



(گنو؟) لینوکس توکار (Embedded Linux)

هدایت وطن خواه

همایش روز آزادی نرم افزار اصفهان

فهرست مطالب

- لینوکس توکار؟!!
- مهم‌ترین و معمول‌ترین ملاحظات
- تفاوت‌های معمول سخت‌افزاری
- تولید یک لینوکس توکار
- نرم‌افزارهای پرکاربرد
- کتاب‌خانه‌های گرافیکی
- توسعه نرم‌افزار برای لینوکس توکار
- خلاصه
- لینک‌های مفید

لینوکس توکار؟!!

- استفاده از لینوکس (معمولا به صورت یک توزیع گنو/لینوکس کوچک) در رایانه‌های توکار
- رایانه‌ی توکار؟
 - گوشی‌های تلفن همراه
 - دستیارهای دیجیتال شخصی
 - پخش‌کننده‌های صوتی/تصویری
 - تجهیزات شبکه (مودم ADSL، نقاط دسترسی بی‌سیم، ...)
 - تجهیزات پزشکی
 - اتوماسیون صنعتی
 - ...

مهم‌ترین و معمول‌ترین ملاحظات سیستم‌های توکار

- حافظه کم
- قدرت پردازشی کم
- فضای ذخیره‌سازی کم
- بوت سریع
- قطع ناگهانی برق/خاموش شدن ناگهانی
- عدم تعامل کاربر با سیستم‌عامل

تفاوت‌های معمول سخت‌افزاری

- پردازنده و معماری ارتباطی
- فرآیند بوت
- دستگاه ذخیره‌سازی: انواع حافظه‌های فلش
- سیستم‌های پرونده‌ی اختصاصی: JFFS2 و ...
- تکنولوژی‌های مختلف حافظه اصلی

تولید یک لینوکس توکار

- نحوه نگاه به مسئله
- پایین به بالا
- بالا به پایین
- ابزارهای مورد استفاده:
 - بدون ابزار آماده
 - ابزارهای ساخت/ایجاد لینوکس توکار
 - استفاده و سفارشی سازی از یک توزیع گنو/لینوکس توکار
 - تبدیل یک توزیع گنو/لینوکس (مثل فدورا یا دبیان) به توزیع توکار

نکات مشترک

- تولید بر روی سیستم‌های معمول رومیزی یا سیستم‌های پر قدرت
- انتخاب نرم‌افزارهای مورد استفاده
- زنجیره ابزار (Toolchain) مورد استفاده
- ابزارهای بومی: کامپایل ابزارهای **cross-compile**
- ابزارهای **cross-compile**: کامپایل نرم‌افزارهای نهایی روی سیستم میزبان
(سیستم مورد استفاده برای تولید)
- «نصب» لینوکس توکار
- ابزار/روش انتقال سیستم عامل به دستگاه
 - برنامه‌های برنامه‌ریزی
 - انتقال از شبکه/پورت‌های ارتباطی

تولید یک توزیع توکار به صورت دستی

- نصب زنجیره ابزار مربوط به سیستم هدف بر روی سیستم میزبان: ابزارهای لازم برای cross-compiling
- برپاسازی محیط لازم برای کامپایل برنامه‌ها با ابزارهای نصب شده
- پیکربندی و کامپایل هسته لینوکس
- پیکربندی و کامپایل تمامی نرم‌افزارهای مورد نیاز
- نصب نرم‌افزارهای کامپایل شده در یک شاخه مجزا
- ایجاد یک فایل سیستم با قالب مورد نیاز از نرم‌افزارهای نصب شده
- ایجاد فایل سیستم/تصویر مورد نیاز برای بوت سیستم از بوت لودر و/یا هسته
- نصب تصویر بوت و فایل سیستم بر روی سخت‌افزار هدف

ابزارهای ساخت لینوکس توکار

- تسهیل و خودکارسازی بسیاری از قسمت‌های فرآیند تولید یک لینوکس توکار
- نمونه‌ها:

• Buildroot: یک سیستم پیکربندی و ساخت شبیه هسته‌ی لینوکس

• OpenEmbedded: سیستم تولید با انعطاف بالا و کمی پیچیده

• Scratchbox

• PTXdist

• LTIB

• Crosstools (فقط زنجیره ابزار)

• ...

استفاده از توزیع‌های توکار موجود

- سفارشی‌سازی یک توزیع توکار موجود براساس نیازمندی‌های کار
 - معمولا روند تولید ساده‌تری دارند
 - برای محصولات مشابه ایده‌آل هستند
 - در صورت تفاوت قابل توجه نیازمندی‌های کاربر چندان سودی ندارند
- نمونه‌ها:
 - Meego/Maemo
 - توزیع Ångström
 - Openmoko Linux
 - HP webOS
 - Fedora ARM

توکارسازی یک توزیع گنو/لینوکس!

- مناسب برای سخت افزارهای توکار مبنی بر x86
- یک روش بالا به پایین
- نسبتا سریع
- نیازمند دانش کافی از توزیع مورد نظر و حذف قسمت های زائد
- نامناسب در صورت محدودیت قابل توجه منابع

نرم افزارهای پر کاربرد

- بوت لودر: u-boot ، syslinux ، ...
- هسته سفارشی شده
- هسته زمان حقیقی
- کتابخانه‌ی زبان سی: glibc و uClibc
- Busybox
- سیستم گرافیکی پایه
- استفاده از FrameBuffer
- X Server
- ...

کتابخانه‌های گرافیکی

Gtk ●

Qt ●

پشتیبانی از X11 و Linux FrameBuffer ●

قدیم: Qtopia / Qt Extended ●

جدید: ●

Qt Extended Improved ●

Qt on Embedded Systems ●

DirectFB ●

OpenGL ●

SDL ●

... ●

توسعه نرم افزار برای لینوکس توکار

- تولید بر روی سیستم رومیزی با کمک ابزارهای قدرتمند و `cross-compiler`
- تست و اشکال زدایی نرم افزار با کمک شبیه سازهای نرم افزاری مثل `qemu`
- تست و اشکال زدایی بر روی سیستم هدف به صورت از راه دور، مثال: اجرای `gdb` سرور بر روی سیستم هدف و اشکال زدایی با کمک ابزارهای قدرتمند رومیزی
- نمونه: استفاده از `Eclipse` یا محیط های مبنی بر آن برای تولید و اشکال زدایی نرم افزار تا مرحله ی نهایی
- تست نهایی بر روی سیستم هدف

- لینوکس توکار: استفاده از لینوکس، معمولا به همراه تعدادی از ابزارهای آزاد دیگه، تو کامپیوترهای توکار
- بسته به محدودیت‌ها و سخت‌افزار مورد استفاده تهیه می‌شود
- در صورت وجود، ممکن است از توزیع‌های آماده بتوان استفاده کرد
- معمولا لازم است که کلیه نرم‌افزارها را برای سخت‌افزار هدف کامپایل کنیم
- کامپایل برنامه‌ها را روی سیستم هدف انجام نمی‌دهیم، و روی سیستم‌های میزبان با **cross-compiling** انجام می‌دهیم
- هسته باید کاملا سازگار با سخت‌افزار هدف کامپایل شود
- تست و اشکال‌زدایی برنامه‌های سیستم هدف نیز در حد امکان در سیستم‌های عادی انجام می‌گیرد

- Elinux: http://elinux.org/Main_Page
- Free Electrons Documents & Tutorials:
<http://free-electrons.com/>

متشکرم،

پایان