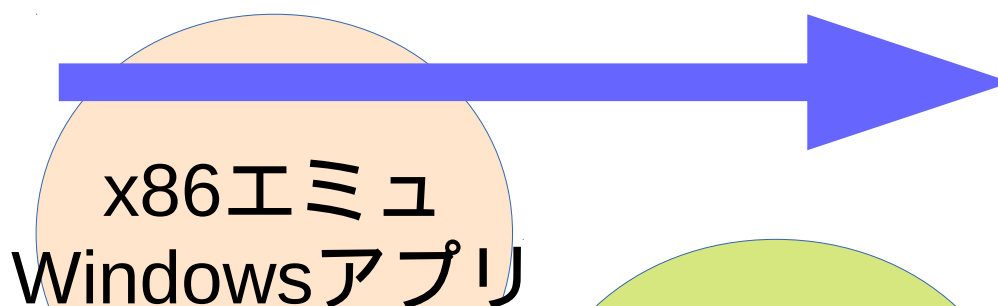
- My name: Kapper
- Twitter account: [@kapper1224](#)
- HP: <http://kapper1224.sakura.ne.jp>
- Slideshare:
- My Hobby: Linux, *BSD, and Mobile ARM Devices
- My favorite words: Record than experiment important
- Test Model: Netwalker(PC-Z1,T1)、Nokia N900、DynabookAZ、RaspberryPi
Nexus7(2012、2013)、Hercules eCAFE EX HD、Jetson TK-1、
OpenPandora、ARM Chromebook、ZTE OPEN C (FirefoxOS)
台湾Android電子辞書 無敵CD-920、CD-928
- Recent Activity:
 - Netwalker on the Linux from Scratch.
 - Hacking of Android Nexus7.
 - I have recently often use the ARM Chromebook.

今回のテーマ

- 今回はマニアックにAndroidを攻めます！
- Windows中華タブレットも始めました
- 中華PADに中華スマートウォッチまで（出来たら）
- 始めての方は前回のNexus7の資料もご一読を
- 資料数が多くなりすぎたので、時間の都合で後半の内容を割愛する可能性があります。
-

今回の話の流れ

簡単
アプリ
入れるだけ



難しい
マニア向け

Mikutter
遠隔操作
Debian Noroot
(通常ユーザー)

x86エミュ
Windowsアプリ

OS載換え
Root化
MultiROM
pkgsrc

ブートローダー
Kernel
SoC

Linux
ディストリ
chroot環境
(root)

重要

長く使う為に

やれる事が
限られる

対象ユーザー（伝えたい事）

Mikutterマニア

ハック、改造用途向け

デベロッパー
ディストリ
ビューター

サーバ用途

「End of Life」延長

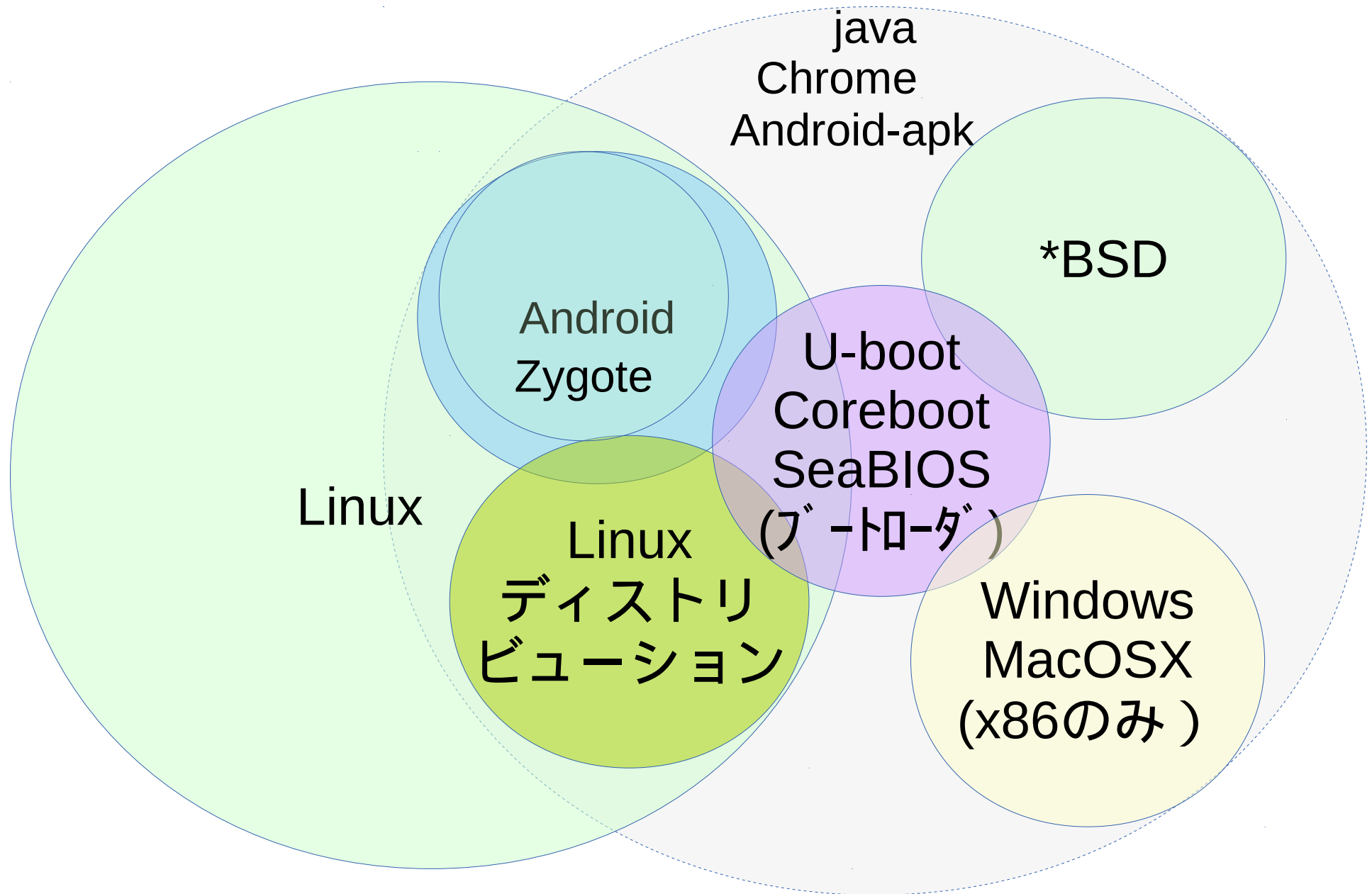
Office用途

最初のご提案

- まずBluetoothキーボードを入手しましょう。
あるととっても便利です。マウスもあるとOfficeで
もっと便利ですが、各自のご判断で。



Androidの全体像



Android改造の簡単な歴史

Android初期 ~ 2.3
root取れない
Webkitのバグから
Root化
↓
改造

Android ~ 4.2
Nexus7登場
ブートローダーが
ロックされていない
↓
Root化可能

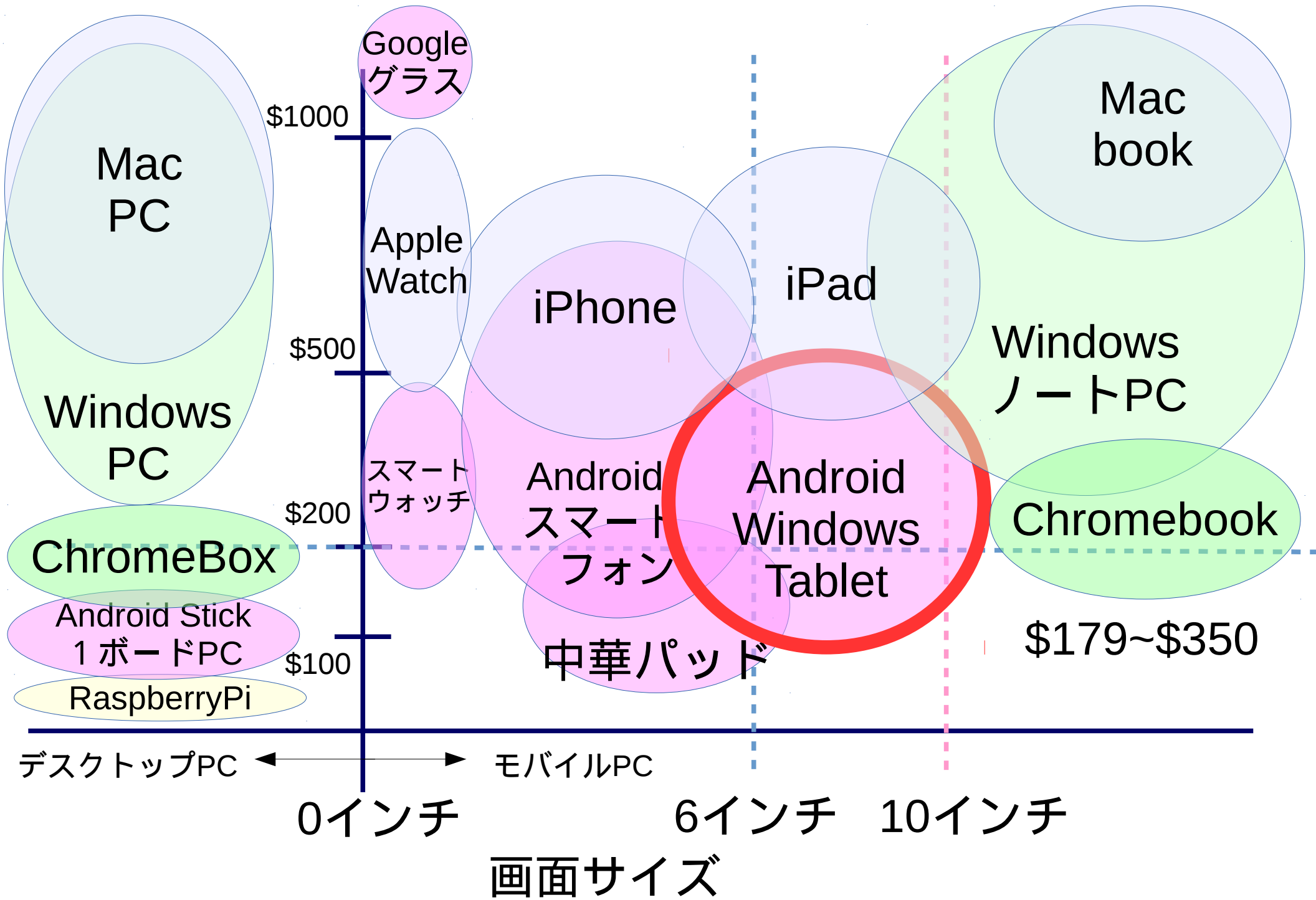
Android ~4.x
中華タブレット全盛
Root化アプリ多数
↓
アプリ入れるだけ

機種限定
規制厳しい

誰でも出来る

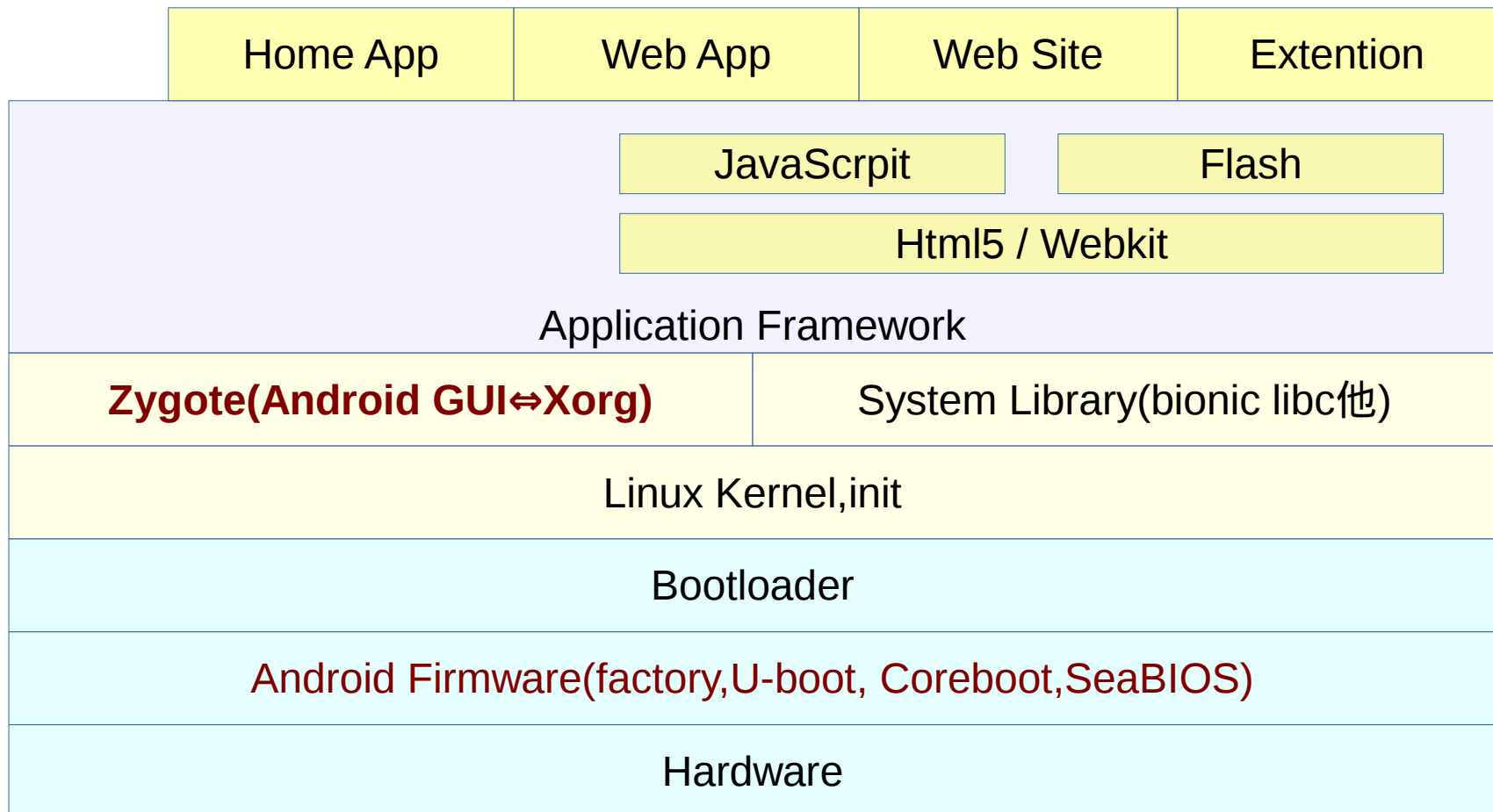
アプリで簡単

PC市場の中のAndroid



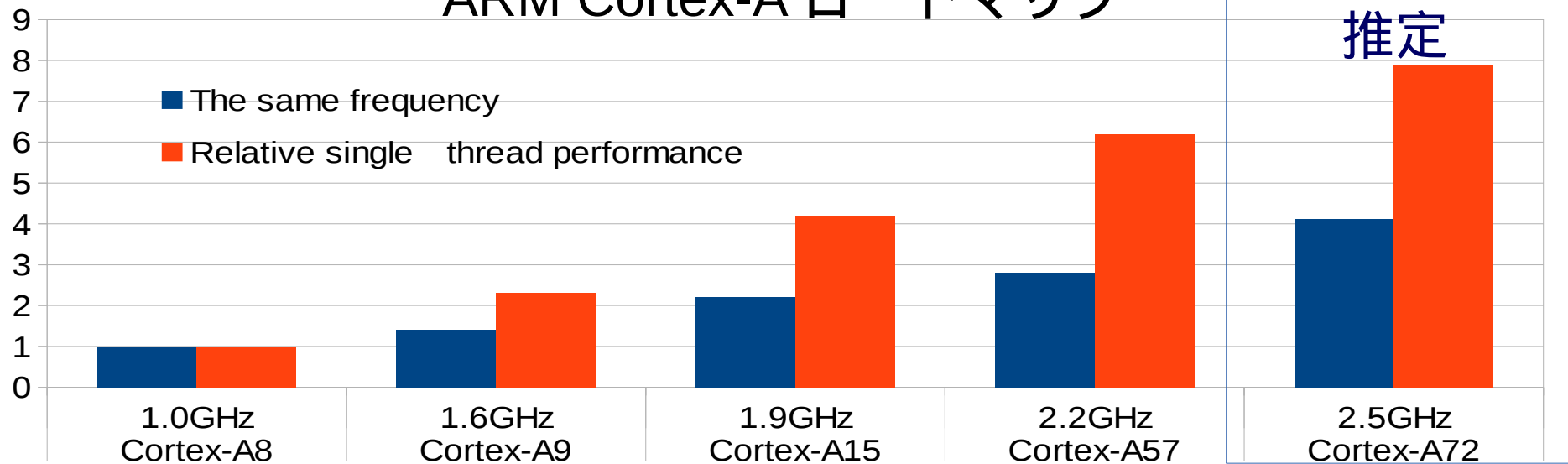
Android OSの仕組み

- Linux Kernelはほぼ同一。Xを使わずにFramebufferを用いて表示、DalvikVM(Java VM)や各種ライブラリを独自実装。libc(bionic libc)もBSDライセンスの独自。
- Bootloaderもメーカー既製品でなくU-bootなど置き換えればほぼ同一。Xを使っていないので別途使用。



ARM端末のベンチマークスコア

ARM Cortex-A ロードマップ

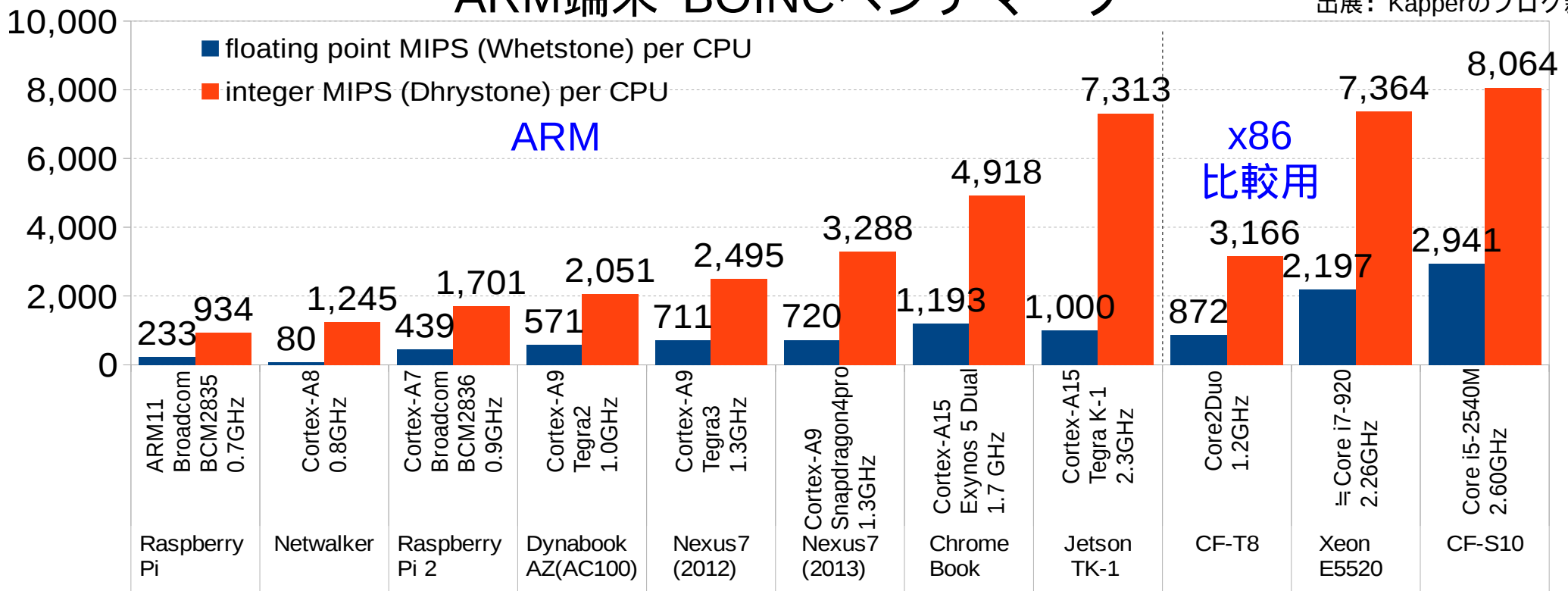


出展: A Walk Through the Cortex-A Mobile Roadmap

<http://community.arm.com/groups/processors/blog/2013/11/19/a-walk-through-the-cortex-a-mobile-roadmap>

ARM端末 BOINCベンチマーク

出展: Kapperのブログ新館



中華PC 大分類

数が多すぎるので抜粋。個人的主観です（後日整理します）

プロセッサ	OS	Root 対応	代替OS、独自ビルド	推定 価格	コメント
中華タブレット 中華スマートフォン 中華Netbook	Android FirefoxOS (ARM、MIPS)	○	CM他	数千円 ～3万円	比較的安価 低スペック 低バッテリー
Androidスティック Androidセットボックス	Android (ARM)	○	CM他 PicUntu	数千円 ～2万円	旧機種は安い 新型は高性能
Chromebook Chromebox	ChromeOS (ARM、x86)	○	Ubuntu、Debian他	1万円 ～3万円	中華系は少数 高性能
ATOM中華Windows Tablet	Windows Android(x86)	△	移植過渡期	1万円 ～2万円	最近の流行 低バッテリー
Windows Stick	Windows (x86)	△	移植過渡期	1万円 ～2万円	意外と高い
1Board PC	Linux(ARM)	○	基本は何でもあり	数千円 ～1万円	RaspberryPi 互換
中華Android スマートウォッチ	Android (ARM、MIPS)	△	なし	数千円 ～2万円	意外と高い 不便

Kapperのネット独断調査（間違っていたらごめんなさい。修正します）

AndroidをLinux端末として使う際に

- Androidとは違う使い方をしたい→何に使うの？
 - 1、Officeやエディタ(キーボード付きミニ端末)
 - 2、Windowsとかx86エミュレータ
 - 3、Mikutter
 - 4、宗教上の理由(以下略)
 - 5、Androidに飽きた、余った、リサイクルしたい
 - 6、FirefoxOSとかUbuntuTouchとかモバイルOS
- 想定される方法
 - 1、Chrootして違うLinuxディストリを起動
 - 2、Debian Norootであっさり起動
 - 3、qemuなどi386エミュレータ
 - 4、ブートローダ書換えOS載替え
 - 5、busyboxやgcc使って自分で作っちゃえw

VNC、RDP ビュワーで遠隔モニタ代わり¹³

- 最初の基本はモニタ代わり、遠隔操作して使う
- RaspberryPiやWindowsの端末として
- Nexus7に限りませんが端末が余ったら使ってみましょう
- TeamviewerやRDPもAndroidに移植されています

BlackberryQ5



Nexus7+Bluetoothキーボード



中華スマートウォッチ



Debian Norootを使ってみる

- Debian NorootはAndroidのアプリでroot権限を取らずにchrootでインストール出来るDebian
- Bluetoothキーボードを使えば、簡易ノートPC
- 中身はfakechrootを使い通常ユーザー権限でchroot環境を構築し、Xserver-SDLでXorgを表示している
- ストレージが内臓/sdcardを前提にしているので容量に課題あり
- 誰でも導入出来る容易さが人気
- 最近のVerはAndroid4.0以降で対応

台湾 無敵シリーズ電子辞書 CD-920



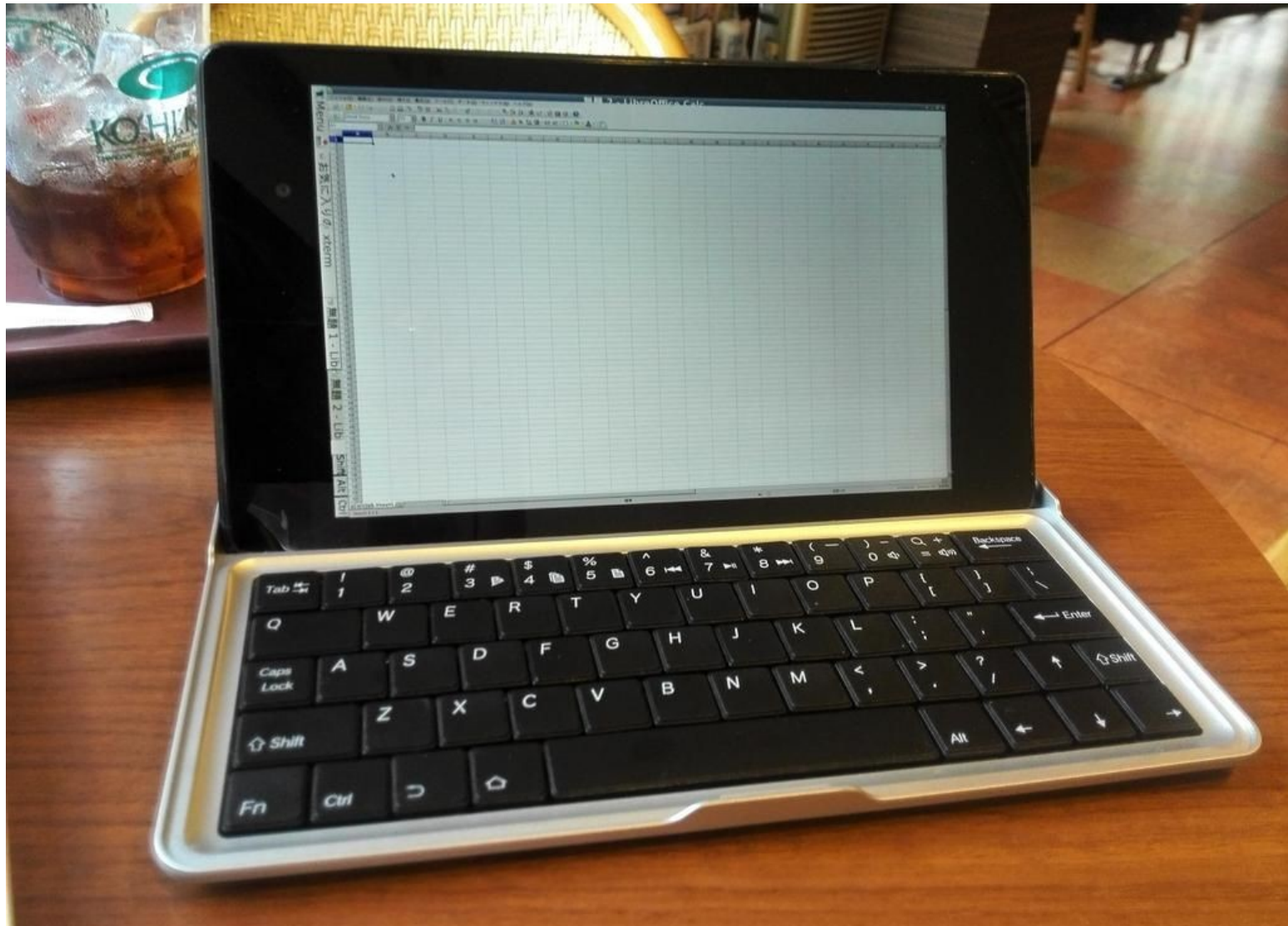
MikutterをDebian norootで使う

- Debian norootを入れればMikutterも動きます。
root端末から、**apt-get install mikutter**
Mikutterは正式にJessieとWheezy Backport
から対応しています。



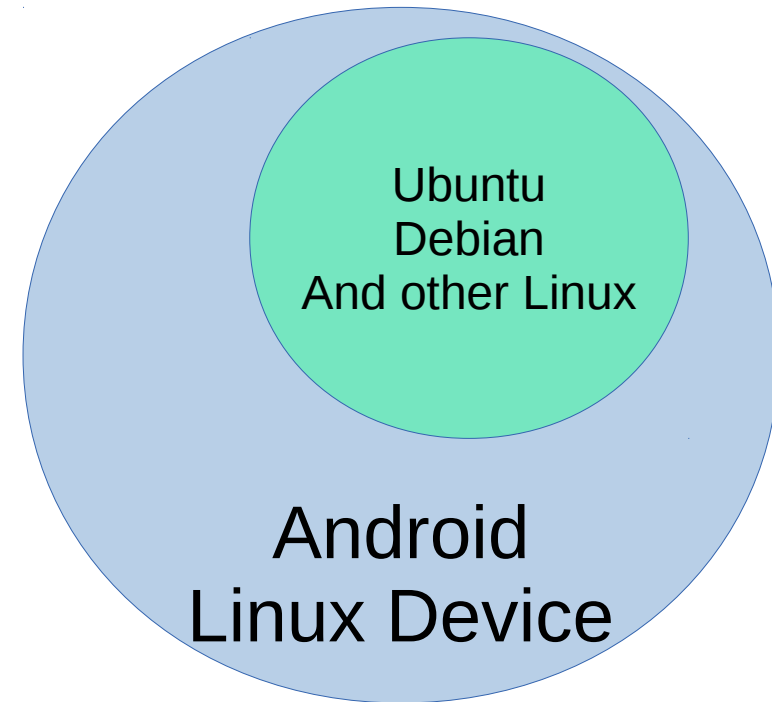
LibreofficeをDebian norootで使う

- Debian norootを入れてLibreofficeも使えます。
スマホやタブレットがOfficeマシンに早や代わり
`sudo apt-get install libreoffice`



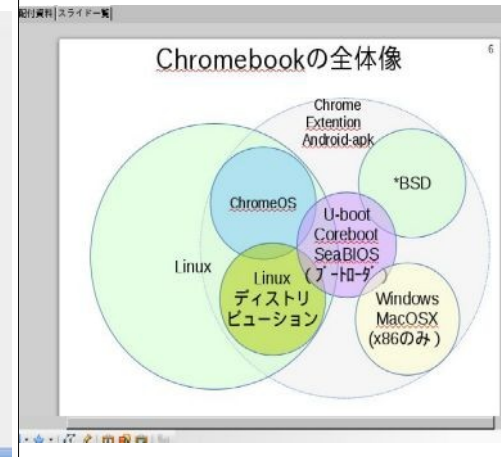
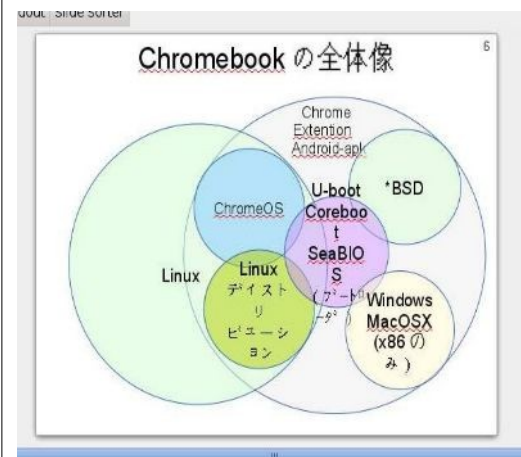
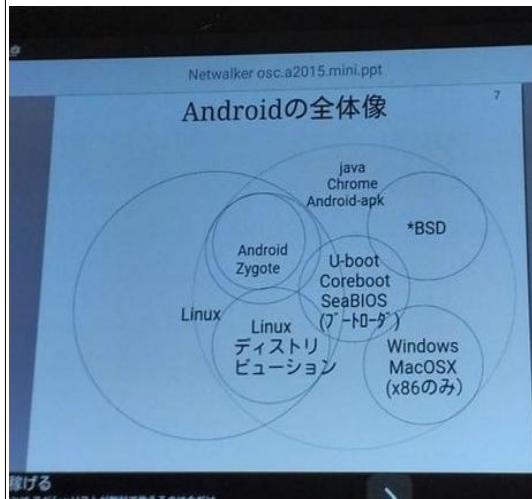
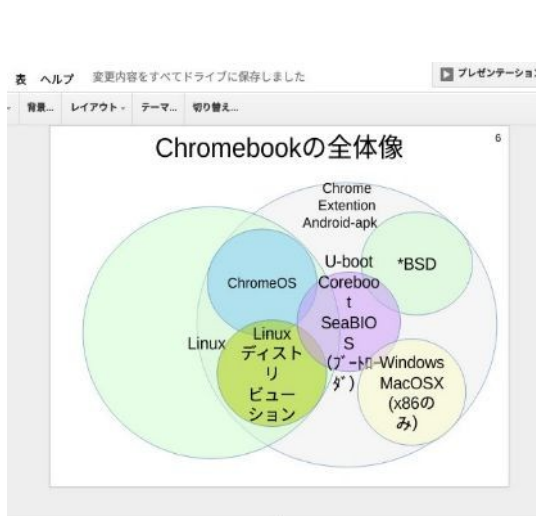
ChrootでARM Linuxインストール

- ChrootはLinux環境をもうつ
- 違うバージョンのOSを起動出来る
- Chrootの中から外の環境へアクセス出来ない
- ChrootはRootユーザーのみ.
一般ユーザーではFakechrootが必要
(Debian Norootなどで使用)
- Kernelは標準OSのまま使用するので、
ドライバやモジュールは、オリジナルKernelのまま。
→オリジナルKernelがサポートしていない
ドライバやOSは使用できない



Android上での互換Office比較

- Android上で動作するOffice環境を比較。閲覧は問題ないが無償版では修正に制限がある。現状ではマウスとキーボード必須かな？
Debian NorootでインストールしたLinuxディストリの場合LibreOfficeでフォントを追加できるので調整可能。お勧め。



△Googleスライド
○MSOffice App
MSやGoogle標準Webオフィス。無料版は表示専用で編集できない。購入必要。
完成度は高いがChromeOSに標準搭載されているフォント数が少ないので、他の環境で作ったデータはフォントが変換されてズレる。
フォント環境はChromeOSの標準環境で追加できないので課題。

△Doc To Go
(Android用互換Office)
互換Officeの一例。無償版は広告付いたり制限あり。性能は思ったより高いが表示と操作性にまだ難があり、Office置き換えまでは届かない。

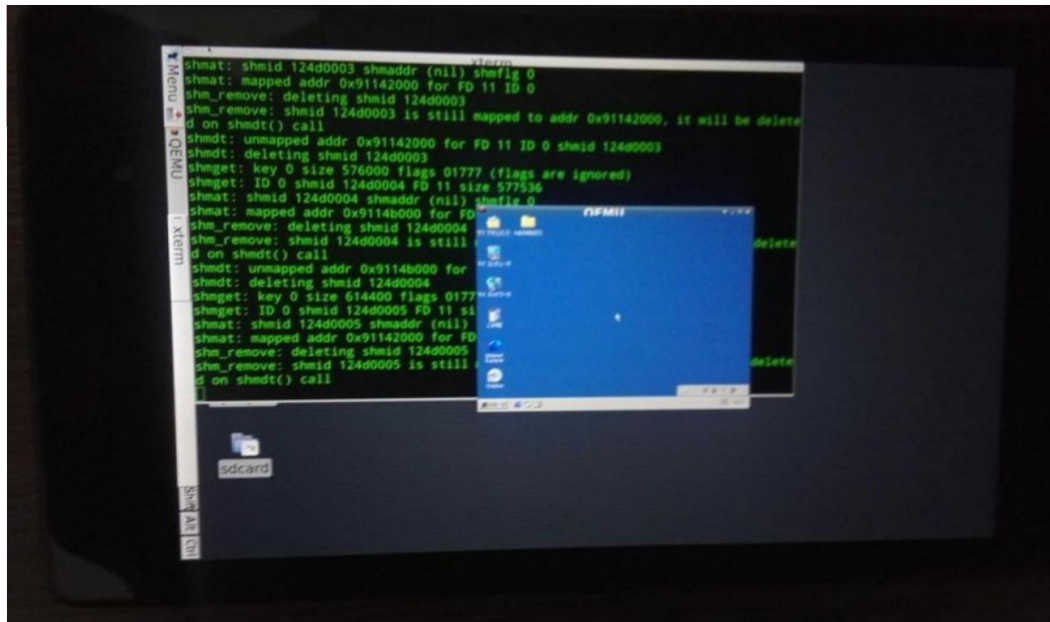
△rollAPP Web
LibreOffice3.3
RollAPPが供給しているWeb版OSSアプリ。LibreOfficeも対応しているが、英語版なので文字がズレる。フォントが少ない。動作もかなり重い。今後の多言語化対応に期待。

◎Debian Noroot上
Libreoffice
Debian Noroot用LibreOffice。容量と日本語化が手間だが、完成度は高い。フォントの追加修正が簡単なので調整が可能
お勧め

Android用PCエミュレータで代替してみる

- 動作は遅く限定的ですが、PCエミュからWindowsを起動することも可能です。
- PC: Bochs,qemu,DOSBox,
MAC:vMacMini
- qemuの場合、Windows以外でもNetBSDも動作しました。

Windows9x、2000



NetBSD ておくれLiveイメージ



DOSBoxでWindows95を動かそう

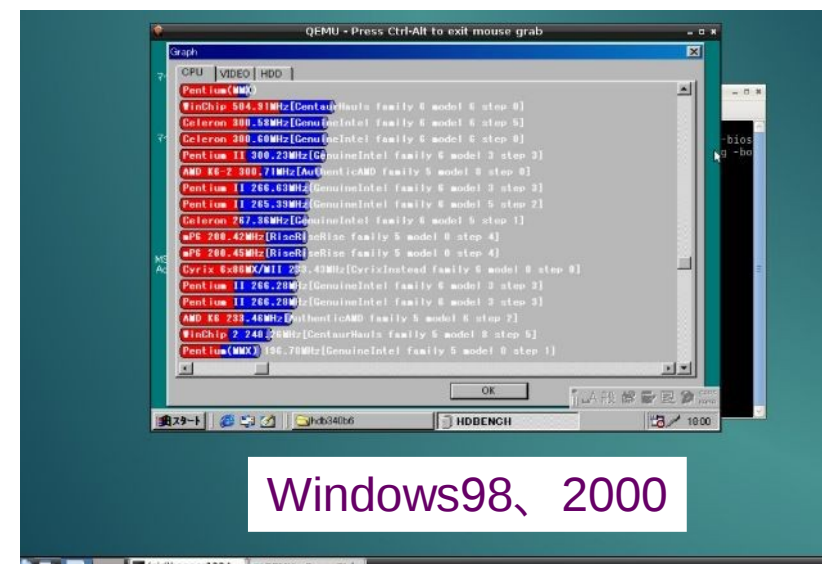
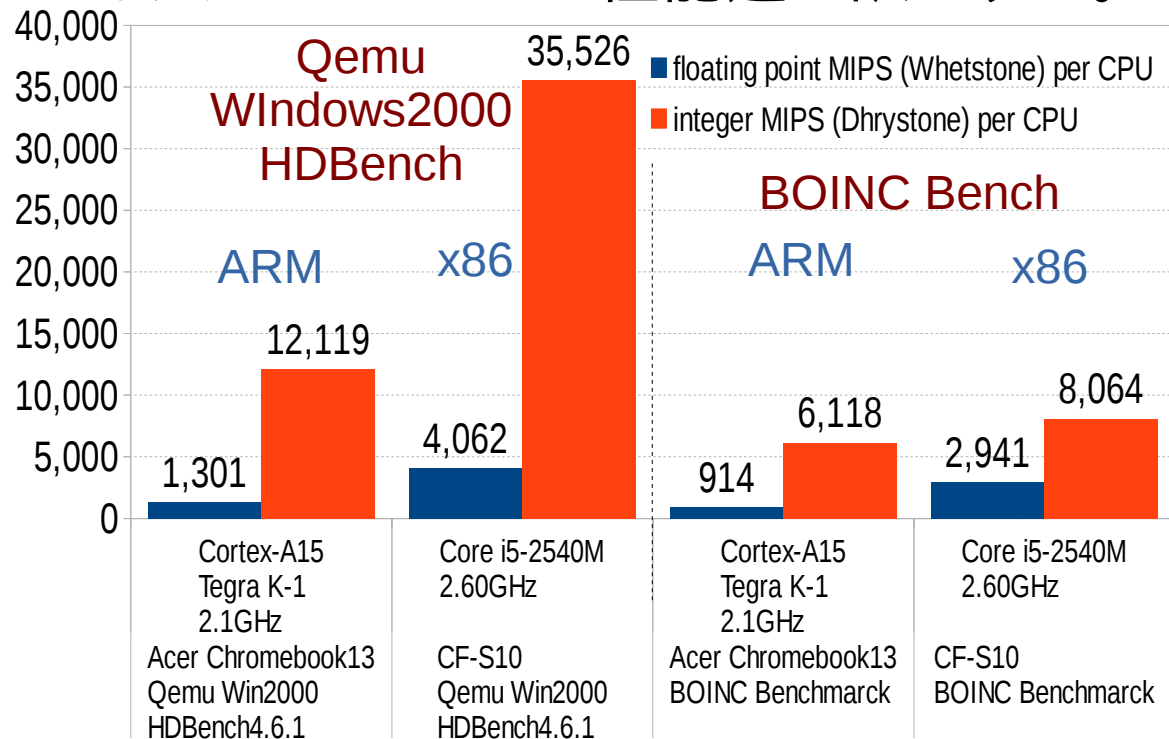
- DOSBoxらWindows9xを起動することも可能です。
- ARMなど移植性が高く、エミュレータの中では高速
- Androidに限らず、沢山の機種やOSで動作出来ます
- もちろんレトロゲームやJW-Cadなども動きます

Windows95



エミュレータで補完してみる

- Ubuntu、Debianを使えばWine、Qemu、Bochs,Steamなどである程度のアプリが動いたりします。(Exagear-DesktopはAndroidで動作せず)
- x86エミュレーションを使えば、WindowsやNetBSDなんかも。
TegraK-1でWin98がHDBenchでPentium2 266MHz相当でした。
Nexus9でWindowを動かす際も似たようなスペックになるかと。
- ARMでのQemuエミュのベンチマークは
x86:ARM=3:1
で浮動小数演算能力にほぼ比例
そりゃARMがエミュ性能遅い訳ですわ。



Windows98、2000



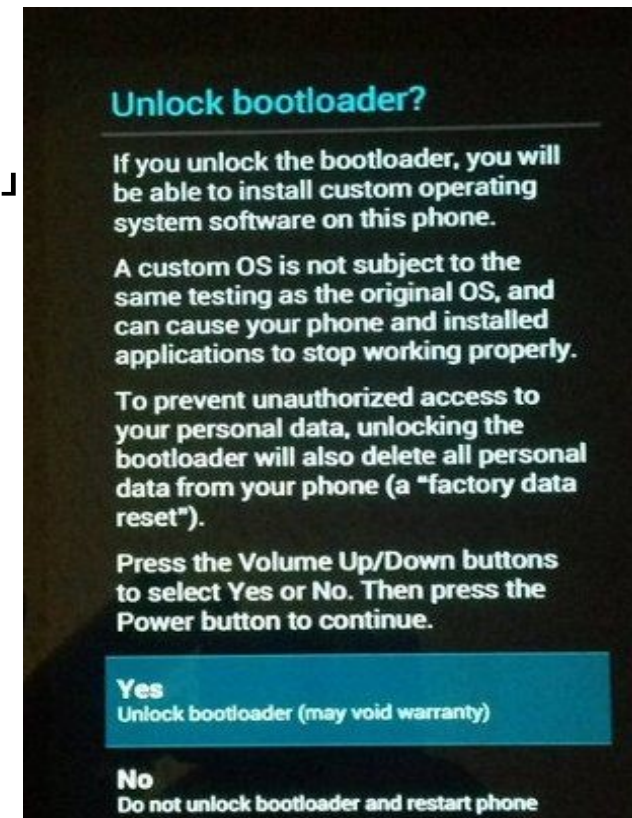
NetBSD ておくれLiveイメージ

AndroidのRoot化

- Androidをroot化することで、様々なアプリと環境が使えるようになります。
root環境でなければ出来ない設定もある。
root化するとメーカーサポートは受けられなくなりますので注意。
- Androidのroot化
 - 1、root化アプリを使ってファームウェアを書換える
 - 2、bootloaderをUnlockする
 - 3、rootアプリを使う
- root化して出来る事
 - 1、OS載せ替え、chroot化、シンボリックリンク
 - 2、アクセス権限の変更、内部アクセス、ターミナル
 - 3、オーバークロック

Root化,Nexus7でのUnlock例

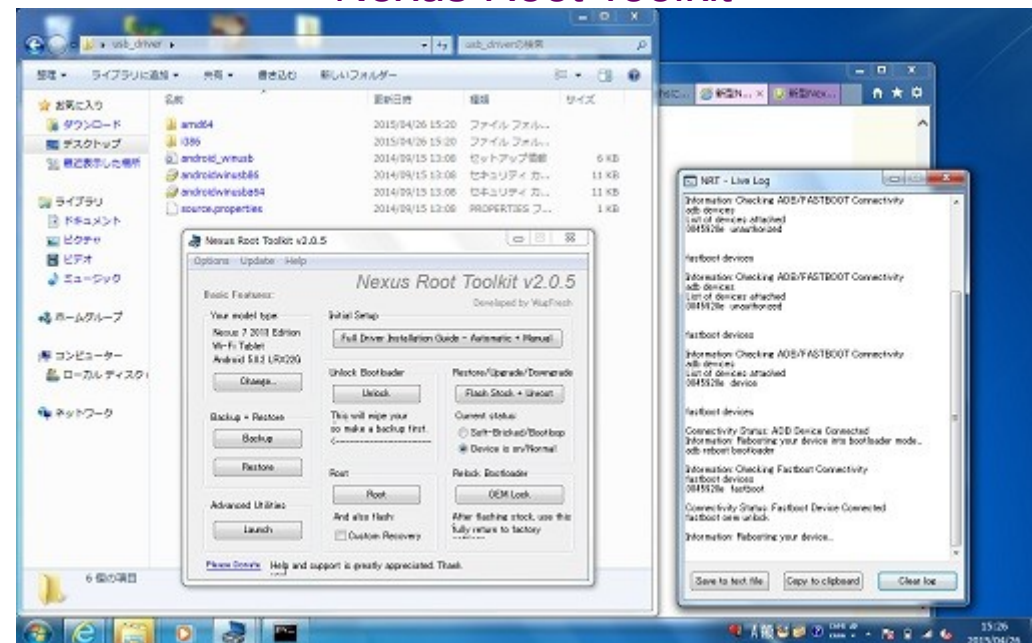
- AndroidのUnlockの一例としてNexus7で試してみます。
Nexus7はメーカーの規制が無い機種です。
個別の機種に関しては、難しい機種も沢山あります。今回は割愛します。
- 用意する物
Ubuntu12.04以降がインストールされたPC (Windowsでも可)
MicroUSB-USBケーブル
Nexus 7 Tablet本体 (8GB, 16GB, or 32GB)
- Android SDKのコマンド、fastbootのインストール
- Nexus7の電源ボタン長押しで電源をOFFにします。
電源が切れましたら右上あたりにあります「電源ボタン」
「volume downボタン」を押しながら再起動します。
ドロイド君の起動画面が出てくるはずです。
- Nexus7とUbuntuPCをMicroUSB-USBケーブルで
繋ぎます。
- PCからfastbootコマンドを実行
`sudo fastboot oem unlock`
- Unlock Bootloaderの画面で「Yes」を選択し再起動



Nexus Root Toolkit

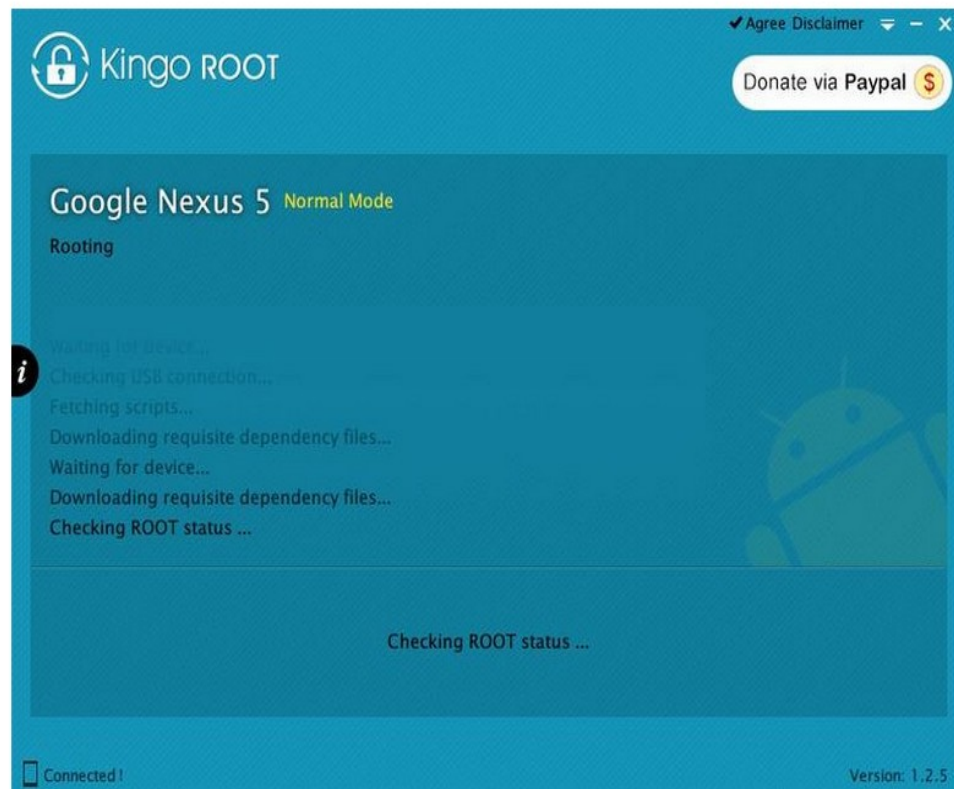
- Windows用ですがGUIでNexus7のUnlock、root化が出来るアプリ
- 専用の環境を作りたくない人にお勧め
 - 1、Nexus Root toolkitアプリのダウンロード
 - 2、Google USBドライバのインストール
 - 3、USBでNexus7を繋いで認識
 - 4、機種を選択しボタンをクリックしてUnlock、root化

Nexus Root Toolkit



スマホ、中華タブレットRoot化

- 最近はRoot化アプリがGooglePlayなどでも公開
- 一例としてKingo Android Root
多数のデバイスをサポート
改造して遊ぶ事が非常に簡単な時代になりました
使用のリスクは自己責任で



Supported Manufacture

MOST POPULAR MANUFACTURERS



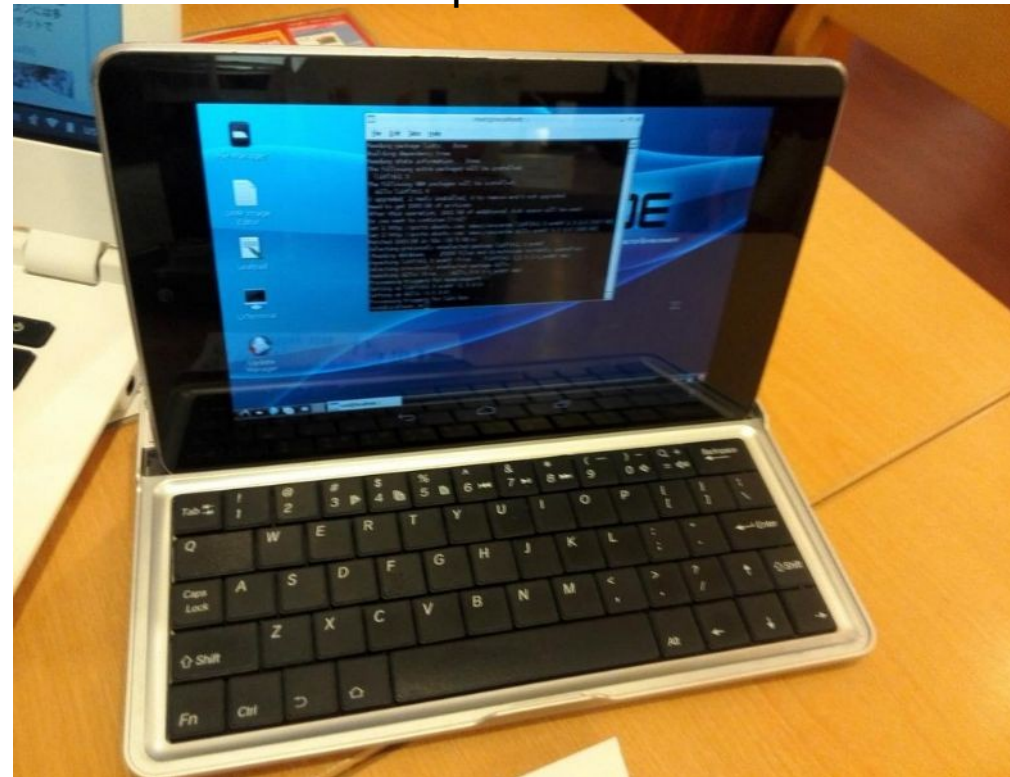
MORE MANUFACTURERS



Debian KitとComplete Linux

- Android上でchroot環境を構築してapt-getしてパッケージを入れるアプリ。Debian化が出来る。
- Androidをシンプルにサーバ化できる。root化が
いるものといらないものと。
- Debian KitはAndroidの親プロセスのルート環境でapt-getが使える仕組みなので、自由度が高い(ハイリスク?)
- Debian Norootが入れられない古い環境向けも

Complete Linux



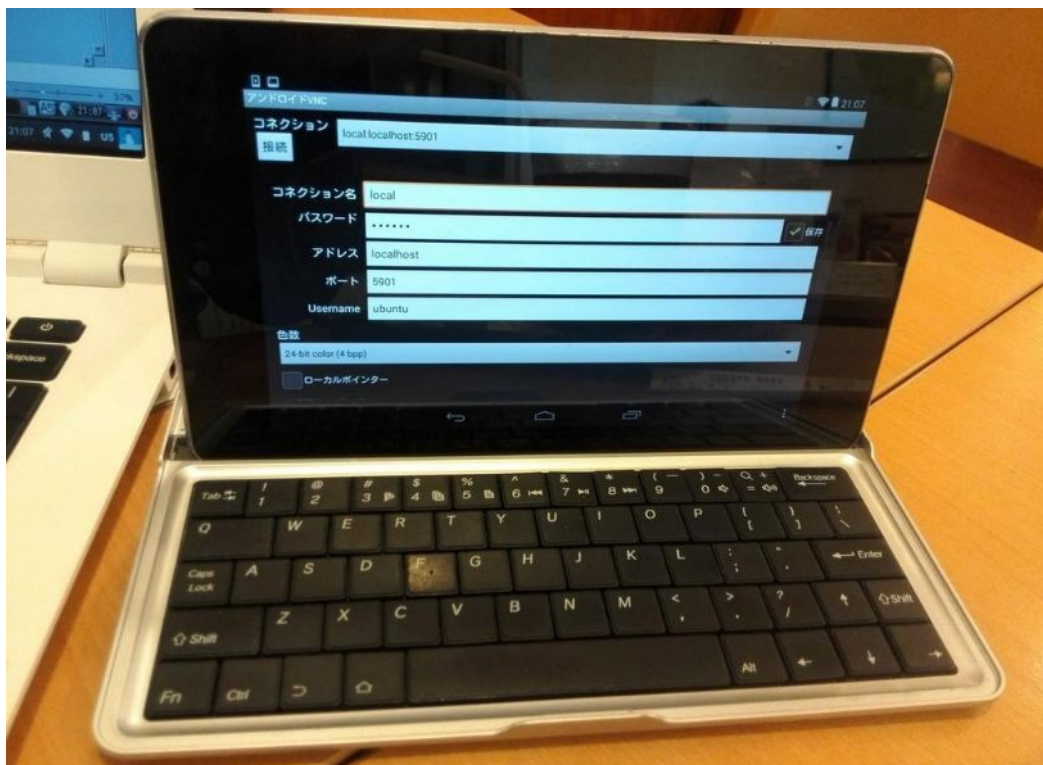
Xサーバを起動する

- AndroidのZygote環境上ではXアプリは使えませんが、Xserverを起動すればXアプリが使えます。
- Androidで使えるXサーバ
 - 1、Xserver-SDL
 - 2、VNC Server
 - 3、Zygoteを落としてXサーバを起動
⇒タッチパネルが効かないのでマウス必須

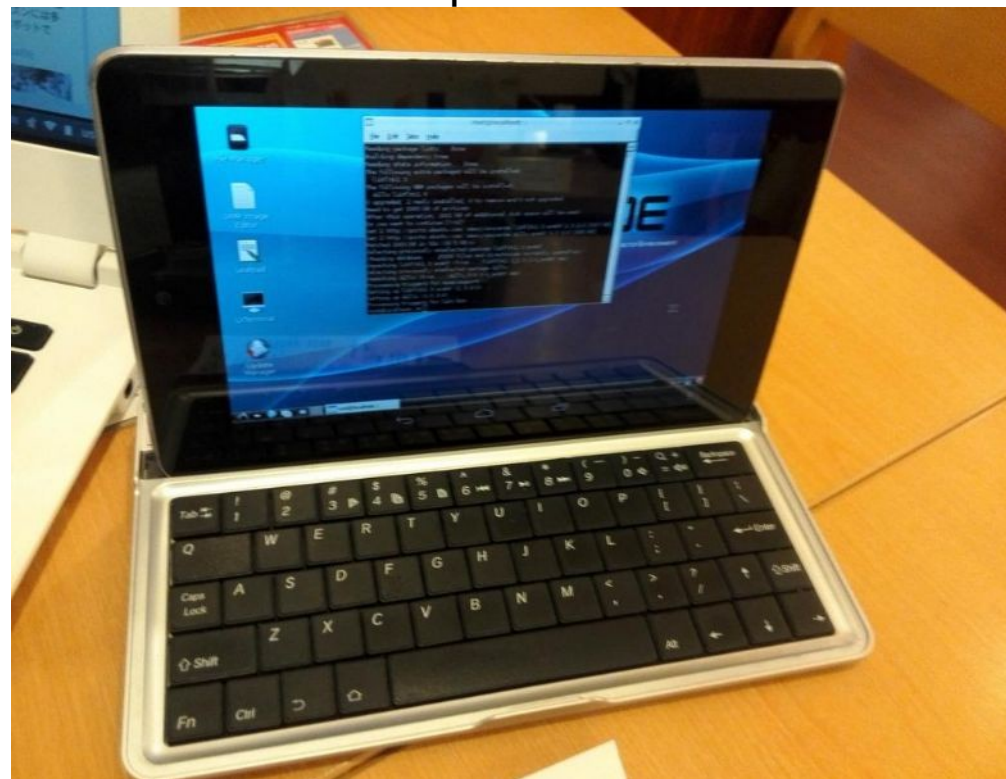
TightVNCserverとVNCviewer

- もっとも容易なXサーバの代替がVNCserver
- chroot環境でTightVNCserverを起動すれば容易にAndroid上でX環境を立ち上げられます。
`sudo apt-get install tightvncserver`
`tightvncserver -geometry 1024x600`
- もちろん、遠隔操作して使うことも可能です。

VNC Viewer



Complete Linux



Xserver-SDLを単独で使う

- Xserverの代替としてDebian norootでも使われているXserver-SDLを単独でも使えます。
- 使い方はXserver-SDLを起動させてから、Chroot環境で端末から、
env DISPLAY=xxx.xxx.xxx.xxx:0 lxsession &
と起動しAndroid上で再度Xserver-SDLを表示

Xserver-SDL

Launch these commands on your Linux PC:

```
env DISPLAY=172.22.57.22:0 metacity &  
env DISPLAY=172.22.57.22:0 gimp
```

To tunnel X over SSH, forward port 6000
in your SSH client

Complete Linux



Zygoteを落としてXサーバを使う

- Kill Android GUI
`setprop ctl.stop media & setprop ctl.stop zygote`
- Restart Android GUI
`setprop ctl.start media & setprop ctl.start zygote`
- #mount chroot img;cd into folder and other mounts
`mount -o loop -t ext4 debian_arm.img deb`
`cd deb`
`busybox mount --bind /dev dev`
`mount -t devpts devpts dev/pts`
`mount -t proc proc proc`
`mount -t sysfs sysfs sys`
`chroot . /bin/bash`
- #and link fb0
`ln -s /dev/graphics/fb0 /dev/fb0`
`startx -- vt1 &`
- #find display
`cd /tmp/.X11-unix && for x in X*; do echo ":{x#X}"; done`
- #set display
`export DISPLAY=:1`

Chroot環境を自分で構築してみる

- 初期設定のAndroidにchrootは出来ないので、USB経由かSDCard経由にインストール。
- 端末からchrootコマンドで起動。

```
sudo debootstrap --arch=armhf --variant=minbase --foreign  
wheezy /media/PHONE¥ CARD/debian  
http://http.debian.net/debian
```

```
export SDCARD=/sdcard  
export ROOT=$SDCARD/debian  
export  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:  
$PATH  
export HOME=/root  
mount -o remount,exec,dev,suid $SDCARD  
for f in dev dev/pts proc sys ; do mount -o bind /$f $ROOT/$f ; done  
chroot $ROOT /bin/bash -l  
debootstrap/debootstrap --second-stage
```

一息いれましょう。Linux何に使うの？

- OSC浜名湖、名古屋、京都でご意見が沢山ありましたのでご質問
- 皆様、パソコンを日常何に使っていますか？
 - 1.UbuntuやDebianとか*BSD、Linux使った事がありますか？
 - 2.RaspberryPiとか1ボードPCを使った事がありますか？
 - 3.電子工作やってますか？興味がありますか？
 - 4.IT勉強会カレンダーご存知ですか？勉強会に興味有りますか？
 - 5.Windowsガジェットやアプリのセミナーに興味有りますか？
(cygwin,qemuソフトウェアや中華Winタブレットなどガジェットヲタク関連)
 - 6.ウェアラブル端末(スマートウォッチ、スマートグラス他)に興味ありますか？
- Linuxって何に使うのですか？メリットは？→何個該当しましたか？
 - 1.WebやSMS、FlashゲームやOfficeなど
 - 2.SkypeやLineなど(Android appやWineなどでも流用)
 - 3.古いPCやスマホのリサイクル用OS、リモートデスクトップ端末
 - 4.サーバとかクラウド環境。NASやルータにも(アクセス数制限がない)
 - 5.3D CAD、3D CGソフトやCAE、Mathematicaなど無料で使いたい。
 - 6.エミュレーションで他OS環境(ゲーム機、Android、FirefoxOSなど含む)
 - 7.開発言語を無料で使いたい。ソフトを作りたい。~2個 一般人
 - 8.RaspberryPiで電子工作をDIYしたい~6個 ヘビーユーザー
 - 9.自宅クラスタ作りたい。超省エネマシンを使って節電したい~8個 マニア、開発者
 - 10.Aircrack-ngで無線LANをモバイルクラッキングしたい~10個 ギーク
 - 11.BOINC廃人でクラスタ演算。がん研究や宇宙人の探索に貢献したいw~12個 廃人
 - 12.北朝鮮OSやMIPS Linuxなどマニアックに使いたい(おい

スマートウォッチ 近況

数が多すぎるので抜粋。個人的主観です（後日整理します）

OS、機種	長所	短所	アプリ (Watchface省略)	Root 対応	コメント
Apple Watch	世界主流 デザイン アプリ多数	高い 母艦iPhone5~ 電池短い	ヘルスケア マップ、Twitter、 リモコン操作、色々	?	世界主流 アプリ多数 買ったばかり
Android Wear	Ingress マップ	UI不便 電池短い	Ingress、マップ Fit、Youtubeなど	OK 要改造	愛用 テストなう
Android中華ウォッチ	アプリ多数	画面が狭い 電池が短い	Android殆ど、艦これ Ingress、リモート他	OK 要改造	テストなう 面白い
Tizen ウォッチ	ブラウザ 大解像度	電池が短い Galaxyのみ	OperaMini最強 スマホっぽい	OK 要改造	テストなう ブラウザ最強
Pebble	Hack可能 電池長い	アプリが 少ない	連携機能 PebbGPS	?	テストなう 先駆者
Bluetooth Watch	安い	最低限	電話、通知機能	?	未使用
Ruputer	PIM	—	元祖 W-PS-DOS ver1.16	—	買えなかった 1998年
Wrist PDA PalmOS	Palm	—	Palm最強	—	買えなかった 2003年
腕時計	電池長い	—	腕時計	—	最強伝説 スマホで十分？

Kapperのネット独断調査（間違っていたらごめんなさい。修正します）

Androidスマートウォッチは実はrootを取ったりadbがそのまま使えたり、apkをインストール出来る

- エミュレータを入れたり、アプリ追加したりしている人も時々見かけます

HOME | SYMPOSIUM | BLOG | RESEARCH | PUBLICATIONS | STUDENT PROJECTS | DOWNLOADS | MISC. | CONTACT

How to root Android 5.0.1 on your Sony SmartWatch 3 (SWR50)



Submitted by Michael Roland on Thu, 2015-01-29 17:38

In my previous post I started to dissect the NFC functionality of our Sony SmartWatch 3 (SWR50). Unfortunately, without having root access it is pretty much impossible to explore the whole file system of the watch. More specifically, the interesting parts of the file system (device files under `/dev` and certain application binaries under `/system/bin`) are concealed from the adb shell user by SELinux. Therefore, we aimed for getting root access on our SWR50.

Luckily, there was already a guide for rooting the SWR50 available on XDA Developers: [\[GUIDE\]\[ROOT\] Smartwatch 3 KNX01V](#). This guide worked like a charm on our watch that was still running Android 4.4W (KNX01V). Unfortunately, we then upgraded to Android 5.0.1 (LWX48P) and, even though the su binaries survived the upgrade, the su command no longer managed to gain root permissions. As a first shot we tried to re-run the rooting boot image from the guide. This wasn't a good idea though. **BE WARNED!** Booting the watch with that boot image soft-bricked our device. The device still booted into Android but never started the Android UI due to hanging in an endless loop. After I recovered from that shock and after some time (hours?!) of debugging, I — **luckily** — managed to recover from that soft-brick by (simply) wiping the data partition.

So why did this happen?

In order to find out what happened, I downloaded the [Android Image Kitchen toolchain](#) and extracted the root file system from the rooting boot image. After analyzing the startup scripts of the rooting boot image, the answer to this question was rather simple: This happened because the `init` scripts of the boot image perform a whole bunch of modifications on the data partition of the watch. While these modifications were just the same that are performed during a regular boot of KNX01V, these modifications were incompatible with the new layout of the data partition in Android 5. Hence, Android 5.0.1 (LWX48P) got stuck in a boot loop.

Creating a less intrusive rooting boot image

I, then, decided to create a less intrusive version of the rooting boot image by removing all unnecessary parts of the `init` scripts. The new image only mounts the system partition, starts a minimal set of system components and (as the original rooting boot image) executes a script (`/sbin/superuser.sh` in the root file system of the image) containing the commands to install the su binaries. Particularly, the new version no longer touches the data partition. You can download this image [here](#). However, this does not solve the problem of getting root privileges on Android 5.0.1. The method of rooting included into that image is simply incapable of obtaining root privileges on Android 5.

Getting root on the SWR50 with Android 5.0.1 (LWX48P)

Consequently, I decided to dig a bit deeper into the prerequisites of rooting Android 5. Luckily, [Chainfire](#) already did a great job on creating SuperSU for Android and making it available as an update.zip ready to be installed through recovery mode. So all I had to do was to extract the installation script and the installable files from the UPDATE-SuperSU-v2.40.zip file. I used version 2.40, which was the most current version at that time. I then ported the installation script to the `/sbin/superuser.sh` shell script of the rooting boot image. Moreover, I reduced the SuperSU installation to only the core components necessary to gain root access through `adb shell`. Specifically, I included the su binaries for the SWR50 hardware platform only and skipped the installation of the SuperSU UI app. You can download this new image [here](#) (and [this repository](#) contains the source files for image creation using Android Image Kitchen). A quick test (boot that rooting boot image and wait until the watch reboots into Android) revealed that SuperSU 2.40 is capable of obtaining root privileges on LWX48P.

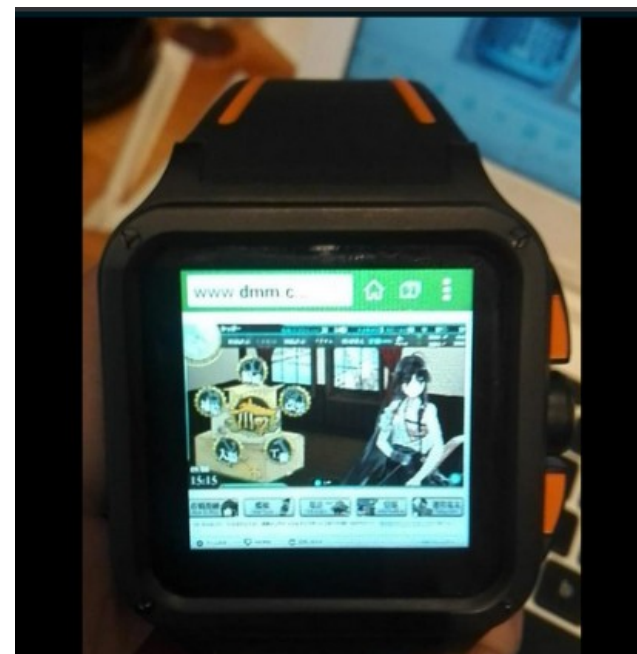
Android WearでDOSbox Win95

- adbコマンドでDOSboxのapkファイルをUSBで入れて、Windows95を動かす動画が2014年に流行りました。



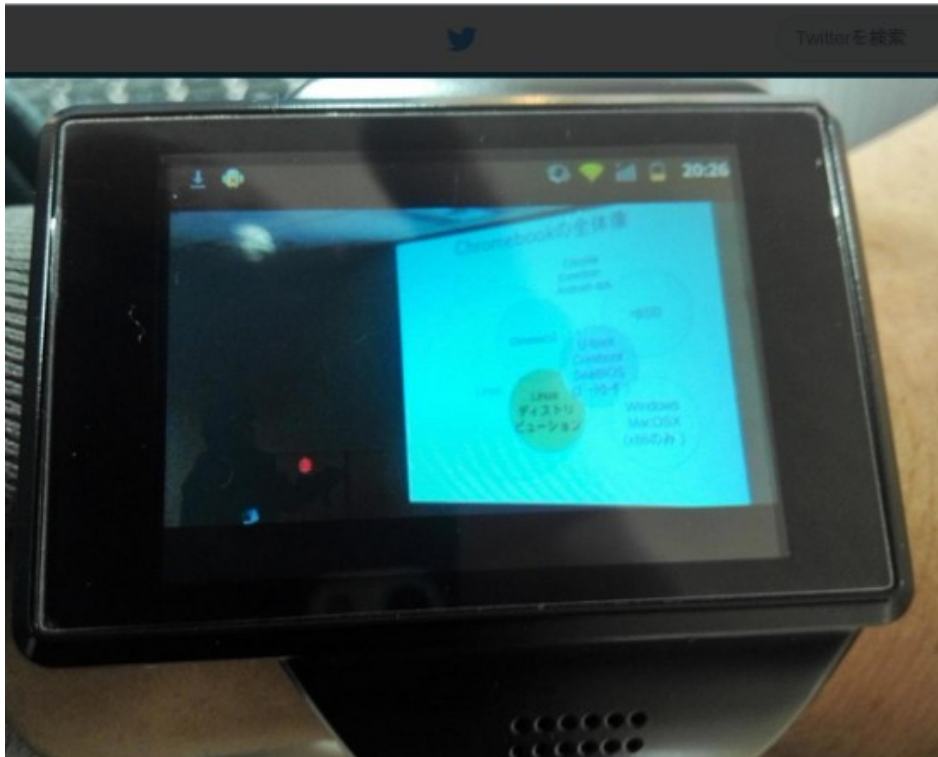
Android 中華ウォッチ

- Androidがそのまま乗った中華ウォッチ
- アプリはAndroidのが大半動作
- 画面が狭いので表示に難あり
- 時計ツールとしては不便だが何でも出来る
- リモート端末やプレイヤーや艦これに



リモートしたり動画やSlideshareとか

- Androidアプリを用いて動画やリモートデスクトップなど
- 実用性はとにかく高解像度化すれば、出来る事が増える
- 会議中にこっそりTwitterとかニュース見たり(おい
- 台風情報とか、天気予報とかは便利です
(使わないとかスマホで十分、とか言う意見は禁句)



中華WindowsタブレットのUnix化(補足)³⁹

- WindowsタブレットでCygwinやエミュでUnix環境

1.Cygwin

2.エミュレータ(qemu,virtualbox,VMware他)

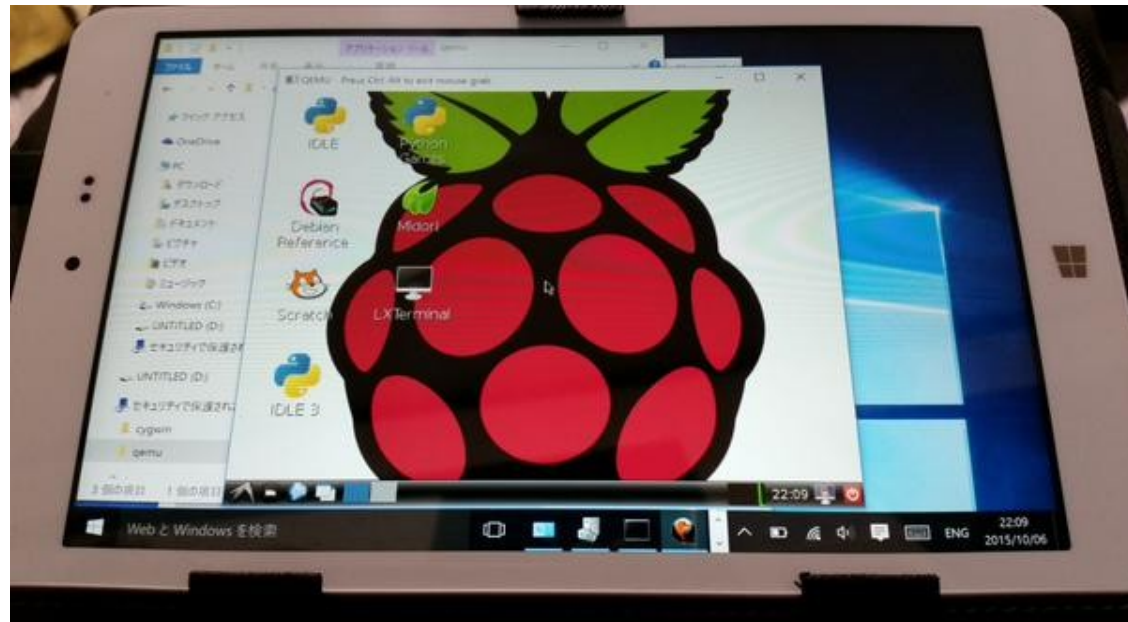
3.Legasy,UEFI BOISからデュアルブート

Linuxのドライバの移植度がまだ低いのが課題
(その筋の専門の人から色々と報告あり)

Cygwin



Qemu



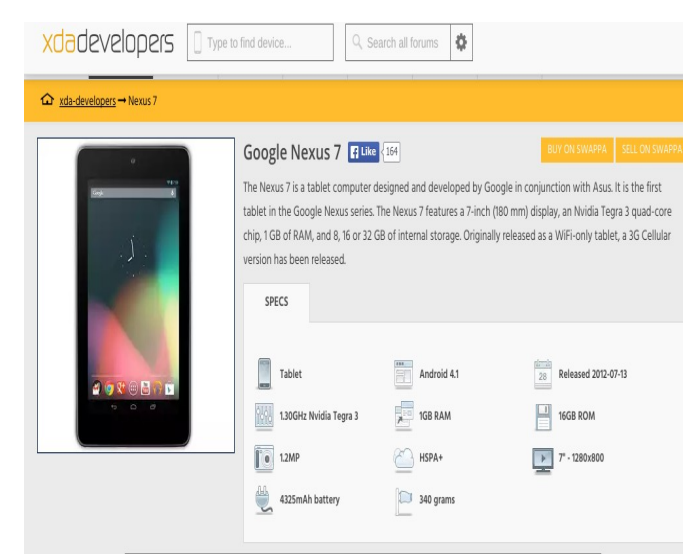
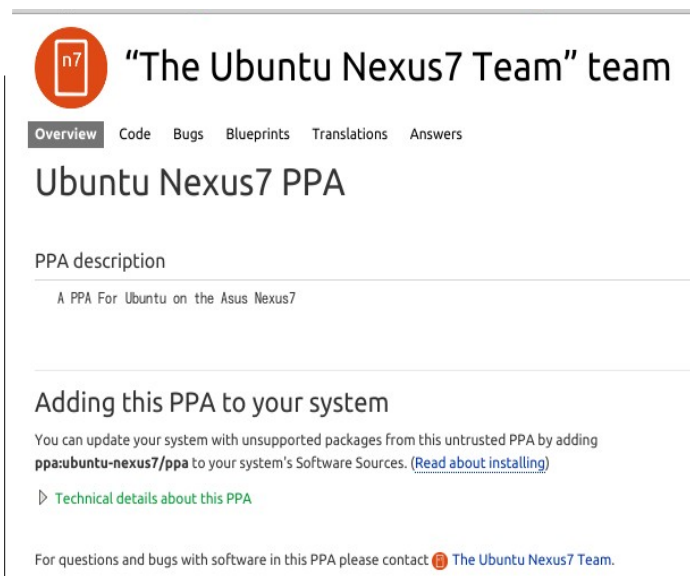
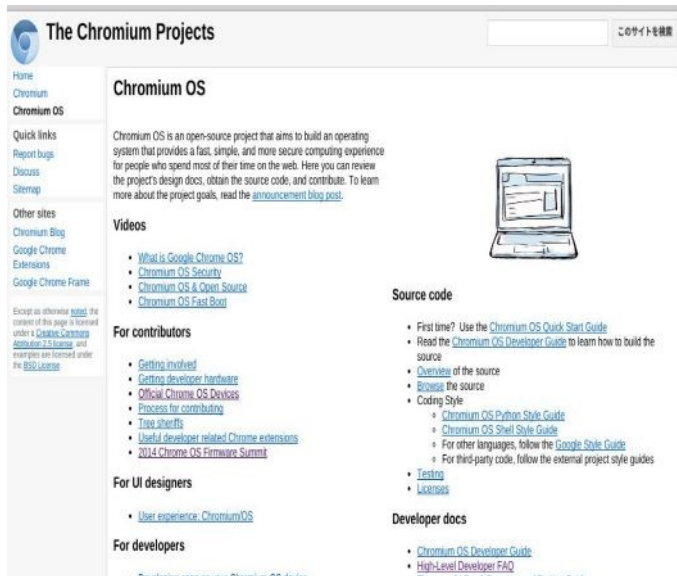
Nexus7のOS乗せ換え

Nexus7にLinuxディストリを入れる

- インストール方法は主に下記3つ。
 - 1、Linuxディストリをchrootインストールする
 - 2、Nexus7用Ubuntu、Bodhiをインストールする
 - 3、ファームウェアを書き換えてブートローダごと各種OSをインストールする

ここからマニアック路線に進みます

Nvidia Tegraハッキングの重要サイト



・ NVIDIA Tegra Public Application Notes
Nvidia公式の開発情報公開HP。
DynabookAZ、Nexus7、Jetson TK-1などのTegraデバイスのSoC情報が載っているサイト。
ソースコードが載っているが、SoCでは結局U-Bootを使っている記載がありました。

・ Ubuntu Nexus7 Team PPA
ソースコードを読むならこちらの方が分かりやすい。
インストーラーからKernel他まで一式あります。
最近はUbuntu Touchへ移行したため、開発停止しています。

・ Xda developer Forum
Nexus7
Androidハッキングの総本山。
多数のスマホ、タブレットの情報が多数。
Nexus7も例外でなく、一度はしっかり読んでみたいサイト

Android対応独自ビルド 状況

Nexus7対応の独自ビルドは数十種類と数が多すぎるので抜粋。

ディストリ	バージョン	インストーラー	Multirom 対応	Root 対応	対応機種
Factoryimg	~5.1.0	◎	○	△	Nexusシリーズのみ
CyanogenMOD	~5.1.x	◎	○	○	多数
AOKP		◎	○	○	多数
ParanoidAndroid		○	○	○	多数
ClockworkMOD		○	○	○	多数
Bodhi Linux	~3.0	○	○	○	Nexus7 2012サポート終了
KDE Plazma		○	△	○	Nexus7 2012サポート終了
FirefoxOS		○	○	△	独自ビルド
Ubuntu Touch		◎	○	○	Nexus7(2013) 4、5、10、6
Tizen		△	△	△	?
SalfishOS		△	△	△	?

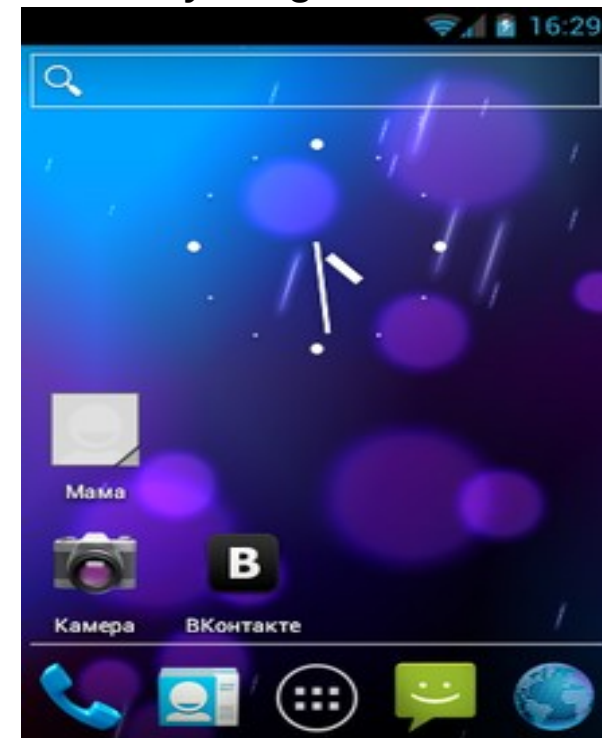
Kapperのネット独断調査（間違っていたらごめんなさい。修正します）

◎：公式サポートあり、○：公式HPに導入報告あり、△：ユーザー導入報告あり

CyanogenMod概要

- CyanogenMod Inc.が開発しているカスタマイズされたAndroidの数ある独自ビルドの一つで各種デバイスにAndroid 4. x を移植
- 公式、非公式なサポートを含めると220デバイスに対応
- ファームウェア（ブートローダー）からOSイメージまで開発。公式サポートされているデバイスであれば、rootを取らずにワンクリックでインストール可能。
- インストールは難しいが謎デバイスにも多数動作報告があり、旧型機種のアフターマーケットに寄与している。
- Nexus7でも正規に対応しているので、簡単に母艦からインストールできます。

CyanogenMod9



CyanogenModインストール

主な流れのイメージ

AndroidにCyanogenModをインストールする

公式サポートデバイス
途方もなく簡単ルート

非公式サポートデバイス
途方もなく難しいルート

Androidのデータをバックアップする

Yes

公式サポート対応している？

No

CyanogenMod Installer
Appをインストール

Windowsに Cyanogen
ModInstallerを
インストール

母艦PCにNVflash、fastbootをインストール

AndroidをUnlock(root化含む)

ブートローダ対応している？

No

ブートローダとドライバ、
設定ファイルをコンパイル

Yes

Androidにブートローダ (U-boot他) を上書きする

No

汎用ARMイメージに
LinuxKernelを
入れ直してrootfs作成

rootfs対応している？

Yes

再起動してCyanogenModのrootfsをSD他からマウント

再起動してCyanogenModが起動

USBで母艦PCに繋いで
ダウンロード、インストール

MultiROMとTWRP

- MultiROMを入れるとブートローダ時点でROMを選択できるようになります。
複数のOSを入れる際には必須です。
Unlockが必要です。
- カスタムリカバリであるTWRPを入れると純正以外のROMを入れたりバックアップしたりできます。NexusRootToolkitで簡単に入れれます

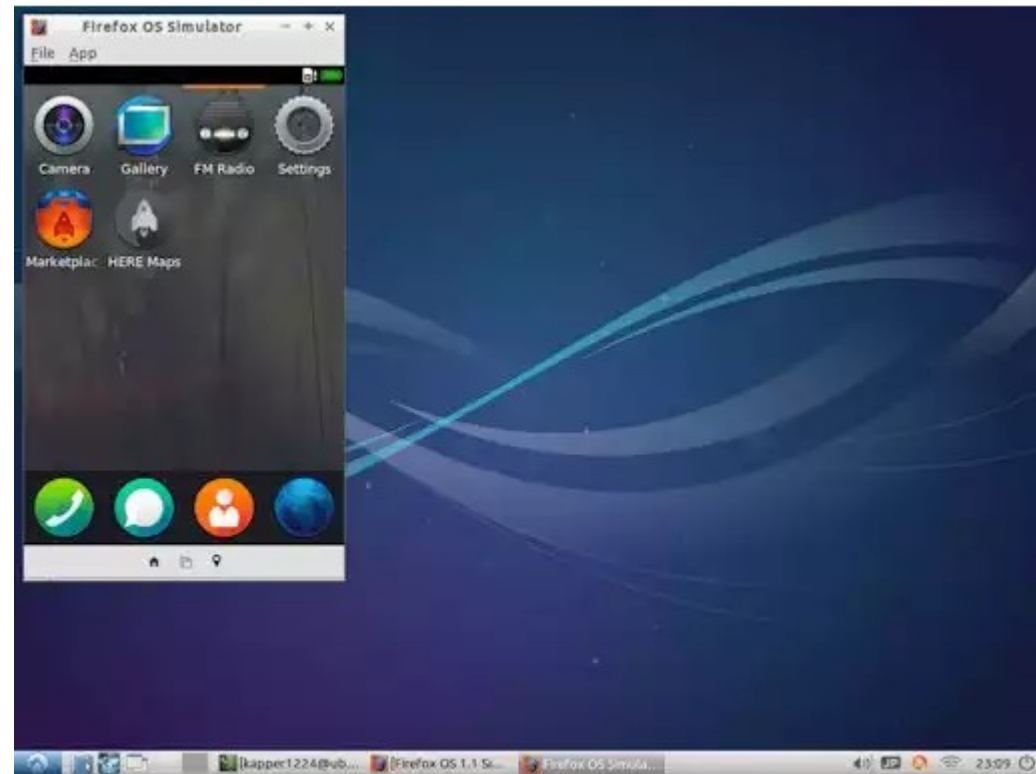
FirefoxOS

- Nexus7でもFirefoxOSをインストールして使うことができます。
- ROMの入れ替えしますのでカスタムROM必須

FirefoxOS



FirefoxOS Simulator



Ubuntu Touch

- Nexus7 2013であればUbuntu Touch対応。
- インストーラーとブートローダーがセット
- 機種限定で使えるのはNexusシリーズ。

Ubuntu Touch



Ubuntu Touch



Nexus7にUbuntu13.04

- 当初Nexus7 2012にUbuntu13.04をインストールできました。デスクトップ版です。
- 今はUbuntuTouchに開発がシフトしています。サポートアウトされました。
現在でもNexus7 2012で使っている人がいる様です。

Ubuntu13.04 Nexus7



Ubuntu13.04 Nexus7



Nexus7の改造とソースコード ブートローダーとOS載せ替え

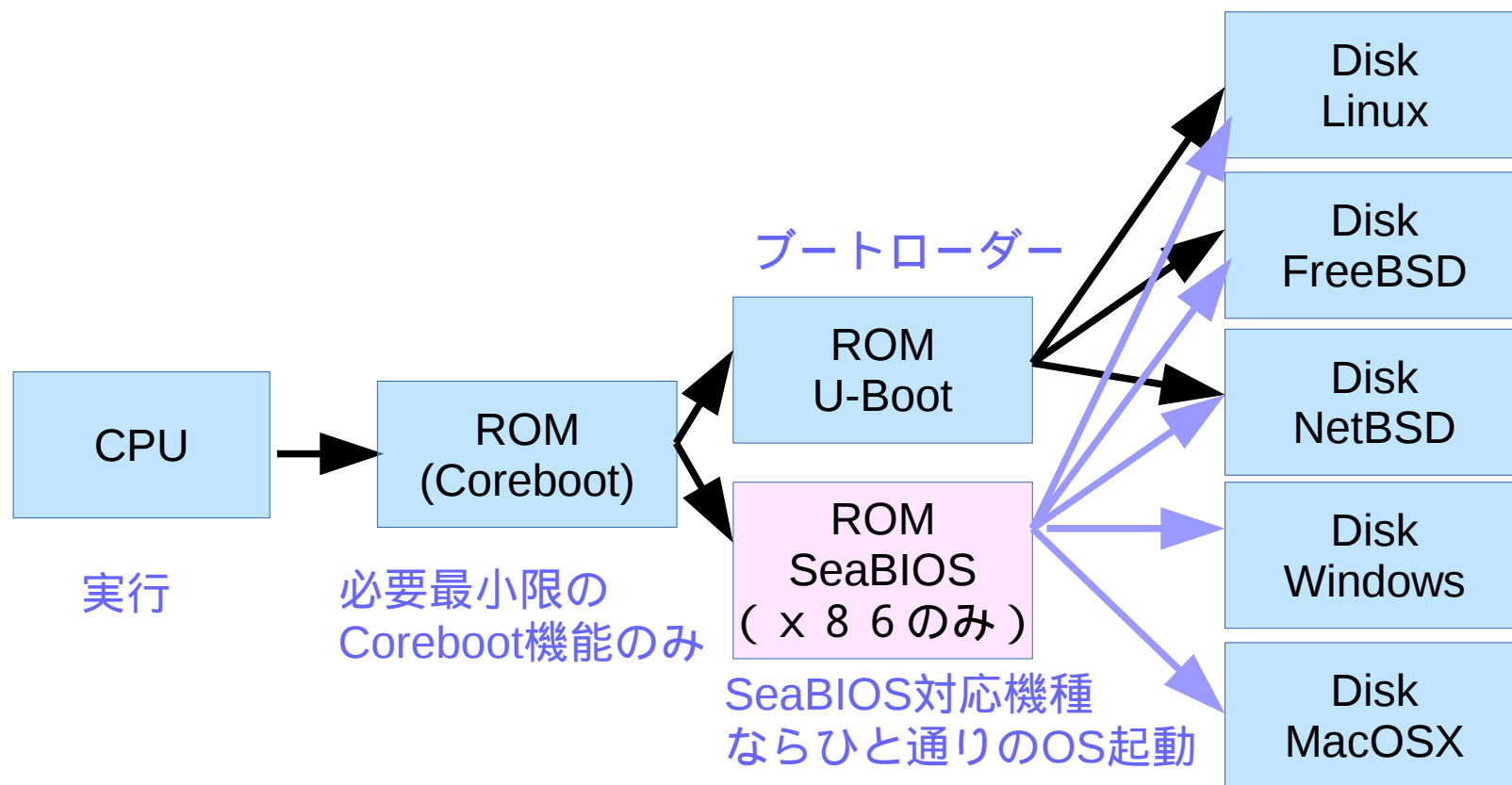
Ubuntuのインストーラーのソースコード 51

- Wgetでイメージをダウンロードしてfastbootで書き込むだけ。

```
set -e
echo "$COUNTER"
echo "# Begining Ubuntu Core Nexus7 installation"; sleep 2
COUNTER=$((COUNTER+20))
echo "$COUNTER"
echo "# Erasing boot partition"
sudo -Sp " -- fastboot erase boot <<<"${SPW}"
COUNTER=$((COUNTER+20))
echo "$COUNTER"
echo "# Flashing boot partition"
sudo -Sp " -- fastboot flash boot ./"${IMAGES[0]}" <<<"${SPW}"
COUNTER=$((COUNTER+20))
echo "$COUNTER"
echo "# Erasing userdata partition"
sudo -Sp " -- fastboot erase userdata <<<"${SPW}"
COUNTER=$((COUNTER+20))
echo "$COUNTER"
echo "# Flashing Ubuntu Core root file system"
sudo -Sp " -- fastboot flash userdata ./"${IMAGES[1]}%.*" <<<"${SPW}"
COUNTER=$((COUNTER+20))
echo "$COUNTER"
echo "# Rebooting device"; sleep 2
sudo -Sp " -- fastboot reboot <<<"${SPW}"
)|
```

Androidのブートローダーって？

- 電源ONしたらメーカー製品はHard Code、1ボードPCはカスタマイズしたBootloaderが立ち上がる設定になっています。
- メーカー製Androidのブートローダーを書き換えるにはUnlockが必要です。
- x86でWindowsなどを動かす場合は、Corebootを入れてからSeaBIOSを起動させる方法がよく使われます。(ARM非対応)



ARMブートローダの仕組みとx86比較

ARMの場合、FlashROMからRAMにブートローダを転送し、Kernelと rootfsを読み込む構成です。ROMに書き込んでるブートローダは「ファームウェア」と呼ぶそうです。Androidの場合、Unlockが必要です。

種類	ROM	RAM	MBR (ディスク)	Kernel	init	ログイン	特徴
x86~ Linux	BIOS デバイス認識 ブートディスク選択		GRUB LILO他	Kernel ドライバ	Daemon 他	CUI Xorg	BIOSと DISKブート ローダー構 成
ARM Android	初期化 ドライバ RAM転送 MLO	Kernel 転送	Kernel rootfs 読み込み	Kernel ドライバ rootfs起動	デバイス Daemon コンテキスト マネージャ Zygote	Dalvik 仮想マシン Android GUI	ブートロー ダーの Unlockが重 要
	Android ブートローダー (ROM)						
x86- Chrome book	初期化 ドライバ RAM転送 CoreBoot	Kernel 転送	Kernel rootfs 読み込み	Kernel ドライバ	Daemon 他	CUI Xorg	ブートロー ダー以降は 共通
	ブートローダー (U-boot、SeaBIOS)						
ARMLinux ディストリ Chrome book	初期化 ドライバ RAM転送 MLO	Kernel 転送	Kernel rootfs 読み込み	Kernel ドライバ	Daemon 他	CUI Xorg	ブートロー ダー以降は x86と同じ
	ブートローダー (U-boot他)						

ARM関連のブートローダ比較

明確な定義は決まってないですが、ROMブートローダとDiskブートローダの2つに分類される事が良くあります。

種類	対応MPU	対応OS	対応ストレージ	対応フォーマット	起動モジュール	ライセンス
GNU GRUB	x86	Linux *BSD MacOSX MSDOS	HDD,floppy USB,LAN TFTP,Serial	全種類	DISK他	GPLv3
LILO	x86	Linux *BSD MSDOS	HDD,floppy USB,LAN TFTP,Serial	全種類	DISK他	BSD Licence
eCos Redboot	ARM,x86,68k,MIPS,Altera,PowerPC,SuperH他	Linux *BSD MacOSX MSDOS	HDD,floppy USB,LAN TFTP,Serial	JFFS2、EXT2、EXT3、EXT4、FAT他	ROM、RAM	Mod GPLv2+
Das U-boot	ARM,x86,68k,MIPS,Altera,PowerPC,SuperH他	Linux *BSD Android他	HDD,floppy USB,LAN,Zip TFTP,Serial NFS	Cramfs、EXT2、EXT3、EXT4、FAT、FDOS、JFFS2、ReiserFS、UBIFS、YAFFS2	ROM、RAM	GPLv2
SeaBIOS	x86	Win,MAC BSD他	SecondaryROM Disk他	EXTx、FAT32、NTFS他	ROM、RAM Coreboot	LGPLv3
Multirom	ARM	Android Linux	Android SecondaryROM USB	EXTx、FAT32、NTFS	ROM、RAM	GPLv3

1、ソースコードをgit-cvs /<vendor>/<name>.

```
git clone https://android.googlesource.com/device/ti/panda
cd panda
git log --max-count=1 kernel
```

2、Kernelバージョンのチェック

```
dd if=kernel bs=1 skip=$(LC_ALL=C grep -a -b -
o '$\x1f\x8b\x08\x00\x00\x00\x00\x00' kernel | cut -d ':' -f 1) | zgrep -a 'Linux version'
```

3、Kernelソースのダウンロード

```
git clone https://android.googlesource.com/kernel/common.git
git clone https://android.googlesource.com/kernel/tegra.git
```

4、gccのPATHを指定

```
$ export PATH=$(pwd)/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin:$PATH
```

もしくはAndroidのビルドツールチェーンをダウンロード

```
git clone https://android.googlesource.com/platform/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6
```

5、ビルド

```
export ARCH=arm
export SUBARCH=arm
export CROSS_COMPILE=arm-eabi-
cd omap
git checkout <commit_from_first_step>
make panda_defconfig
make
```

ZImageの指定

```
export TARGET_PREBUILT_KERNEL=$your_kernel_path/arch/arm/boot/zImage
```

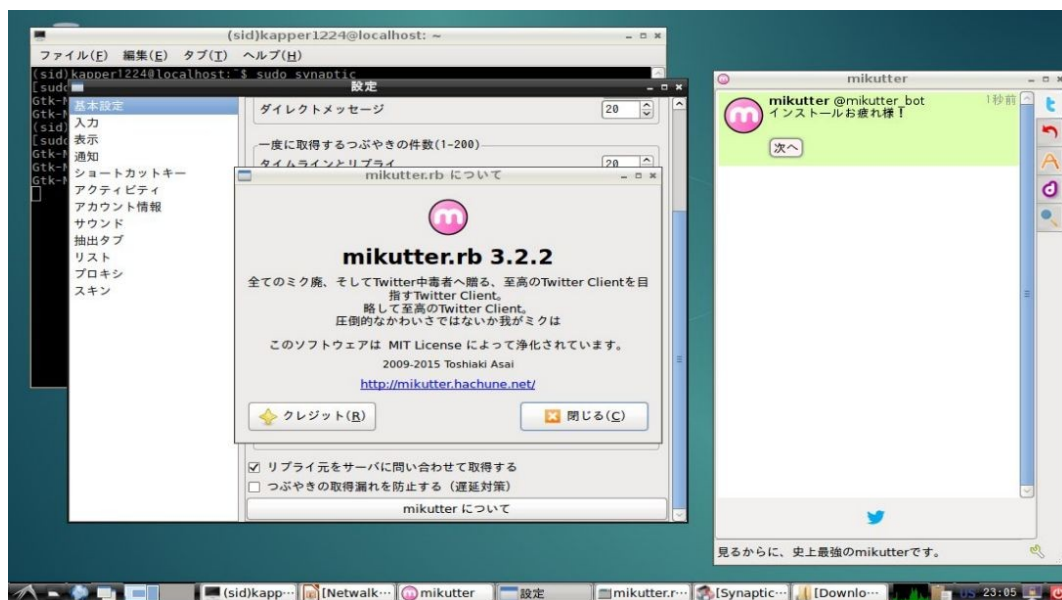
AndroidのLinuxKernelビルドコンフィグ

This table lists the name and locations of the kernel sources and binaries:

Device	Binary location	Source location	Build configuration
shamu	device/moto/shamu-kernel	kernel/msm	shamu_defconfig
fugu	device/asus/fugu-kernel	kernel/x86_64	fugu_defconfig
volantis	device/htc/flounder-kernel	kernel/tegra	flounder_defconfig
hammerhead	device/lge/hammerhead-kernel	kernel/msm	hammerhead_defconfig
flo	device/asus/flo-kernel/kernel	kernel/msm	flo_defconfig
deb	device/asus/flo-kernel/kernel	kernel/msm	flo_defconfig
manta	device/samsung/manta/kernel	kernel/exynos	manta_defconfig
mako	device/lge/mako-kernel/kernel	kernel/msm	mako_defconfig
grouper	device/asus/grouper/kernel	kernel/tegra	tegra3_android_defconfig
tilapia	device/asus/grouper/kernel	kernel/tegra	tegra3_android_defconfig
maguro	device/samsung/tuna/kernel	kernel/omap	tuna_defconfig
toro	device/samsung/tuna/kernel	kernel/omap	tuna_defconfig
panda	device/ti/panda/kernel	kernel/omap	panda_defconfig
stingray	device/moto/wingray/kernel	kernel/tegra	stingray_defconfig
wingray	device/moto/wingray/kernel	kernel/tegra	stingray_defconfig
crespo	device/samsung/crespo/kernel	kernel/samsung	herring_defconfig
crespo4g	device/samsung/crespo/kernel	kernel/samsung	herring_defconfig

PKGSRC on Android

- NetBSDのパッケージャーpkgsrcをUbuntu chromebookでも使ってみる⇒何で？
⇒1, 最新版のパッケージを使いたい。**Mikutterとか随時最新版使いたい**
2, Ubuntuの**PPAでARMバイナリパッケージが殆ど無い**。
ソースパッケージから1個1個手動コンパイル必要。非常に面倒。
3, **OS、アーキテクチャからpkgsrcが独立**しておりUnix、Linux、MacやWinでも使える
ほどの高い移植性がARM環境では大変重宝する
4, Debian SidとかDeb開発版をLTSとかと**混載させたくない**。独立させたい。
5, 仮にUbuntu側とかでARM、**Kernelサポートアウト**されても**継続して最新版使いたい**。
メーカー公式サポートが終わってからも使える最強パッケージャー。
- **最新版Mikutterが光の早さで移植**されて使えます。
最近ディストリ間でMikutterがいかに短い時間でリリースされるか競争してません？



台湾MOPCON2015、KOF2015

58

- 台湾MOPCON、KOFに出展予定
- 内容:OSC東京、KOF2015
スマートウォッチ関係かタブレット関係の予定
- 台湾MOPCON2015 10月31日～11月1日
KOF2015 11月7日
OSC浜名湖 1月23日
OSC東京 2月24～25日
- 東海道らぐのセッションにも是非遊びに来て下さい。
台湾COSCUP2015の話を報告します。
- 質問、ご要望はブースまで是非気軽にお聞き願います。
まだ分からない事沢山あります、ゴメンナサイ。
技術的な話題や議論、雑談を含め何でも。
講演資料は会場でも配布しております。
楽しみにお待ちしております。

中華タブレット プロセッサ別分類

数が多すぎるので抜粋。個人的主観です（後日整理します）

プロセッサ	スペック	Root対応	代替OS、独自ビルド	推定価格	コメント
RK3066,RK2926 RK3088, RK3188,RK3268		OK	Picuntu CM	安価向け	主流
AllwinnerA20 AllwinnerA31					主流
MT8389 MT6577					主流
Exnos5250					サムスン
K3V2					
iMX6					
Atom Z3735 ATM7029					Windowsタブ
OMAP4470					開発終了？

Kapperのネット独断調査（間違っていたらごめんなさい。修正します）