



# ندوة كابترونيك

## مقدمة لـ لينكس المضمنة

توماس بيتسازوني

thomas.petazzoni@bootlin.com

© حقوق الطبع والنشر 2004-2018، بوتلين.

رخصة المشاع الإبداعي BY-SA 3.0.

التصحيحات والأقتراحات والمساهمات والترجمات هي موضع ترحيب!

bootlin



هندسة Linux و kernel المدمجة



# توماس بيتازوني

## توماس بيتازوني

مهندس ومدرس Linux المضمن في Bootlin منذ يناير 2008 مستخدم ومطور Linux منذ عام 2000 أكثر من 120 يوماً من التدريب على المضمن حول العالم لتطوير نواة Linux، وتكامل نظام Linux المضمن، وتحسين وقت التشغيل واستهلاك الطاقة، والاستشارات، ل مختلف العملاء على ARM، MIPS،

الأنظمة القائمة على 68x و Blackfin  
مساهم رئيسي في Buildroot، وهو نظام Linux مفتوح المصدر وبسيط وسريع  
نظام البناء



Bootlin، المتخصصة في Linux المضمن، منذ عام 2005 تركيز قوي على العلاقات المجتمعية التدريب لتطوير نظام Linux المضمن لتطوير نواة Linux وبرامج تشغيل الأجهزة الجلسات العامة القادمة في أفينيون وليون وتولوز، أو جلسات عند العملاء

## موقع

جميع المواد التدريبية متحركة مجاناً بموجب ترخيص المشاع الإبداعي.

التطوير والاستشارات دعم اللوحة تطوير الحزمة أو تحسينها لتطوير النواة وبرامج التشغيل لتكامل نظام Linux المضمن إدارة الطاقة، ووقت التشغيل، وتدقيق الأداء وتحسينه لتطوير تطبيقات Linux المضمنة



# عملاء بوتلن

**ALSTOM** | Transport



**ATMEL**  
Everywhere You Are®

**EBV Elektronik**  
| An Avnet Company |

**MICRONAS**

Nokia Siemens  
Networks

**logiplus**

**freescale™**  
semiconductor

**MARVELL®**

**Gamesa**



**deneo**  
Embedded

**orange** **Continental**®

**SIEMENS**

**THALES**

**Hydro  
Québec**

**ST**



# تدريبات البوتلن





# جدول أعمال

مقدمة: مبادئ البرمجيات مفتوحة المصدر والحرة، المزايا في المساحة المضمنة، الأجهزة الالازمة لنظام التشغيل Linux المضمن ٠ المصادر المفتوح للأنظمة المضمنة: الأدوات، ومحمّلات الإقلاع، والنواة، وأساسات النظام، والرسومات [والوسائل المتعددة](#)، والشبكات، والوقت الفعلي، وما إلى ذلك.

عملية تطوير نظام Linux المدمج ٠ الدعم التجاري، دعم المجتمع

أندرويد

الاستنتاج

سؤال وجواب

# حول البرمجيات الحرة



برمجة الـ C

■ ريتشارد ستالمان: مشروع جنو ومفهوم "البرمجيات الحرة". بداية 1983. تم تطوير تطوير دول مجلس التعاون الخليجي C و GDB و libC وما إلى ذلك.

في عام 1991 أطلق لينوس تورفالدز مشروع Linux، وهو نواة نظام تشغيل تشبه Unix. جنباً إلى جنب مع برمجيات جنو ومكونات البرمجيات الحرة الأخرى، فإنه يخلق نظام تشغيل كامل وقابل للاستخدام: جنو/لينكس، 1995. لينكس يستخدم على نطاق واسع على أنظمة الخادم. 2000 يتم استخدام Linux على نطاق واسع في الأنظمة المدمجة، 2005 يتم استخدام Linux على نطاق واسع في أنظمة سطح المكتب.

لم تعد البرمجيات الحرة شيئاً "جديداً"، بل أصبحت راسخة منذ سنوات عديدة



يعتبر البرنامج مجانيًا عندما يوفر ترخيصه لجميع مستخدميه الحريات التالية:

□ حرية تشغيل البرنامج لأي غرض □ حرية دراسة كيفية عمل البرنامج وتغييره □ حرية إعادة توزيع النسخ □ حرية توزيع نسخ من الإصدارات المعدلة

وتحمّل هذه الحريات للاستخدام التجاري وغير التجاري، دون تمييز.

تعني هذه الحريات أن كود المصدر متاح، ويمكن تعديله ليتناسب مع احتياجات منتج معين، ويمكن توزيع النتيجة على العملاء □ توافق جيد مع الأنظمة المدمجة!

# مزايا المصادر المفتوحة المضمنة أنظمة



# إعادة استخدام المكونات

- الميزة الرئيسية عند استخدام Linux والمكونات مفتوحة المصدر المضمنة الأنظمة هي القدرة على إعادة استخدام المكونات الموجودة.
- يوفر النظام البيئي مفتوح المصدر بالفعل العديد من المكونات للمعايير الميزات، بدءاً من دعم الأجهزة إلى بروتوكولات الشبكة، مروراً بالمكتبات المتعددة الوسائط والرسومية والتشفير وما إلى ذلك.
- بمجرد أن يكون الجهاز أو البروتوكول أو الميزة منتشرة على نطاق واسع بما فيه الكفاية، فرصة كبيرة لوجود مكونات مفتوحة المصدر تدعمها.
- يسمح بتصميم وتطوير المنتجات المعقدة بسرعة، بناءً على المنتجات الموجودة عناصر.
- لا ينبغي لأحد أن يعيد تطوير نواة نظام تشغيل أخرى، أو مكبس TCP/IP أو مكبس USB أو مكتبة أدوات رسومية أخرى. □ يسمح بالتركيز على القيمة المضافة لمنتجك.



## تكلفة منخفضة

يمكن نسخ البرامج المجانية على أي عدد تريده من الأجهزة، مجاناً. إذا كان نظامك المضمن يستخدم برامج مجانية فقط، فيمكنك **تقليل** تكلفة البرامج إلى الصفر. حتى أدوات التطوير مجانية، إلا إذا اخترت إصدار Linux تجارياً مضموناً.

يسمح بالحصول على ميزانية أعلى للأجهزة أو زيادة مهارات الشركة ومعرفتها



مع المصدر المفتوح، لديك الكود المصدري لجميع المكونات في نظامك [ ]يسمح بإجراء تعديلات وتغييرات وضبط وتصحيح أخطاء وتهسينات غير محدودة لفترة غير محدودة من الوقت دون القفل أو التبعية من بائع خارجي [ ]يجب أن يكون صحيح أنه يجب تجنب المكونات غير مفتوحة المصدر عند تصميم النظام وتطويره

[ ]يسمح بالتحكم الكامل في الجزء البرمجي من نظامك



# جودة

يتم استخدام العديد من المكونات مفتوحة المصدر على نطاق واسع، في ملايين الأنظمة جودة أعلى مما يمكن أن ينتجه التطوير الداخلي، أو حتى البائعين الملكية

بالطبع، ليست كل المكونات مفتوحة المصدر ذات نوعية جيدة، ولكن معظمها المستخدمة على نطاق واسع هي. يسمح بتصميم نظامك بمكونات عالية الجودة في

أسس



# اختبار المكونات المحتملة

كون المصدر مفتوحاً متاحاً مجاناً، فمن السهل الحصول عليه وتقييمه ④ يسمح بدراسة العديد من الخيارات بسهولة أثناء الاختيار ④ **أسهل** بكثير من إجراءات الشراء والعرض التوضيحي المطلوبة مع معظم الأشخاص

## منتجات خاصة

④ يسمح باستكشاف الإمكانيات والحلول الجديدة بسهولة



# دعم المجتمع

تم تطوير مكونات البرامج مفتوحة المصدر بواسطة مجتمعات المطورين والمستخدمين

يمكن لهذا المجتمع تقديم دعم عالي الجودة: يمكنك الاتصال مباشرةً بالمطورون الرئيسيون للمكون الذي تستخدمنه غالباً ما يكون هذا أفضل من الدعم التقليدي، ولكن يحتاج المرء إلى فهم كيفية عمل المجتمع لاستخدام إمكانيات دعم المجتمع بشكل صحيح. يسمح بتسريع حل المشكلات عند تطوير نظامك



# المشاركة في المجتمع

## إمكانيات المشاركة في مجتمع التطوير لبعض

المكونات المستخدمة في الأنظمة المضمنة: الإبلاغ عن الأخطاء، واختبار الإصدارات أو الميزات الجديدة، والتصحيحات التي تعمل على إصلاح الأخطاء أو إضافة ميزات جديدة، وما إلى ذلك. في معظم الأحيان لا تمثل المكونات مفتوحة المصدر القيمة الأساسية للمنتج: بل هي الفائدة من الجميع للمساهمة مرة أخرى. بالنسبة للمهندسين: طريقة محفزة للغاية للتميز خارج الشركة، والتواصل مع الآخرين في نفس المجال، وفتح إمكانيات جديدة، وما إلى ذلك.

## بالنسبة للمديرين : عامل التحفيز للمهندسين يسمح للشركة بذلك

معترف بها في مجتمع المصادر المفتوحة وبالتالي تحصل على الدعم بسهولة أكبر وتكون أكثر جاذبية لمطوري المصادر المفتوحة



☒ خيار كبير: دعم المجتمع أم الدعم التجاري؟ أي شركة للدعم؟ أي حل برمجي؟ في الوقت نفسه، هناك قوة وعيوب في البرمجيات الحرة [والمصدر المفتوح](#) ☒ مهارات جديدة مطلوبة مقارنة بالتطوير المعدني أو التطوير باستخدام

أنظمة التشغيل المدمجة التقليدية. ☒ الحاجة إلى التدريب [الحاجة إلى](#) تعيين ملفات تعريف جديدة ☒ الخوف من الترخيص [مبالغ فيه](#) بشكل عام

# الأجهزة لنظام التشغيل Linux المدمج



## بنية المعالج

نواة Linux ومعظم المكونات الأخرى المعتمدة على البنية أ مجموعة واسعة من أبنية 32 و 64 بت x86\_64، كما هو موجود على منصات الكمبيوتر الشخصي، ولكن أيضًا الأنظمة المدمجة (الوسائط المتعددة، الصناعية) ARM مع مئات من شرائح SoC المختلفة (الوسائط المتعددة، الصناعية) PowerPC (التطبيقات الصناعية في الوقت الفعلي بشكل أساسي) MIPS (تطبيقات الشبكات بشكل أساسي) SuperH (بشكل أساسي تطبيقات أجهزة الاستقبال والوسائط المتعددة) Tile, Xtensa, Cris, FRV, AVR32, M32R (نواة ناعمة لـ DSP) Microblaze (بنية Blackfin Xilinx FPGA) Coldfire, SCore,

يتم دعم كل من معماريات MMU و no-MMU على الرغم من أن البنى التي لا تحتوي على MMU لها بعض القيود.

نظام Linux غير مصمم لوحدات التحكم الصغيرة. إلى جانب سلسلة الأدوات ومحمل التشغيل والنواة، تكون جميع المكونات الأخرى بشكل عام مستقلة عن البنية



ذاكرة الوصول العشوائي: يمكن لنظام Linux الأساسي جدًا أن يعمل في حدود 8 ميجابايت من ذاكرة الوصول العشوائي، بل أكثر من ذلك سيطلب النظام الواقعي عادةً ما لا يقل عن 32 ميجابايت من ذاكرة الوصول العشوائي. يعتمد على نوع وحجم التطبيقات. التخزين: يمكن لنظام Linux الأساسي جدًا أن يعمل في مساحة تخزين تبلغ 4 ميجابايت، ولكن عادةً هناك حاجة إلى المزيد.

يتم دعم تخزين الفلاش، كلاً من فلاش ،NOR وNAND مع أنظمة ملفات محددة يتم دعم تخزين الكتل بما في ذلك بطاقات SD/MMC وeMMC ليس من المثير للاهتمام بالضرورة أن تكون مقيداً للغاية على مقدار ذاكرة الوصول العشوائي/التخزين: إن وجود مرونة على هذا المستوى يسمح بذلك إعادة استخدام أكبر عدد ممكن من المكونات الموجودة.



# تواصل

□ تدعم نواة Linux العديد من ناقلات الاتصالات الشائعة

□ SPI □

□ I2C

يمكن

□ سلك واحد

□ USB

□ SDIO

□ وأيضاً دعم شامل للشبكات Ethernet و Wifi و Bluetooth و NAC

وما إلى ذلك. IPv4 و IPv6 و TCP و UDP و SCTP و DCCP وما إلى ذلك.

□ جدار الحماية والتوجيه المتقدم والبث المتعدد



# والنظام على الرقاقةARM

■ تعد ARM إحدى البنى الأكثر شيوعاً المستخدمة في أنظمة Linux المضمنة تصميم ARM نوى ووحدة المعالجة المركزية (مجموعات التعليميات، وذاكرة التخزين المؤقت، MMU وما إلى ذلك) وتتبع التصميم إلى المرخص لهم المرخص لهم ARM مؤسسون، Freescale، Texas Instruments، ST Ericsson، Atmel، مع العديد من الأجهزة الطرفية، في شريحة تسمى ، للنظام على الشريحة

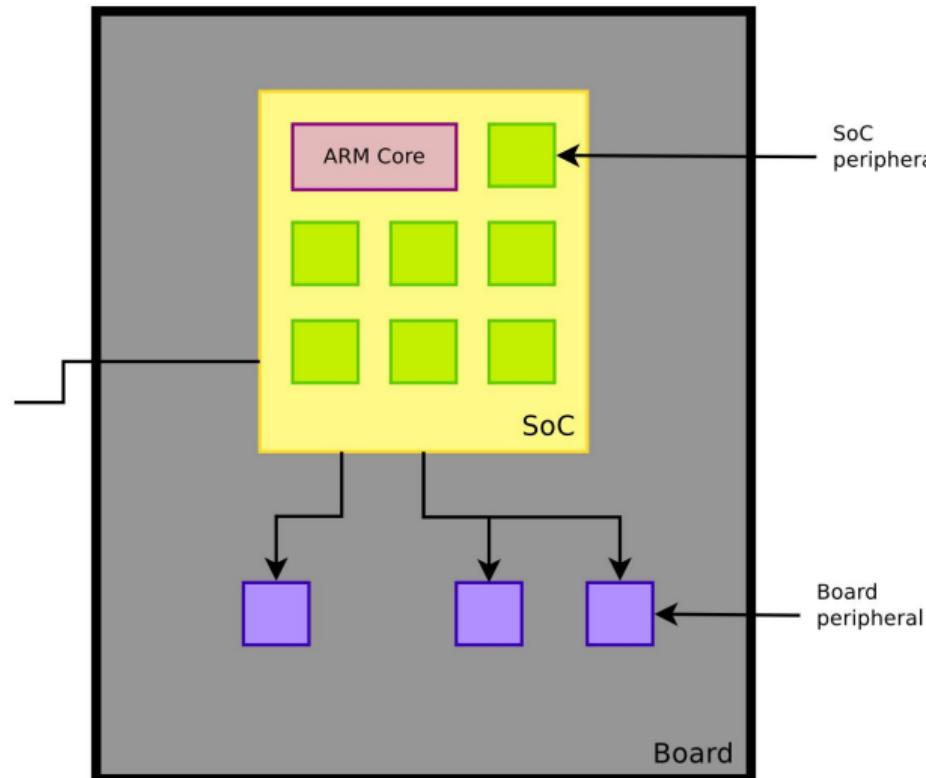
■ يقدم كل مؤسس نماذج مختلفة من شركة نفط الجنوب، مع مجموعة مختلفة من الأجهزة الطرفية، والطاقة، واستهلاك الطاقة، وما إلى ذلك.

■ يسمح مفهوم SOC بتقليل عدد الأجهزة الطرفية اللازمة على اللوحة، وبالتالي تكلفة تصميم وبناء اللوحة.

■ يدعم Linux شرائح SOC من معظم البائعين، ولكن ليس كلهم، وليس بنفس الطريقة مستوى الوظيفة.

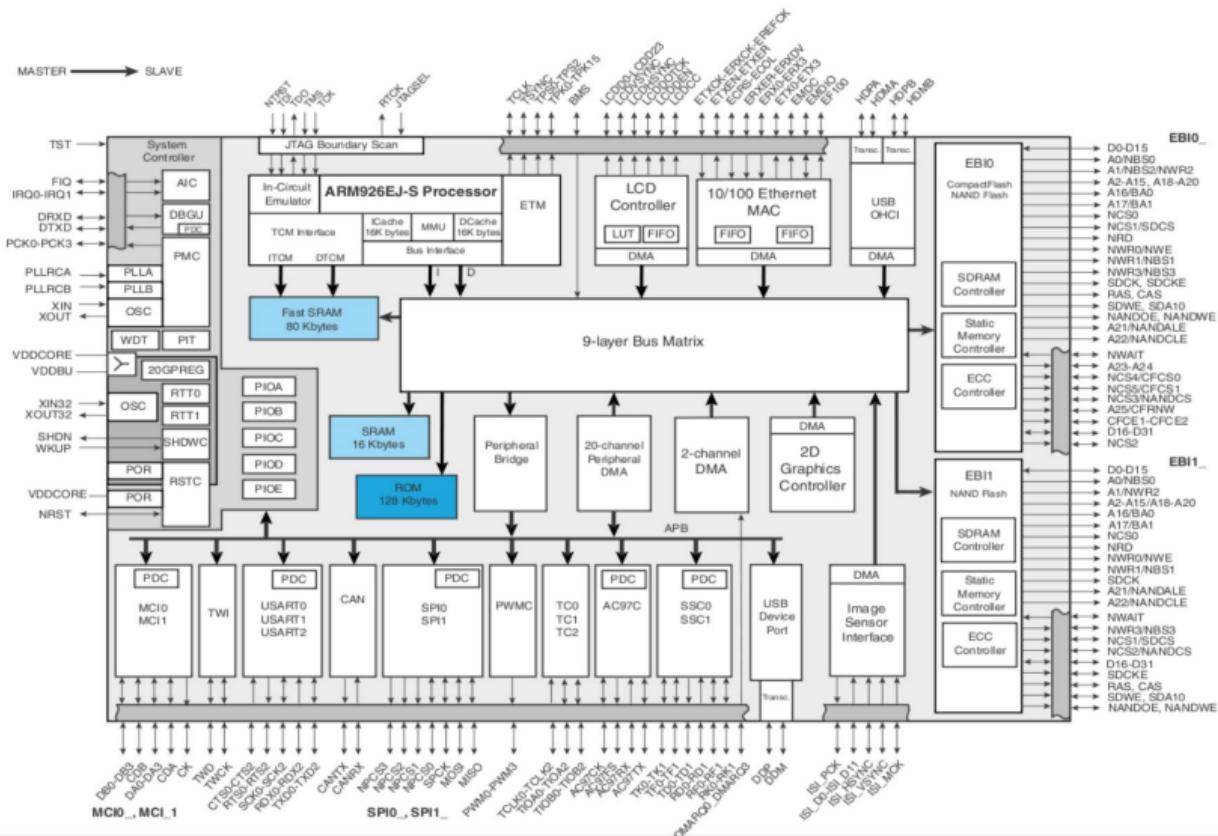


# النظام على الرقاقة ARM



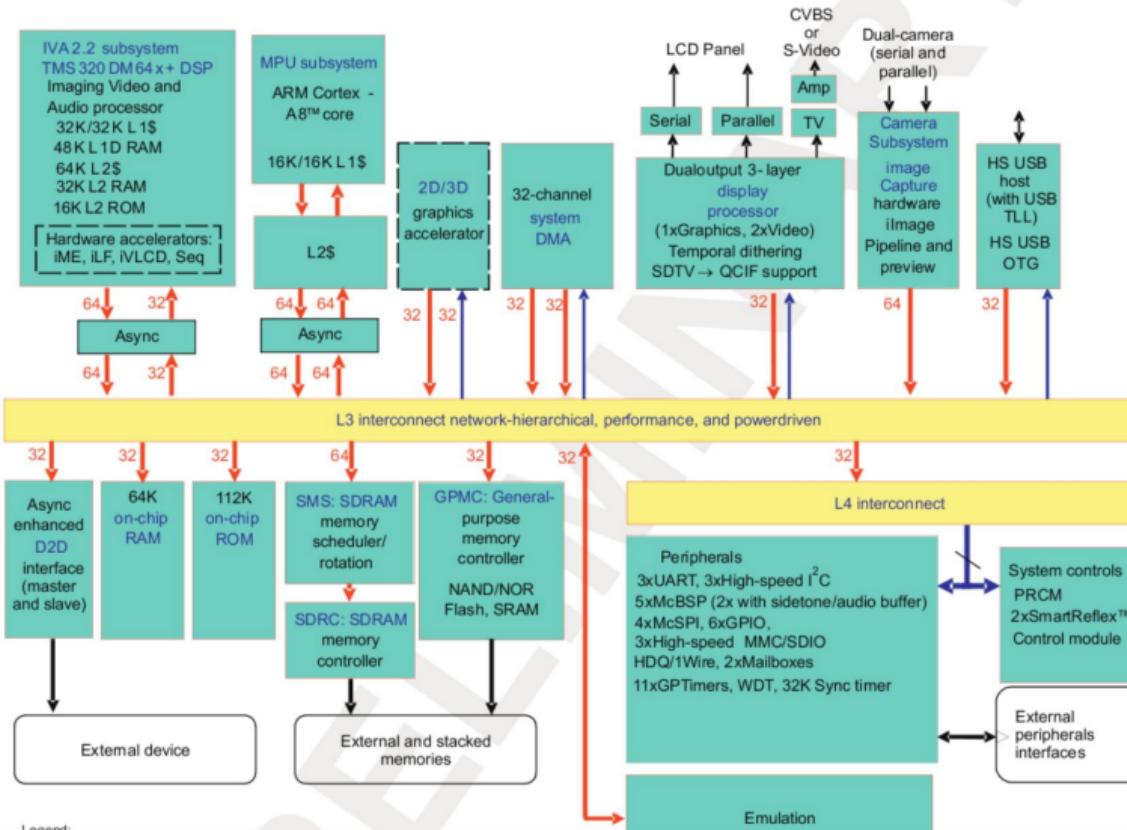


# تميل AT91SAM9263





# تكساس إنسترومنتس OMAP3430





## نوع منصات الأجهزة

■ منصات التقييم من بائع شركة نفط الجنوب. عادةً ما تكون باهظة الثمن، ولكن العديد من الأجهزة الطرفية مدمجة.  
بشكل عام غير مناسب للمنتجات الحقيقية.

■ مكون على الوحدة، وهو عبارة عن لوحة صغيرة تحتوي فقط على وحدة المعالجة المركزية/ذاكرة الوصول العشوائي/ال فلاش وعدد قليل من المكونات الأساسية الأخرى، مع موصلات للوصول إلى جميع الأجهزة الطرفية الأخرى. يمكن استخدامها لبناء المنتجات النهائية بكميات صغيرة إلى متوسطة.

■ منصات تطوير المجتمع، وهي اتجاه جديد لجعل شركة نفط الجنوب (SOC) معينة شائعة ومتاحة بسهولة. وهي جاهزة للاستخدام ومنخفضة التكلفة، ولكنها عادةً ما تحتوي على أجهزة طرفية أقل من منصات التقييم. إلى حد ما، يمكن أيضًا استخدامها ل المنتجات الحقيقية. ■ منصة مخصصة. مخططات لوحات التقييم أو منصات التطوير

أصبحت متاحة مجانًا بشكل متزايد، مما يسهل تطوير منصات مخصصة.



ARM Cortex-A8، TI OMAP3530 (600

(RAM) المكون الموجود في الوحدة ذاكرة الوصول العشوائي (PowerVR SGX, DSP) ٠

سعة 256 ميجابايت

٠ uSD 256 ميجابايت NAND (اختياري) ٠ بلوتوث، واي فاي (اختياري)

115-229 دولاراً أمريكيّاً لوحات التطوير متاحة بسعر 49-229 دولاراً أمريكيّاً، مع

العديد من الأجهزة الطرفية: Ethernet، LCD، TRAU، I2C، SPI، و وما إلى ذلك.

٠ <http://www.gumstix.com>

 Overo FE



Ships with:  
2 x u.fl antennas for Bluetooth and 802.11g  
4 x retaining spacers  
Power supply not included



مكون في Freescale i.MX 51, 800 MHz

الوحدة ذاكرة وصول عشوائي (RAM) سعة

256/512 ميجابايت

512 ميجابايت ناند

FPGA Xilinx Spartan 6A

168

لوحات التطوير متاحة بسعر e 248 مع الأجهزة الطرفية (إيثرنت،

سلسلية، إلخ) وإمكانية الوصول إلى SPI و PWM و I2C و USB و LCD و SD و OIPG و لوحة المفاتيح وما إلى ذلك.

<http://www.armadeus.com>





# كالاو TNY A9G20

كالاو TNY A9G20 مكون Atmel AT91 9G20، 400

في الوحدة

64 ميجابايت من ذاكرة الوصول العشوائي

256 ميجابايت ناند

152

توفر العديد من لوحات التوسيع

<http://www.calao->





# بيجل بورد XM

جيجا هرتز، القشرة TI OMAP DM3730 (1

منصة A8، PowerVR GPU، DSP) □

المجتمع

512 ميجابايت من ذاكرة الوصول العشوائي

microSD، Ethernet □ HDMI، S-Video.

eMMC، الكاميرا، الصوت □ التوسعة: GPIO،

USB، I2C، SPI، LCD، UART،  
إلخ.

دولاراً 149

□ <http://beagleboard.org>





# كرة الثلج

القشرة المزدوجة ، ST Ericsson AP9500  
GPU (مالي)

منصة تنمو المجتمع

1 جيجا رام

microSD بسعة 8-4 جيجابايت، بطاقة eMMC ذاكرة

HDMI، S-Video، الصوت

واي فاي، بلوتوث، إيثرنوت

مقاييس التسارع، مقاييس المغناطيسية،  
الجيرومتر، نظام تحديد المواقع

USB، I2C، SPI، LCD، UART، التوسيع:  
جيبيو، الخ.

ـ 442 هـ إلى 961 هـ

**ARM**

**Cortex**  
Low-Power Leadership from ARM

Dual Cortex A9

NEON™

**mali**



1GByte LP-DDR



4/8 GByte e-MMC

<http://igloocommunity.org>



# بداية سريعة للفريسكالي

Cortex A8 1 جيجا هرتز Freescale I.MX53 (1

منصة تنمو المجتمع

1 جيجا رام

ذاكرة eMMC بسعة 8-جيجابايت، وبطاقة microSD

الصوت LVDS، LCD، VGA، HDMI،

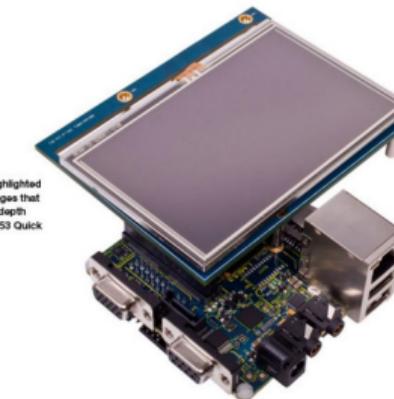
مقاييس التسارع، SD/MMC، microSD،

ساتا، إيثرنت، أوسب

التوسيع: الكاميرا I2C، SPI، SSI، LCD،

149

i.MX53 Quick Start Board



Roll over the highlighted areas in the images that follow for an in-depth look at the LMK53 Quick Start Board



# اتمیل SAM9X35-EK

اتمیل (400 میجا هرتز) AT91SAM9x35

مجلس تقييم الشركة المصنعة

128 ميجابايت من ذاكرة الوصول العشوائي

NAND سعة 256 ميجابايت، وفلاش SPI سعة 4 ميجابايت

إيثرنت، 2 CAN، 2 مضيف/جهاز USB

فتحة uSD، MMC/SD/SDIO

شاشات الكريستال السائل / شاشة تعمل باللمس

زيجي، SPI، I2C، إلخ

590 دولارًا





# معايير اختيار الأجهزة

تأكد من أن الأجهزة التي تخطط لاستخدامها مدعومة بالفعل بواسطة Linux kernel، وتحتوي على أداة تحميل تشغيل مفتوحة المصدر، وخاصة SOC الذي تستهدفه. يعود الحصول على الدعم في الإصدارات الرسمية للمشاريع bootloader (kernel)، أمراً كبيراً.

أفضل: الجودة أفضل، وتتوفر إصدارات جديدة.

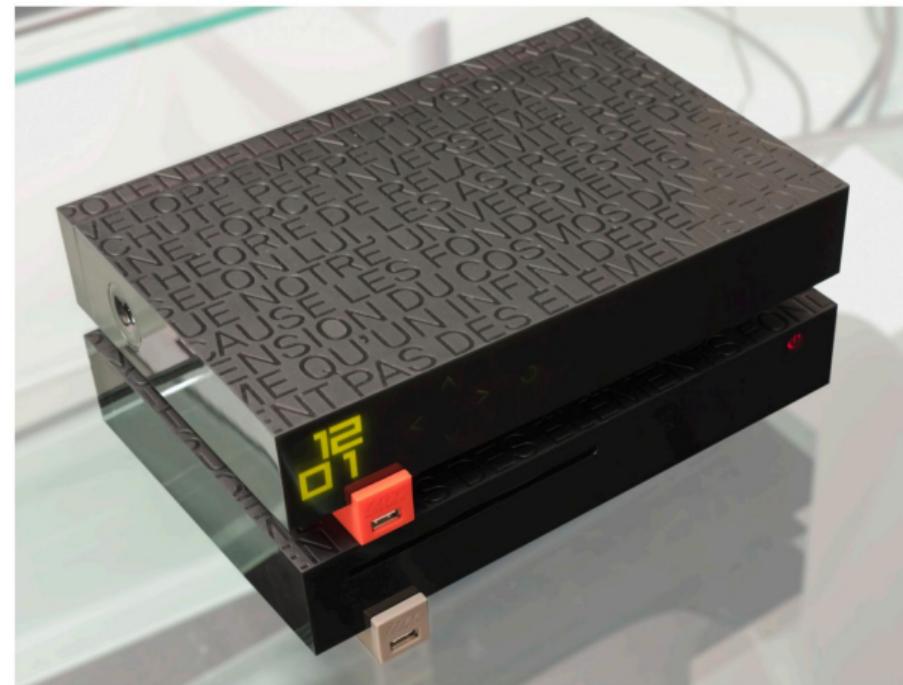
لا يساهم بعض موردي SOC وأو موردي اللوحات بتغييراتهم مرة أخرى إلى Linux kernel. اطلب منهم القيام بذلك، أو استخدم منتجًا آخر إذا استطعت. القياس الجيد هو رؤية الدلتا بين النواة والنوافذ الرسمية.

بين الأجهزة المدعومة بشكل صحيح في نواة Linux الرسمية والأجهزة المدعومة بشكل سيئ، ستكون هناك اختلافات كبيرة في وقت التطوير والتكلفة.

# الأنظمة المدمجة باستخدام Linux



# جهاز المستهلك: صندوق الإنترنيت





# جهاز المستهلك: وحدة التخزين المتصلة بالشبكة





# جهاز المستهلك: التلفاز



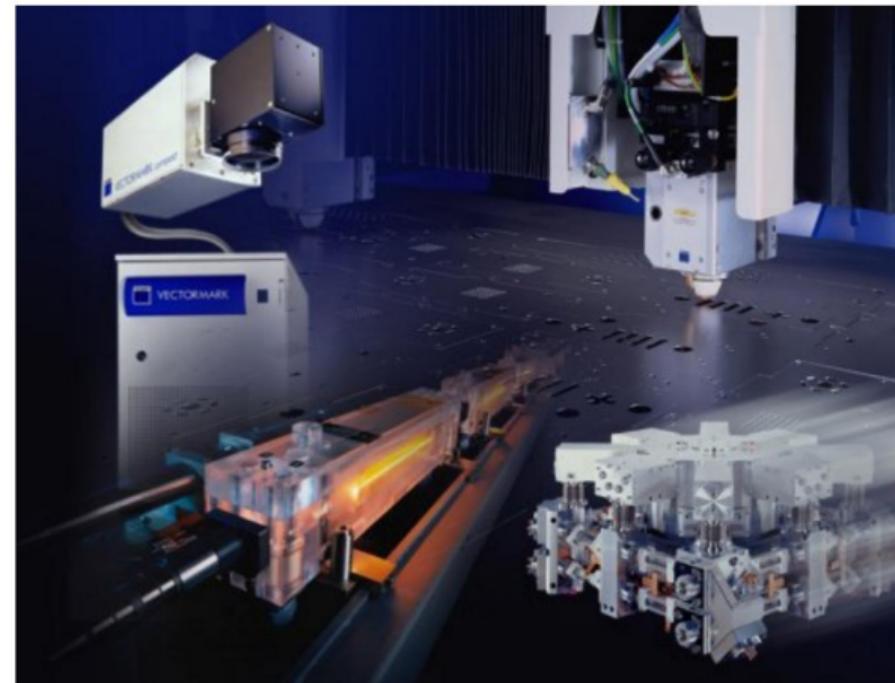


لجهاز الاحترافي: محطة نقاط البيع





# النظام الصناعي: القطع بالليزر





# النظام الصناعي: توربينات الرياح





# النظام الصناعي: حلب البقر





# النظام الصناعي: معدات إزالة الثلوج





## النظام الصناعي: نظام كشف التلوث البحري





## النظام الصناعي: آلة زراعة الكروم



# مكونات مفتوحة المصدر للجزء المضمن أنظمة



## أربعة مكونات رئيسية

يحتاج كل نظام Linux إلى أربعة مكونات رئيسية للعمل:

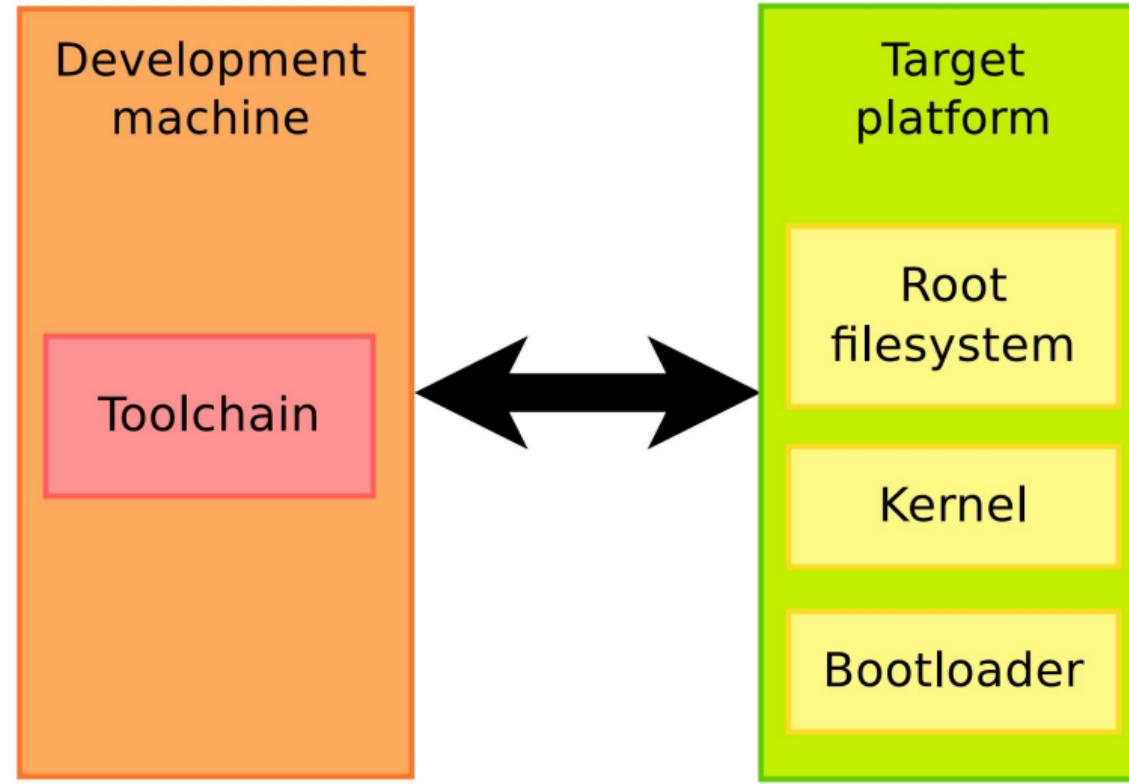
▪ سلسلة الأدوات، التي لا تعمل على النظام الأساسي المستهدف، ولكنها تسمح بإنشاء التعليمات البرمجية للهدف من آلة التطوير.

▪ أداة تحميل التشغيل، المسئولة عن عملية التمهيد الأولية للنظام، وعن تحميل النواة في الذاكرة، Linux Kernel مع جميع برامج تشغيل الأجهزة

▪ نظام الملفات الجذري الذي يحتوي على كافة التطبيقات والمكتبات



# أربعة مكونات رئيسية





## سلسلة الأدوات

عادةً ما تكون سلسلة الأدوات عبارة عن سلسلة أدوات تجميعية مشتركة؛ فهي تعمل على جهاز تطوير وتولد تعليمات برمجية للنظام الأساسي المضمن. لديها المكونات التالية:

أداة مساعدة للمعالجة الثنائية تتضمن مجّمعاً ورابطًا `gcc`، مترجم `C/C++`(والمزيد)، وهو المعيار في المصادر المفتوحة `binutils`.

### عالم

مكتبة لغة `C`، والتي توفر واجهة POSIX لتطبيقات مساحة المستخدم. توفر العديد من مكتبات لغة `C`، `glibc`، `eglibc`، `uClibc`، `glibc`، وأحجام `C` ميزات مختلفة. `gdb`، مصحح الأخطاء، الذي يسمح بتصحيح الأخطاء عن بعد



## الحصول على سلسلة أدوات التجميع المتقطعة

تعتبر سلاسل الأدوات المجمعة مسبقاً هي الحل الأسهل. تحظى سلاسل الأدوات من [CodeSourcery](http://www.codesourcery.com/) بشعبية كبيرة. NG هي أداة تعمل على أتمتة عملية إنشاء سلسلة الأدوات.

<http://www.crosstool-ng.org> يتيح مرونة أكبر من سلاسل الأدوات المترجمة مسبقاً.

عادةً ما تكون أنظمة بناء Linux المضمنة قادرة أيضاً على إنشاء أنظمة خاصة بها سلسلة أدوات التجميع المتقطعة.

تأكد من حصولك على سلسلة أدوات تتوافق مع أجهزتك واحتياجاتك. غالباً ما تكون سلاسل الأدوات التي يقدمها بائعو الأجهزة قديمة وصادئة.



# محمول الإقلاع: المبدأ

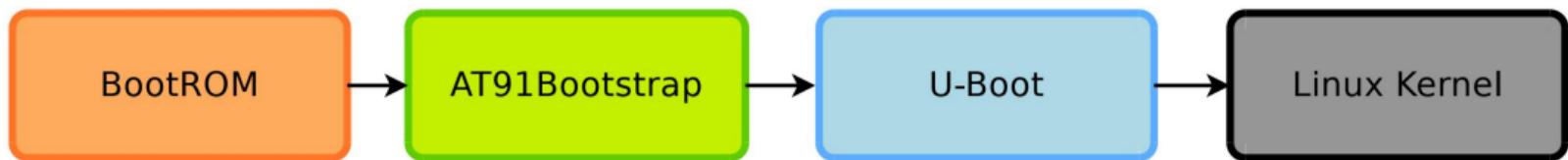
□ يتمثل دور أداة تحميل التشغيل في تهيئة بعض الأجهزة الطرفية الأساسية للأجهزة، وتحميل صورة نواة Linux وتشغيلها. □ عملية التمهيد لأحدث المعالجات المضمنة هي كما يلي:

1. يقوم المعالج بتنفيذ التعليمات البرمجية في ذاكرة القراءة فقط (ROM)، لتحميل أداة تحميل التشغيل للمرحلة الأولى منها المنفذ التسلسلي أو بطاقة SD ، SPI flash.
2. يقوم برنامج تحميل التشغيل للمرحلة الأولى بتهيئة وحدة التحكم في الذاكرة وعدد قليل من الأجهزة الأخرى الأجهزة الطرفية، ويقوم بتحميل أداة تحميل التشغيل للمرحلة الثانية. لا يوجد أي تفاعل ممكن مع أداة تحميل التشغيل للمرحلة الأولى هذه، وعادةً ما يتم توفيره بواسطة بائع وحدة المعالجة المركزية.
3. توفر أداة تحميل التشغيل في المرحلة الثانية المزيد من الميزات: عادةً ما تكون الصدفة مزودة بـ الأوامر، فهو يسمح بمعالجة أجهزة التخزين، والشبكة، وتكون عملية التمهيد، وما إلى ذلك. عادةً ما يكون محمول الإقلاع هذا عاماً ومفتوح المصدر.

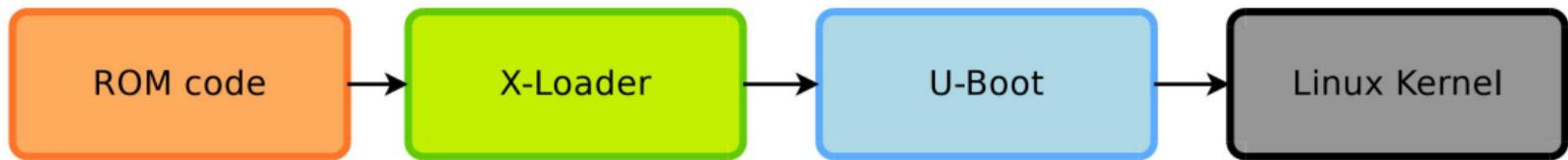


## محمول الإقلاع: المبدأ

### Atmel AT91



### Texas Instruments OMAP3





## برامج تحميل التشغيل مفتوحة المصدر

U-Boot هو المعيار الفعلي في أدوات تحميل التشغيل مفتوحة المصدر، متوفّر على Microblaze، x86، NIOS، SuperH، Sparc. يتوفّر دعم كبير للأجهزة، وعدد كبير من الميزات (الشبكات، USB، SD، وما إلى ذلك)، Barebox، ARM، PowerPC، MIPS، m68k.

أداة تحميل تشغيل أحدث مفتوحة المصدر، بتصميم أنظف من U-Boot <http://www.denx.de/wiki/U-Boot>

ولكن دعم أقل للأجهزة في الوقت الحالي. [www.barebox.org](http://www.barebox.org)

<http://>

المعيار لأجهزة الكمبيوتر x86. [GRUB](#).

<http://www.gnu.org/software/grub/>

تأكد من أن أجهزتك تأتي مع أحد أدوات تحميل التشغيل المعروفة مفتوحة المصدر.



# نواة لينكس

تعد نواة Linux جزءاً أساسياً من النظام. ويوفر ما يلي الرئيسية سمات:

إدارة العمليات إدارة الذاكرة الاتصال بين العمليات، وأجهزة ضبط الوقت برماج تشغيل الأجهزة للأجهزة: الإدخال، والصوت، والشبكة، والتخزين، والرسومات، والبيانات

الاستحواذ، والمؤقتات، وOIPG، وما إلى ذلك. أنظمة الملفات الشبكات إدارة الطاقة تحتوي نواة Linux على آلاف الخيارات لتمكين الميزات أو تعطيلها بشكل انتقائي

اعتماداً على احتياجات النظام.

بموجب ترخيص.

<http://www.kernel.org>



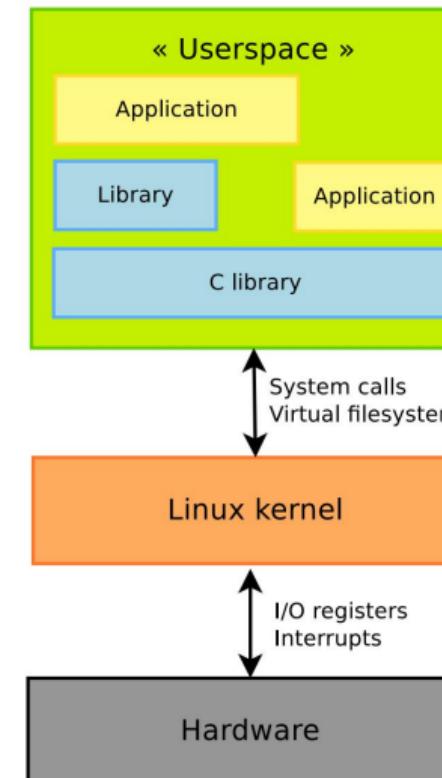
## النواة مقابل مساحة المستخدم

تعمل نواة Linux في الوضع المميز. يمكنه الوصول والتحكم في المعدات. يتمثل دورها في مضاعفة الموارد المتاحة وتوفير واجهات متماسكة ومتسقة لمساحة المستخدمين. مساحة المستخدم هي مجموعة التطبيقات والمكتبات التي تعمل على النظام. هم

العمل في وضع غير مميز. ويجب عليهم المرور عبر النواة للوصول إلى موارد النظام، بما في ذلك الأجهزة. توفر النواة العزلة بين التطبيقات.



# النواة مقابل مساحة المستخدم





# نواة النظام الأساسي الخاص بك

تتميز نواة Linux بسهولة النقل عبر البنية. لذلك، يتم تقسيم الكود إلى عدة مستويات:

كود عام لجميع البنية: ARM, x86, PowerPC, Freescale i.MX, الخ. كود خاص بنية ما كود خاص بنظام على شريحة: Atmel AT91, TI OMAP3.

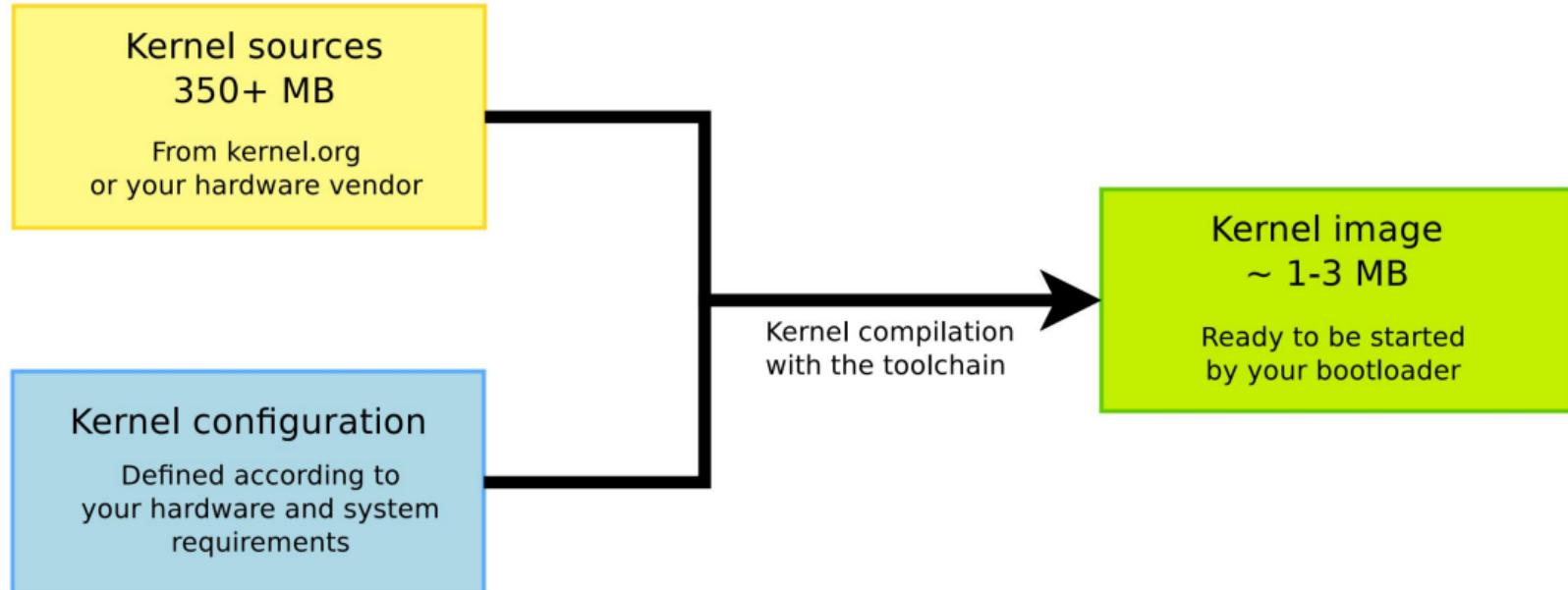
إلخ.

رمز خاص باللوحة، والذي يتكون من شركة نفط الجنوب معينة مع أجهزة طرفية إضافية على اللوحة

إذا كانت شركة SOC الخاصة بك مدعومة جيداً، فإن الجزء الوحيد من التعليمات البرمجية الذي يحتاج إلى تعديل هو الجزء الخاص باللوحة. إلا إذا كنت بحاجة إلى برامج تشغيل إضافية للأجهزة.



# تجميع النواة





# برمجة النواة

تختلف بيئه البرمجة داخل نواة Linux تماماً عن تلك الموجودة في مساحة المستخدمين: واجهة برمجة تطبيقات مختلفة، وقيود مختلفة، وآليات مختلفة لا توجد مكتبة C قياسية، تتوفّر واجهة برمجة تطبيقات محددة لا توجد حماية للذاكرة

يلزم عادةً البرمجة في النواة من أجل تكييف النواة مع لوحة معينة كتابة برامج تشغيل الأجهزة

هناك العديد من الموارد الخاصة ببرمجة النواة:

برامج تشغيل أجهزة Linux، كتاب من O'Reilly Linux، برامج تشغيل أجهزة Linux الأساسية، كتاب من Bootlin Materials Training على برمجة Kernel من Prentice Hall



# نظام الملفات الجذر

في نظام Linux يتم تخزين التطبيقات والمكتبات والتكونيات والبيانات في ملفات في نظام الملفات يتم استخدام تسلسل هرمي عالمي واحد من الدلائل والملفات لتمثيل جميع الملفات الموجودة في

النظام، بغض النظر عن وسيلة التخزين أو الموقع.

يتم تركيب نظام ملفات معين يسمى نظام الملفات الجذر في جذر هذا النظام تسلسل.

يحتوي نظام الملفات الجذر هذا عادةً على كافة الملفات الازمة لكي يعمل النظام على وجه صحيح.



# أنظمة الملفات

يدعم Linux kernel نطاقاً واسعاً من أنواع أنظمة الملفات:

Linux هي أنواع أنظمة الملفات الافتراضية لنظام التشغيل. وهي قابلة للاستخدام على ext2, ext3, [ext4](#) أجهزة الحظر.

jffs2, ubifs هي أنظمة الملفات التي يمكن استخدامها على أجهزة الفلاش (ومضات. SPI و NOR و NAND) لاحظ أن بطاقات SD/MMC أو مفاتيح USB ليست أجهزة فلاش، ولكنها تحظر الأجهزة. squashfs هو نظام ملفات مضغوط للغاية للقراءة فقط، وهو مناسب لجميع ملفات النظام التي لا تتغير أبداً. vfat, nfts، [أنظمة ملفات Windows العالمية](#)، مدعومة أيضاً من أجل التوافق cifs [nfs](#). هما أهم أنظمة ملفات الشبكة المدعومة



## الحد الأدنى من محتويات نظام الملفات الجذر

□ تطبيق init، وهو التطبيق الأول الذي تبدأ النواة عند تشغيل النظام. init هو المسؤول عن بدء تشغيل الأصداف وخدمات النظام والتطبيقات.

□ الغلاف والأدوات المرتبطة به للتفاعل مع النظام. عادةً ما تعمل القشرة عبر منفذ تسلسلي. مكتبة C، التي تطبق واجهة POSIX، والتي تستخدمها جميع التطبيقات. مجموعة من ملفات الجهاز. هذه ملفات خاصة تسمح للتطبيقات بتنفيذ العمليات على الأجهزة التي تديرها النواة. تسلسل هرمي نموذجي لنظام Unix، مع bin و lib و dev و /bin و /lib و sbin و usr و proc و sysfs و proc.



# Busybox

في نظام Linux العادي، يتم توزيع جميع المكونات الأساسية في مشاريع مختلفة ولم يتم تنفيذها مع وضع القيود المضمنة في الاعتبار. يوفر Busybox مجموعة عالية التكثيف لجميع الأوامر الأساسية المطلوبة في نظام Linux: cp, mv, sh, wget, grep, init, modprobe, udhcpc, httpd.

يتم تجميع كل هذه الأوامر في ثنائي واحد، والأوامر عبارة عن روابط رمزية لهذا الثنائي. فعالة للغاية في استخدام المساحة على الأنظمة التي يتم فيها استخدام التجميع الثابت.

بموجب ترخيص GPLv2

<http://www.busybox.net>



# تطوير التطبيقات في النظام الأساسي

مع Busybox ومكتبة C فقط، لدينا نظام Linux مدمج يعمل بكامل طاقته نظام

تطبق مكتبة C واجهة POSIX المعروفة، والتي توفر واجهة برمجة التطبيقات (API) للتحكم في العمليات، والإشارات، وعمليات الملفات والأجهزة، والمؤقتات، والأنابيب، ومكتبة C القياسية (وظائف السلسلة، وما إلى ذلك)، وإدارة الذاكرة، والإشارات، والذاكرة المشتركة وإدارة الموارد والشبكات وما إلى ذلك.

هذه بيئه مريحة جدًا بالفعل لتطوير التطبيقات بلغة C أو C++ التي يمكنها التفاعل مع الأجهزة، وإجراء بعض العمليات الحسابية، والتفاعل مع الأجهزة.

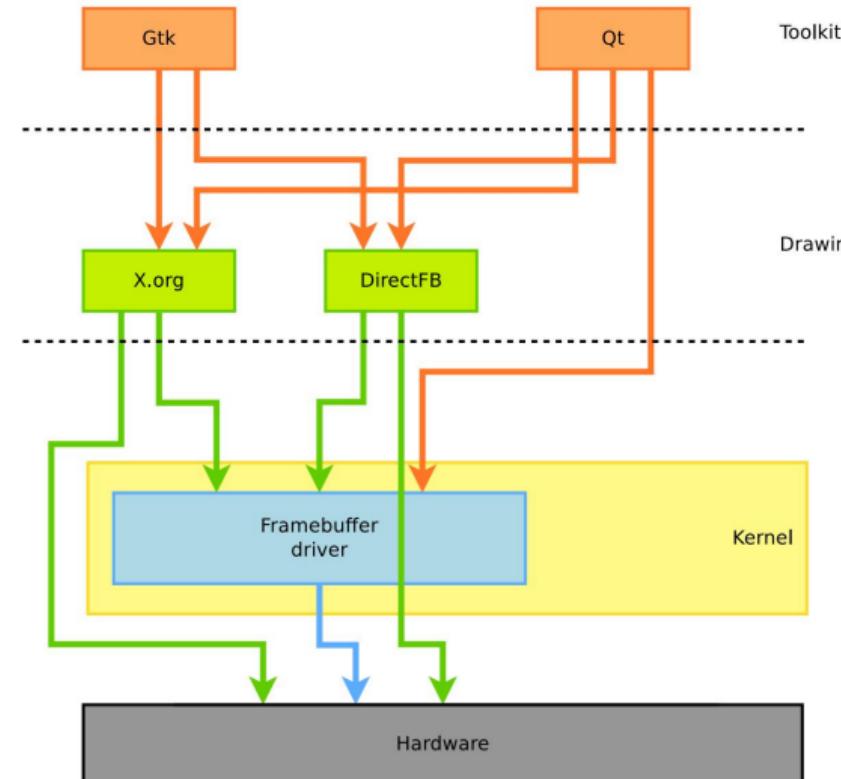
Busybox، يمكنك بسهولة توفير خادم ويب لمراقبة بفضل نظام.



## التفاعل مع الأجهزة

هناك طرق مختلفة للتفاعل مع الأجهزة:

- تحتوي النواة على برنامج تشغيل للجهاز، وفي هذه الحالة يمكن الوصول إليه إما من خلال ملف جهاز في `/dev` باستخدام واجهة برمجة تطبيقات الملف القياسي، أو من خلال ملف نصي في نظام ملفات `sysfs` أو من خلال واجهة برمجة تطبيقات الشبكة.
- يمكن لتطبيقات مساحة المستخدم التي تعمل كجذر استخدام الجهاز الخاص `/dev/mem` للوصول مباشرة إلى الذاكرة الفعلية، أو استخدام إطار عمل `IO` للنواة بشكل أفضل.
- بالنسبة لحافلات ، `I2C` و `SPI` يوجد `/dev/i2cdevX` و `/dev/spidevX` خاص لإرسال/استقبال الرسائل على الناقل، دون الحاجة إلى برنامج تشغيل جهاز `kernel`. بالنسبة لـ `USB` ، تتيح مكتبة `libusb` تشغيل أجهزة `USB` لمساحة المستخدم.
- يعتمد اختيار أحد الحلول على الآخر على نوع الجهاز، وال الحاجة إلى التفاعل مع أنظمة `kernel` الفرعية الموجودة.





# رسم

□ X.org, <http://www.x.org>

□ نهج العميل/الخادم. يقوم الخادم بإدارة الأجهزة (الرسومات والمدخلات)، ويوفر بروتوكولاً للعملاء. العملاء هم جميع التطبيقات التي ترغب في رسم الأشياء واستقبال أحداث الإدخال.

□ الحل المستخدم على جميع أنظمة Linux السطح المكتب، يسمح بالتوافق مع الأنظمة الحالية مكتبات، تسمى مجموعات الأدوات الرسمية، لتطوير التطبيقات الرسومية. □ القدرة على استخدام التسريع ثنائي الأبعاد وثلاثي الأبعاد، بشرط وجود برامج تشغيل خاصة بالأجهزة مناج.

□ دايركت إف بي، □ مكتبة فوق برنامج <http://www.directfb.org> تشغيل مخزن إطارات kernel، مما يسمح بالتعامل مع أحداث الإدخال. □ أخف وزناً من خادم X ولكن بدون توافق بروتوكول X11، و مع ميزات أقل قليلاً.

□ يمكن استخدام واجهة برمجة التطبيقات (API) مباشرةً لتطوير التطبيقات، ولكن يمكن أيضاً استخدام مجموعات الأدوات فوقها.



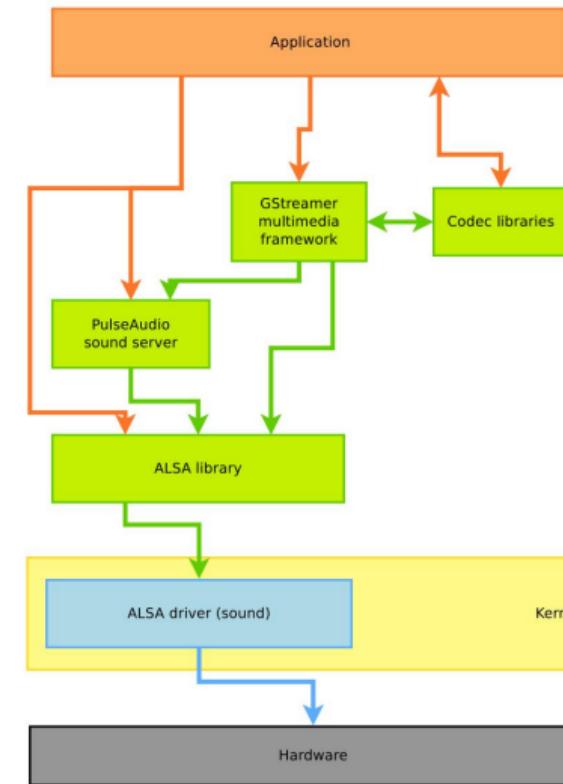
توفر مجموعات الأدوات واجهة برمجة تطبيقات عالية المستوى لإنشاء واجهات رسومية تحتوي على النوافذ والأزرار ومدخلات النص والقوائم المنسدلة ومربعات الاختيار واللوحة القماشية وما إلى ذلك.

### [كيو تي،](http://qt.nokia.com)

إطار تطوير كامل بلغة C++ مع إدارة الأحداث، والشبكات، والمؤقتات، والخيوط، ... XML والرسومات يستخدم كأساس لبيئة سطح المكتب KDE على أنظمة Linux ولكنها يحظى أيضاً بشعبية كبيرة على الأنظمة المدمجة. يعمل على X.org أو على مخزن الإطارات DirectFB المؤقت أو أعلى

جي تي كيه، [أيضاً إطار](http://www.gtk.org) تطوير كامل بلغة C. يستخدم كأساس لبيئة سطح مكتب جنوم على Linux.

ربما أقل شعبية على الأنظمة المدمجة. يعمل على X.org وعلى DirectFB





[www.alsa-project.org](http://www.alsa-project.org) هو النظام الفرعي **ALSA** لأجهزة الصوت kernel

<http://>

**alsa-lib** هي مكتبة مساحة المستخدم التي تسمح للتطبيقات باستخدام برامج تشغيل ALSA. إنها مكتبة منخفضة المستوى إلى حد ما. **PulseAudio** هو خادم صوت. يدير تدفقات صوتية متعددة، ويمكن ضبطها

حجمها بشكل مستقل، وإعادة توجيهها ديناميكياً، وما إلى ذلك. **GStreamer** هو إطار عمل متعدد الوسائط. مع مجموعة [من](#) المدخلات والمخرجات،

المكونات الإضافية لفك التشفير والتشفير، يمكن للمرء إنشاء خطوط أنابيب مخصصة لتشفيروفك تشفير وعرض تدفقات الفيديو أو الصوت <http://www.gstreamer.org>



يُستخدم `X.org` أو مخزن الإطارات عادةً كأجهزة إدخال `Video4Linux kernel` لـ `V4L` يدعم النظام الفرعي لـ `video4linux2`.  
الفيديو وبعض أجهزة إخراج الفيديو التي تقوم بالتركيب.

فـ `GStreamer` تشغيل الفيديو وتشغيله لإطار الوسائط المتعددة



# الشبكات

نظام Linux معروف بقدراته الشبكية:

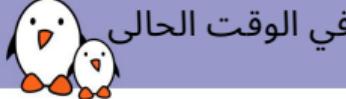
دعم أجهزة WiFi وBluetooth و Ethernet NAC kernel والموجودة في تكوين مساحة المستخدمين المرتبطة بها

خوادم الويب: httpd في htpd، lighttpd، aob و Busybox، وما إلى ذلك. خوادم OpenSSH و

جدار الحماية: SSH: Dropbear، GPRS/Modem: pppd و SSL التشفير / VPN: OpenSSL، OpenVPN و OpenVPN

Netfilter مفتوح، CAN البروتوكولات الصناعية: iptables في النواة، وأمر

SNMP، Modbus TCP، NTP، البريد، خادم البريد، الخ. وأيضاً: SNMP، Modbus TCP،



### ثلاثة حلول للوقت الحقيقي في Linux

استخدم النواة القياسية. لقد تم تحسينه على مر السنين لتطبيقات الوقت الفعلي (استباقي kernel وأجهزة ضبط الوقت عالية الدقة، وراثة الأولوية، ودعم واجهة برمجة التطبيقات POSIX في الوقت الحقيقي، وما إلى ذلك)

استخدم تصحيحات PREEMPT\_RT . هذه عبارة عن تصحيحات ضد نواة Linux تعمل على تحسين سلوكها في الوقت الفعلي (إجراء وقائي أكثر اكتمالاً، ومعالجات المقاطعة في سلاسل الرسائل، وما إلى ذلك) <http://rt.wiki.kernel.org>

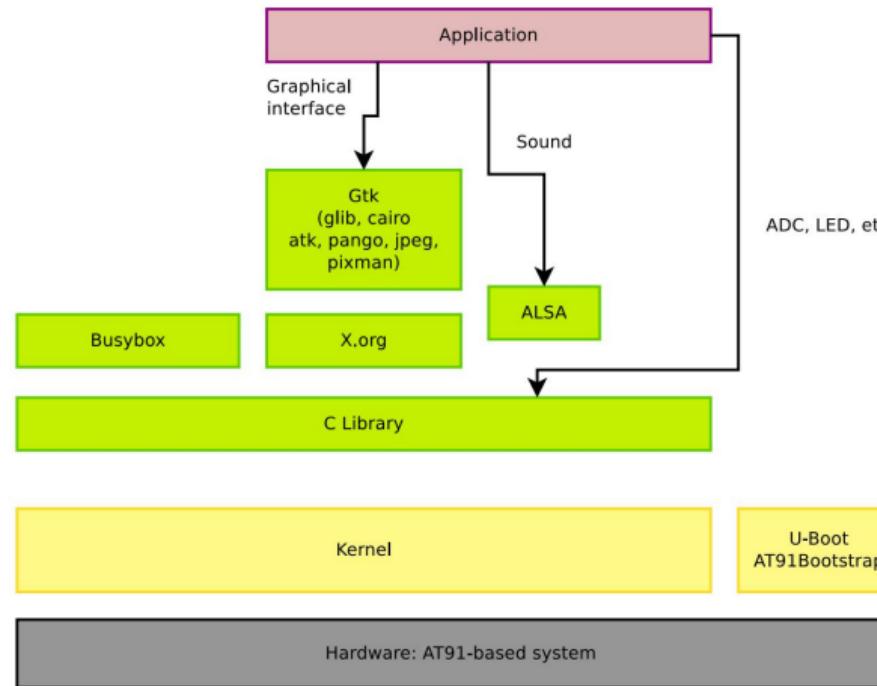
استخدم أحد حلول النواة المشتركة: Xenomai أو RTAI. المهام في الوقت الحقيقي هي

يتم جدولتها بواسطة نواة مخصصة في الوقت الفعلي، ويعمل Linux كمهمة ذات أولوية منخفضة. <http://www.xenomai.org> <http://www.rtai.org>



# مثال: دراجة التمرين

نظام يوضح أداء وتقديم مستخدم الدراجة، مع واجهة رسومية واتصال بالجهاز.



# عملية تطوير Linux المضمنة



# ثلاث خطوات رئيسية

تطوير حزمة دعم اللوحة  $\square$  تكامل النظام  $\square$  تطوير  
التطبيقات



# حزمة دعم مجلس الإدارة (1)

■ يعتبر BSP أساس النظام الذي يعتمد بشكل كبير على الأجهزة:

سلسلة الأدوات ومحمل الإقلاع و ■ هل إصدارات محمّل الإقلاع و Linux kernel حديثة بما فيه

الكافية؟ مع قديمة جدا

الإصدارات التي تفتقد الميزات، والأهم من ذلك، أنك تفقد كل دعم المجتمع.

■ هل يتوفّر الدعم في الإصدارات الرسمية الرئيسية لمحمل التشغيل و النواة؟ وهذا هو الحل الأمثل، فهو يضمن لك الاستفادة من الإصدارات المحدثة.

■ إذا تم توفيرها من قبل بائع الأجهزة، ما حجم الدلتا في الإصدار الرسمي؟ عندما تكون الدلتا كبيرة جدًا، فمن الصعب الترقية إلى الإصدارات الأحدث ■ سيتم حظرك.

■ هل هناك برامج تشغيل ثانوية؟ سوف يمنعونك من ترقية النواة.



## حزمة دعم مجلس الإدارة (2)

تعتبر مكونات حزمة دعم اللوحة باللغة الأهمية. لا تعتمد على القديم الإصدارات مع تعديلات ضخمة من بائع الأجهزة.

تأكد من أنك منفصل: الإصدار الرسمي الذي بدأ التطوير منه، وتعديلات بائع الأجهزة، وتعديلاته الخاصة.



# نظام التكامل

□ دمج جميع المكونات مفتوحة المصدر الازمة لنظامك ومكتباتك وتطبيقاتك المخصصة.

□ يتضمن ذلك تكوين العديد من المكونات وتجميعها بشكل مشترك في بعض الأحيان تبعيات معقدة وأو آلية تجميع غير تافهة، خاصة في سياق التجميع المتبادل.

□ لا تفعل ذلك يدوياً، ولا تعيد اختراع العجلة بكتابة تصميمك الخاص نظام.



# تكامل النظام: الحلول

توفر بعض التوزيعات الثنائية، مثل Debian، للبنيات المدمجة، MIPS و PowerPC، وما إلى ذلك). المزايا: تم تجميع كل شيء بالفعل، وسهولة إضافة/إزالة المكونات، ولطيف

نظام إدارة الحزم. العيوب: ليس هناك قدر كبير من التحكم في تكوين المكون، ولا يوجد كود

تم تحسينه بالضرورة لمنصة أجهزتك، وهو نظام كبير إلى حد ما يحتاج غالباً إلى تجريبه، ولا توجد آلية لإعادة إنتاج البنية. أنظمة بناء Linux المضمنة، التي تبني من المصدر جميع عناصر نظام Linux وتقوم بإنشاء صورة نظام الملفات الجذر (والمزيد)

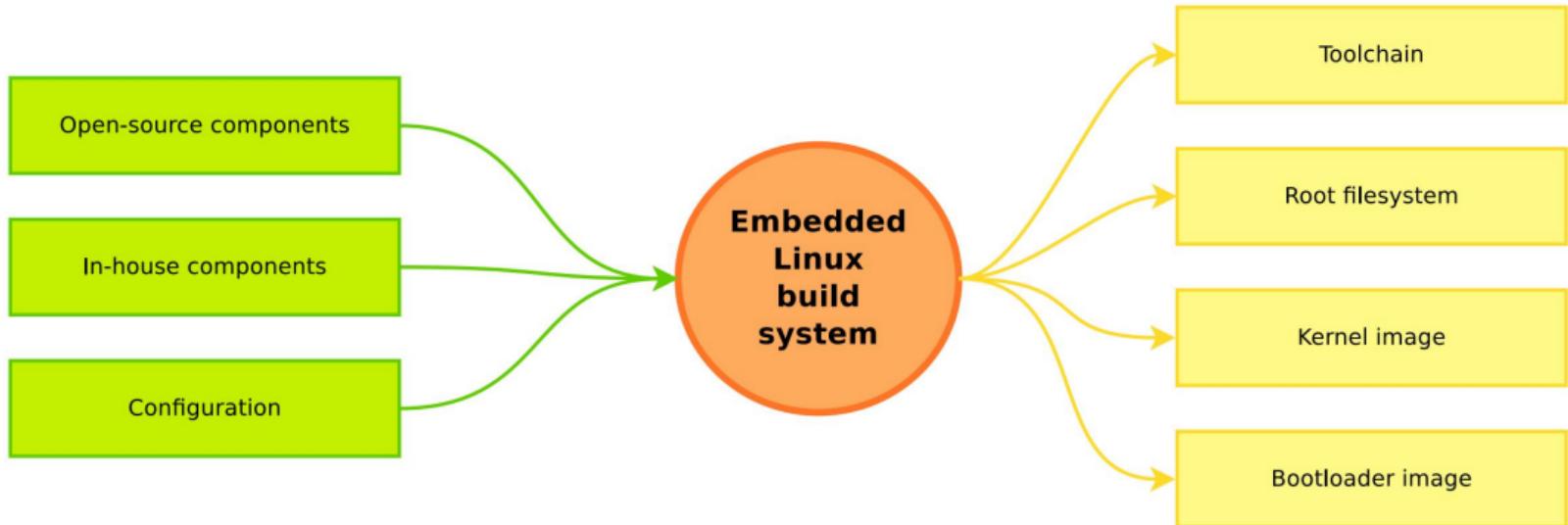
المزايا: تحكم كبير في تكوين النظام والمكونات، آلية آلية لإعادة إنتاج البناء، ونظام خفيف الوزن.

العيوب: الحاجة إلى تعلم أداة جديدة، وأوقات تجميع طويلة.

لا تستخدم نظام ملفات العرض التوضيحي الجذري الخاص ببائع الأجهزة كنقطة بداية: ليس لديك طريقة لإعادة إنتاج الإصدار ولا تتحكم في مكوناته.



# نظام بناء Linux المدمج: المبدأ



بعض أنظمة البناء مفتوحة المصدر الشائعة:

- Yocto, <http://www.yoctoproject.org>، بيلدروت،
- OpenEmbedded, <http://www.openembedded.org>
- OpenBricks، OpenWRT، وغيرها: <http://www.buildroot.org>
- PTXdist، إلخ.



# نظام بناء Linux المضمن: مثال على Buildroot

تتيح واجهة التكوين المشابهة لواجهة kernel تحديد جميع جوانب النظام: بنية وحدة المعالجة المركزية، ومكونات البرامج المطلوبة، ونوع نظام الملفات بصورة نظام الملفات الجذر، وإصدار kernel وتكتوينه، وإصدار أداة تحميل التشغيل وتكتوينه، وما إلى ذلك.

بمجرد الانتهاء من التهيئة، يتولى Buildroot جميع الخطوات: تنزيل جميع المكونات واستخراجها وتصحيحها وتكتوينها وترجمتها وتثبيتها بالترتيب الصحيح.

أكثر من 600 مكون برمجي متاح بالفعل سهل الاستخدام، وإصدارات مستقرة منتظمة، ومجتمع نشط من السهل جدًا إضافة مكونات برمجية جديدة، سواء مفتوحة المصدر أو داخلية العيوب: غالباً ما تكون هناك حاجة إلى عمليات إعادة بناء كاملة، ولا يوجد نظام لإدارة الحزم على

هدف.



# اختيار المكونات مفتوحة المصدر

هناك عدة معايير يجب مراعاتها عند اختيار مكون مفتوح المصدر:

جودة المكونات. هل هناك وثائق كافية؟ هل المكون مستخدم على نطاق واسع (يعد وجوده في أنظمة بناء Linux المضمونة مؤشراً جيداً) حيوية المجتمع. هل لا يزال المكون قيد التطوير بنشاط؟ هل

هل يستجيب المجتمع لتقارير الأخطاء والأسئلة؟ متى كان آخر إصدار مستقر؟ هل هناك نشاط منتظم في نظام مراقبة المراجعة؟

التراخيص. هل يتطابق ترخيص المكون مع متطلبات المنتج؟

المتطلبات الفنية. هل يقدم الميزات التي تحتاجها؟ هل هو مناسب في من حيث التخزين ووحدة المعالجة المركزية واستهلاك الذاكرة؟ هل المكونات المتوفرة بالفعل في أنظمة بناء Linux المضمونة غالباً ما تكون نقطة بداية جيدة.



# تطوير التطبيقات

- إن تطوير التطبيقات في أنظمة Linux المضمنة هو نفسه تماماً تطوير التطبيقات في نظام Linux عادي: نفس أدوات التطوير ونفس المكتبات.
- اللغة القياسية في Linux هي C، والنظام بأكمله والعديد من المكتبات مكتوبة بهذه اللغة. تُستخدم لغة C++ أيضاً على نطاق واسع. توفر العديد من اللغات المترجمة: PHP و Perl و Python و Lua لا تهمل فائدتها للأجزاء غير الحساسة للأداء، لأنها تسمح عادةً بالتطوير بشكل أسرع.



# تصحيح أخطاء التطبيق والنظام

## □ التصحيح المتبادل مع gdb وgdbserver

□ يعمل `gdbserver` على الهدف، ويتحكم في تنفيذ التطبيق لتصحيح الأخطاء □ يعمل `gdb` وربما إحدى واجهاته الرسمية على جهاز [التطوير](#)، ويتحدث إلى `gdbserver` من خلال Ethernet أو المنفذ التسلسلي.

□ تتبع مكالمات النظام باستخدام `strace` وتتبع مكالمات المكتبة باستخدام `ltrace` □ تتبع على مستوى [oprofile.sourceforge.net](http://oprofile.sourceforge.net) □ TTng على مستوى النظام باستخدام [TTng](http://lttng.org), <http://lttng.org> □ OProfile, <http://oprofile.sourceforge.net>



□ تمنح تراخيص البرمجيات الحرة للجميع مجموعة الأربعة الأساسية  
الحرفيات، ولكن لديهم أيضا بعض المتطلبات. □ تنقسم إلى فئتين رئيسيتين:

□ تراخيص الحقوق المتروكة ، التي تتطلب توزيع الإصدارات المعدلة تحتها  
رخصة.

□ التراخيص غير المتروكة الحقوق، والتي لا تتطلب توزيع الإصدارات المعدلة بموجبها  
نفس التراخيص: يمكن الاحتفاظ بها كملكية. □ لدى مؤسسة البرمجيات الحرة ومبادرة المصدر المفتوح قائمة

بـ

التراخيص مع رأيهم فيها:  
<http://www.gnu.org/licenses/license-list.html> <http://opensource.org/licenses/index.html>  
<http://www.gnu.org/>



## الرخصة العامة

يغطي الترخيص الرئيسي للحقوق المتروكة 50% من مشاريع البرمجيات المجانية على سبيل المثال: Linux Kernel وBusybox وما إلى ذلك. يتطلب إصدار أعمال مشتقة بموجب نفس الترخيص، بما في ذلك Boot وـ Busybox.

## التطبيقات التي تعتمد على مكتبة مرخصة بموجب GPL.

يتطلب الترخيص منك شحن كود المصدر الكامل لمكونات GPL مع منتجك، بما في ذلك تعديلاتك على هذه المكونات، ويجب الاحتفاظ بالاسناد.

لا حاجة لتوزيع الكود المصدرى قبل توزيع المنتج. تم تنفيذ الترخيص بالفعل عدة مرات في المحكمة. على الرغم من أن النواة مملوكة لـ GPL، فليس من الضروري أن تكون كل مساحة المستخدمين خاضعة لـ GPL



# وحدات GPL والنواة

بما أن نواة Linux خاضعة لرخصة GPL، فيجب إصدار التعديلات عليها ضمن جي بي إل.

يمكن كتابة بعض أكواد kernel كوحدات نمطية، والتي يمكن تحميلها/تفريغها ديناميكيًا في وقت التشغيل.

أم لا.

هناك جدول حول ما إذا كانت وحدات النواة هي أعمال مشتقة من النواة

لا توجد إجابة نهائية على السؤال، تختلف الآراء. بشكل عام، رأى المجتمع، بما في ذلك العديد من مطوري kernel، أن وحدات kernel الخاصة سينته. "نحن،

مطورو نواة Linux الموقون أدناه، يعتبر أي نظام Linux مغلق المصدر

وحدة النواة أو برنامج التشغيل ضار وغير مرغوب فيه.",

من الأسهل إنشاء كود النواة الخاص بك بـ GPL وترك أجزاء القيمة المضافة فيه

مساحة المستخدم



LGPL

## الرخصة العامة الصفرى

ترخيص أضعف للحقوق المتروكة، يستخدم في العديد من المكتبات على سبيل المثال: Gtk، Qt، alsalib ومعظم المكتبات يتطلب إصدار الأعمال المشتقة بموجب نفس الترخيص، ولكنها غير مجانية

يمكن ربط التطبيقات بمكتبة LGPL ، طالما يمكن استبدال المكتبة (يتم استخدام الارتباط الديناميكي بشكل عام)

يتطلب الترخيص منك، أيضًا إرسال كود المصدر الكامل لـ LGPL المكونات، بما في ذلك تعديلاتك.

احذر من وجود بعض المكتبات تحت رخصة MySQL و مكتبة readline و مكتبة GPL وما إلى ذلك.



# التراخيص غير المتروكة

يتم استخدام العديد من التراخيص غير المتروكة على نطاق واسع. وهي لا تتطلب توزيع الأعمال المشتقة بموجب نفس الترخيص، ولكن يجب أن يظل المشروع الأصلي مقيداً.

## [رخصة بي إس دي](#)

## [ترخيص أباتشي](#)

[رخصة معهد ماساتشوستس للتكنولوجيا](#)

## [الرخصة الفنية](#)

## [رخصة X11](#)

لا يُطلب منك توزيع التعليمات البرمجية المصدر لهذه المكونات ويمكنك دمج التعليمات البرمجية من هذه المكونات في تطبيقاتك الخاصة.



# ترخيص الممارسات الجيدة

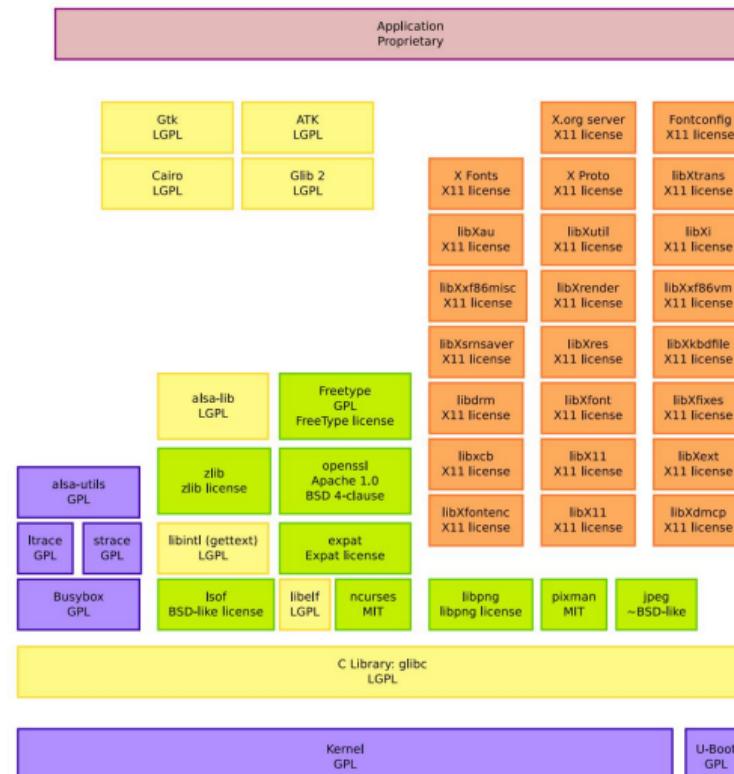
- احتفظ بقائمة دقيقة وكاملة لجميع المكونات التي تستخدمها في جهازك المنتج، مع الترخيص الخاص بكل منها
- تأكد من أن ترخيص المكون يطابق متطلباتك من قبل مستندًا في كل تطوير منتجك عليه.
- احتفظ بتغييراتك منفصلة عن الإصدار الأصلي للمكونات. وهذا يسمح بإجراء ترقيات أسهل، ولكن أيضًا مع احترام التراخيص التي تريده تحديد التغييرات من الإصدار الأصلي بوضوح.

الطرق الممكنة: كومة من التصحيحات مع اللحاف، أو الفروع في نظام مراقبة المراجعة.

- لا تقم بنسخ/لصق كود GPL/LGPL في أجزاء نظامك التي يجب أن تكون كذلك تظل ملكية.
- التزم بالتراخيص بمجرد بدء شحن منتجك.



# تحليل الترخيص





# العمل على نظام Linux المدمج

استخدم Linux في محطة التطوير تم تطوير جميع أدوات المجتمع بواسطة مطوري Linux، باستخدام Linux كنظام تشغيل سطح المكتب الخاص بهم. ستؤدي محاولة استخدام نظام التشغيل Windows أو Mac OS لتطوير Linux للمضمن إلى حدوث صعوبات . على النظام المضمن هو نفس Linux على سطح المكتب: العديد من الأشياء الجيدة

مهندسو Linux المضمنون هم مجرد مستخدمي Linuxمنذ فترة طويلة لديك عميل بريد إلكتروني جيد مطلوب للتفاعل مع المجتمع يجب أن يدعم عميل البريد الإلكتروني الخاص بك الترابط وسائل البريد الإلكتروني النصية فقط (بدون HTML والصحيح

يلف

لا تستخدم Outlook أو Evolution أو KMail بل تستخدم بدلاً من ذلك Thunderbird أو Lotus Notes، أو مخالب البريد، الخ.



# العمل على نظام Linux المدمج

تمتع باتصال جيد بالشبكة وغير مفلتر

تحتاج إلى الوصول إلى العديد من الموارد: مستودعات Git و SVN و قنواتIRC

المناقشة مع المجتمع، والقواعد البريدية

يعد وجود خادم SMTP قياسي مفيداً أيضاً لإرسال التصحيحات. التحكم في مكونات النظام لديك وإنشاء الإجراء واستخدام التحكم في المراجعة

الأنظمة لا تعتمد على النواة المعدة مسبقاً أو أنظمة الملفات الجذرية، تأكّد من حصولك على جميع التعليمات البرمجية المصدر والوثائق أو البرامج النصية لإعادة بناء نظامك بالكامل من البداية. استخدم أنظمة التحكم في المراجعة لتتبع التغييرات التي تجريها على المكونات المختلفة التي تستخدمها في نظامك. يُعد التكامل جزءاً مهماً من العمل مع Linux المدمج: يجب القيام به

بطريقة نظيفة.



# الدعم: حلول تجارية كاملة

بائعون مثل Montavista أو Wind River أو Timesys يوفرون حزمة [متكاملة](#) للوحدة وأدوات بناء النظام والتطبيق

أدوات التطوير لنظام التشغيل Linux المدمج، بالإضافة إلى الدعم.

المزايا: ممثل واحد معروف للتعامل معه، من المفترض أن تكون الحلول مختبرة جيداً والدعم الشامل العيوب: الاعتماد على أدوات خاصة بالبائع، وإصدارات النواة والمكونات الخاصة [بالبائع](#)، والثبيت، والتكلفة العالية، والدعم ليس بالضرورة جيداً



# الدعم: المجتمع

مطورو المكونات المختلفة ومجتمع المصادر المفتوحة كـIRC<sup>إلى</sup> من خلال القوائم البريدية، من الكل عموماً دعمًا جيداً وفي الوقت المناسب. لاحتاج إلى فهم كيفية عمل المجتمع، أو أن تكون جزءاً منه بشكل أفضل، للاستفادة من الدعم الجيد.

المزايا: تكلفة صغيرة، وردود فعل سريعة وفعالة بشكل عام، تتيح لك المهندسين لاكتساب المعرفة

العيوب: دعم الإصدارات الحديثة فقط من المكونات، غير واضح ممثل، بحاجة إلى بعض المعرفة بكيفية عمل المجتمع، ولا يعمل مع الكود المغلق



# الدعم: الدعم التجاري للحلول المجتمعية

- الشركات التي ليس لديها أي منتج محدد، وتقدم الدعم للمكونات الحالية مفتوحة المصدر
- شركات مثل DENX وBootlin ومئات الشركات الأخرى الصغيرة والمتوسطة الحجم شركات.
- المزايا: ممثل واحد معروف، استخدام مصدر مفتوح معروف المكونات بحيث تظل مستقلةً عن موفر الدعم، وهو الدعم الذي يهتم بمشكلتك المحددة حتى في حالة استخدام المكونات القديمة
- العيوب:

# ذكرى المظهر



هـ هو نظام التشغيل المدمج الشهير المستند إلى Linux والذي تم تطويره بواسطة Google، يستهدف الهاتف بشكل أساسـيـ ولكنـه قـابلـ لـلاـسـتـخـدـامـ فيـ تـطـبـيقـاتـ أـخـرـيـ أـيـضـاـ هـ لـيـسـ نـظـامـ Linuxـ تقـليـديـاـ: لقد أعادـتـ Googleـ فـقـطـ اـسـتـخـدـامـ نـوـاهـ Linuxـ وـ

عـدـدـ قـلـيلـ مـكـونـاتـ مـسـاحـةـ الـمـسـتـخـدـمـ،ـ وـلـكـنـ الـفـالـلـيـةـ الـعـظـمـيـ مـنـ النـظـامـ خـاصـ بـنـظـامـ Androidـ يـحـتـاجـ إـلـىـ مـهـارـاتـ وـمـعـرـفـةـ مـحـدـدـةـ لـلـعـملـ عـلـىـ Androidـ

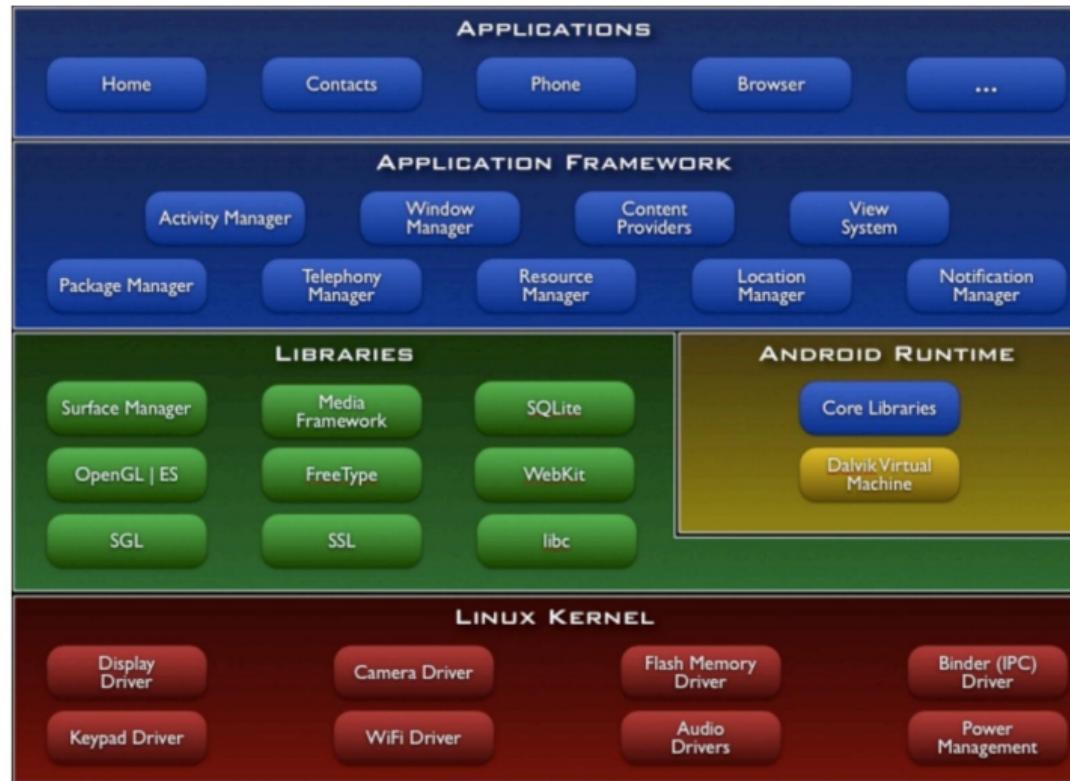
هـ تـحـتـاجـ الـنـوـاهـ إـلـىـ تـعـدـيـلـاتـ كـبـيرـةـ إـلـىـ حدـ ماـ لـتـكـونـ مـتـوـافـقـةـ مـعـ And~roidـ (ـإـدـارـةـ الـطـاـقةـ،ـ وـالـاتـصـالـ بـيـنـ الـعـمـلـيـاتـ،ـ وـماـ إـلـىـ ذـلـكـ)

هـ مـعـظـمـ الـمـكـونـاتـ الـتـيـ طـوـرـتـهاـ Go~gleـ مـرـخصـةـ بـمـوجـبـ تـرـخـيـصـ Apacheـ

الـنـوـاهـ هـيـ الـمـكـونـ الـوـحـيدـ تـحـتـ رـخـصـةـ GPLـ.



# بنية الروبوت





# خاتمة

يقدم Linux وعالم المصادر المفتوحة مجموعة واسعة من المكونات والأدوات الازمة  
تطوير النظام المدمج

تتمتع هذه المكونات بالعديد من المزايا: التركيز على القيمة المضافة، والتكلفة المنخفضة، والتحكم الكامل، وما إلى ذلك. يتوفّر الدعم، سواء من المجتمع أو الشركات التجارية



# خاتمة

يقدم Linux وعالم المصادر المفتوحة مجموعة واسعة من المكونات والأدوات الازمة  
تطوير النظام المدمج

تتمتع هذه المكونات بالعديد من المزايا: التركيز على القيمة المضافة، والتكلفة المنخفضة، والتحكم الكامل، وما إلى ذلك. يتوفر الدعم، سواء من المجتمع أو الشركات التجارية . فماذا عن Linux في منتجك المضمن التالي؟

# أسئلة؟

توماس بيتازوني

thomas.petazzoni@bootlin.com

تطوير نظام Linux المضمن، 23 في 27 يناير 2012، في أفينيون (الإنجليزية) [تطوير نظام Linux المحاضر، 6 في 10 فبراير 2012] تولوز [تطوير نواة Linux وبرامج التشغيل المضمنة، 19 في 23 مارس 2012، في أفينيون (الإنجليزية)] [تطوير Linux الجديد، الفرنسية] [تطوير نواة Linux وبرامج التشغيل المضمنة، 19 في 23 مارس 2012، في أفينيون (الإنجليزية)] [تطوير Linux الجديد، 4 في 8 يونيو 2012، في ليون (الفرنسية)] [تطوير Linux الجديد وبرامج التشغيل، 4 في 8 يونيو 2012، في تولوز (الفرنسية)] [جميع المعلومات على <https://bootlin.com/fr/formation/sessions/>]