Linux en la tostadora

Una breve introducción al desarrollo de sistemas embebidos con GNU/Linux

Linux en la tostadora

Lisandro Damián Nicanor Pérez Meyer
perezmeyer usando gmail.com,
cepanet.com.ar y uns.edu.ar

¿Qué vamos a tostar hoy?

- Para quién es la charla
- Linux y los sistemas embebidos
- Definiciones básicas

Conceptos de:

- Procesadores
- Boot loaders
- Kernel
- Root filesystem
- Dispositivos de almacenamiento
- Toolchains

¿Porqué usar GNU/Linux?

- ¡Porque es software libre! ¿No me creen?
- Maduro y de alta performance → gracias a que es SL
- Soporta un amplio rango de aplicaciones y protocolos (¿el SO con mayor soporte existente?) → porque es SL
- Escalable: desde pequeños sistemas a supercomputadoras → porque cada cuál puede adaptarlo a gusto, porque es SL
- No se requiere el pago de regalías... ¡porque es SL!
- Una gran comunidad por detrás... la comunidad del SL
- Cada vez mas soporte de los fabricantes... porque así lo pide la comunidad

¿Y ya hay productos comerciales con GNU/Linux?

Por supuesto :-)

http://linuxfordevices.com

- Nokia compró a Trolltech para utilizar Qt (la base fundacional del escritorio KDE) en sus dispositivos embebidos.
- Google creó los SSOO Chrome y Android basandose en Linux y pensando en sistemas embebidos

¿Qué es un sistema embebido?

Un sistema:

- Que contiene algún tipo de procesador.
- Generalmente diseñado para un propósito específico.
- No suele tener muchos recursos...
- Generalmente usa software preinstalado.

¿Y qué es Linux?

- Núcleo (kernel) del sistema operativo.
- Ocupado de hacer que las cosas anden.
- ~120 MB de código fuente... comprimido :-)

Tipos de hosts (máquinas de desarrollo)

- Linux
- Unix
- Windows

Cross toolchain

Herramientas de compilación que corren bajo el host pero compilan para la arquitectura del sistema embebido.

RootFS

Sistema de archivos principal.

Tipos de procesadores

- Stand alone: requieren de chips externos para manejar cosas como los buses, la DRAM, etc. Ejemplo: procesadores de PC.
- Sistems en Chip (SoC): cuentan con gran parte de los susbistemas requeridos (manejo de memoria y buses, ethernet, USB, etc.) en el mismo integrado. Ejemplos: Broadcom MIPS (Linksys), Intel ARM (Xscale).

El bootloader: iniciando el sistema

- Es el primer componente de software en ejecutarse
- Es el encargado de inicializar los periféricos como la DRAM, las cachés del procesador,...
- Y recién ahora se puede copiar a memoria y crear variables...
- Y luego cargar el kernel a memoria... e inicializarlo.

Algunos bootloaders: RedBoot, U-Boot: embebidos; LILO, GRUB, loadlin, Coreboot (ex LinuxBIOS): sistemas “mas grandes”.

Pero no es estricto

Funcionalidades de los bootloaders

Pueden cargar un kernel o un rootfs:

- De una memoria en la placa.
- De una tarjeta SD/MMC/...
- De un pendrive.
- De un servidor ftp.
- De un servidor NFS.

Ésta flexibilidad nos permite modificar muy rápidamente un sistema de archivos, por ejemplo, sin necesidad de perder tiempo flasheando... y sin reducir la vida útil de la memoria.

El kernel

- Responsable de manejar el hardware... bit a bit.
- Provee la capa fundamental de abstracción para que la gran mayoría del resto de los programas sean independientes de la plataforma.
- En el caso de los sistemas embebidos, puede ser necesario parchear el kernel para obtener cierta funcionalidad (por ejemplo, dar soporte a un determinado LCD).

Suele requerir leer bastante :-)
Tip: `ARCH=arm CROSS_COMPILE=arm-linux`
(Ése me llevó muuucho tiempo... por no tener de donde leer).

El sistema de archivos root (Rootfs)

Una de las últimas tareas del kernel durante el inicio del sistema es montar el sistema de archivos. Si bien el kernel no dicta una estructura, los diversos programan cuentan con encontrar las cosas en un lugar predeterminado. Por éste motivo, es necesario mantenerse dentro de los estándares.

Algunos directorios importantes

- /bin: binarios de usuario esenciales (bash, pwd, tar,...).
- /boot: Archivos estáticos utilizados por el bootloader.
- /dev: dispositivos y archivos especiales (el “acceso al hardware”).
- /etc: archivos de configuración del sistema.
- /lib: bibliotecas esenciales, módulos del kernel.
- /media y /mnt: puntos de montaje de dispositivos removibles y fijos respectivamente.
- /proc: sistema de archivos virtual con información del kernel y procesos .

Algunos directorios importantes II

- /root: home de root.
- /sbin: binarios esenciales de administración del sistema (fdisk, mount, shutdown, etc.).
- /sys: sistema de archivos virtual con información del systema y control de buses, dispositivos y drivers.
- /tmp: archivos temporales.
- /usr: jerarquía secundaria, contiene gran parte de las utilidades del usuario.
- /var: archivos variables guardados por demonios (servicios) y utilidades.

Linux en la tostadora – Dispositivos de almacenamiento

Los dispositivos de almacenamiento en sistemas embebidos suelen diferir mucho de sus contrapartes de escritorio.

Una de las tecnologías mas importantes en éste ámbito es la tecnología de memorias de estado sólido.

Las mismas suelen ser manejadas por el subsistema Memory Technology Devices MTD.

Son dispositivos que no se pueden tratar directamente como de bloques o caracteres, por lo que requieren de dispositivos bajo /dev especiales... y se particionan ¡modificando código del kernel!

Linux en la tostadora – Dispositivos de almacenamiento

Normalmente un dispositivo de almacenamiento se suele particionar con espacios reservados para:

- El bootloader.
- El kernel.
- El rootfs.
- Alguna mas por conveniencia inherente al dispositivo.

Pero ¿cómo pasamos éstos datos a la memoria?

- El bootloader se suele programar por primera vez a través de JTAG.
- El kernel y el rootfs se pueden transmitir por JTAG (¡muy lento!), por conexión serial (mas rápido... pero puede tardar 45' para 25 MB...) o por conexión ethernet... si el bootloader lo soporta.

Linux en la tostadora – Toolchains

Existen dos maneras distintas de obtener un toolchain: haciendo uno nosotros mismos o utilizando alguno pre hecho.

Hacerlo nosotros mismos nos dá mucho control... pero mucho trabajo.

Utilizar uno pre hecho implica confiar en otros... pero existen empresas que se dedican a ésto (por ejemplo, Montavista), y, por supuesto, proyectos comunitarios como Open Embedded y Buildroot.

¡hay que tener cuidado del vendor lock-in!

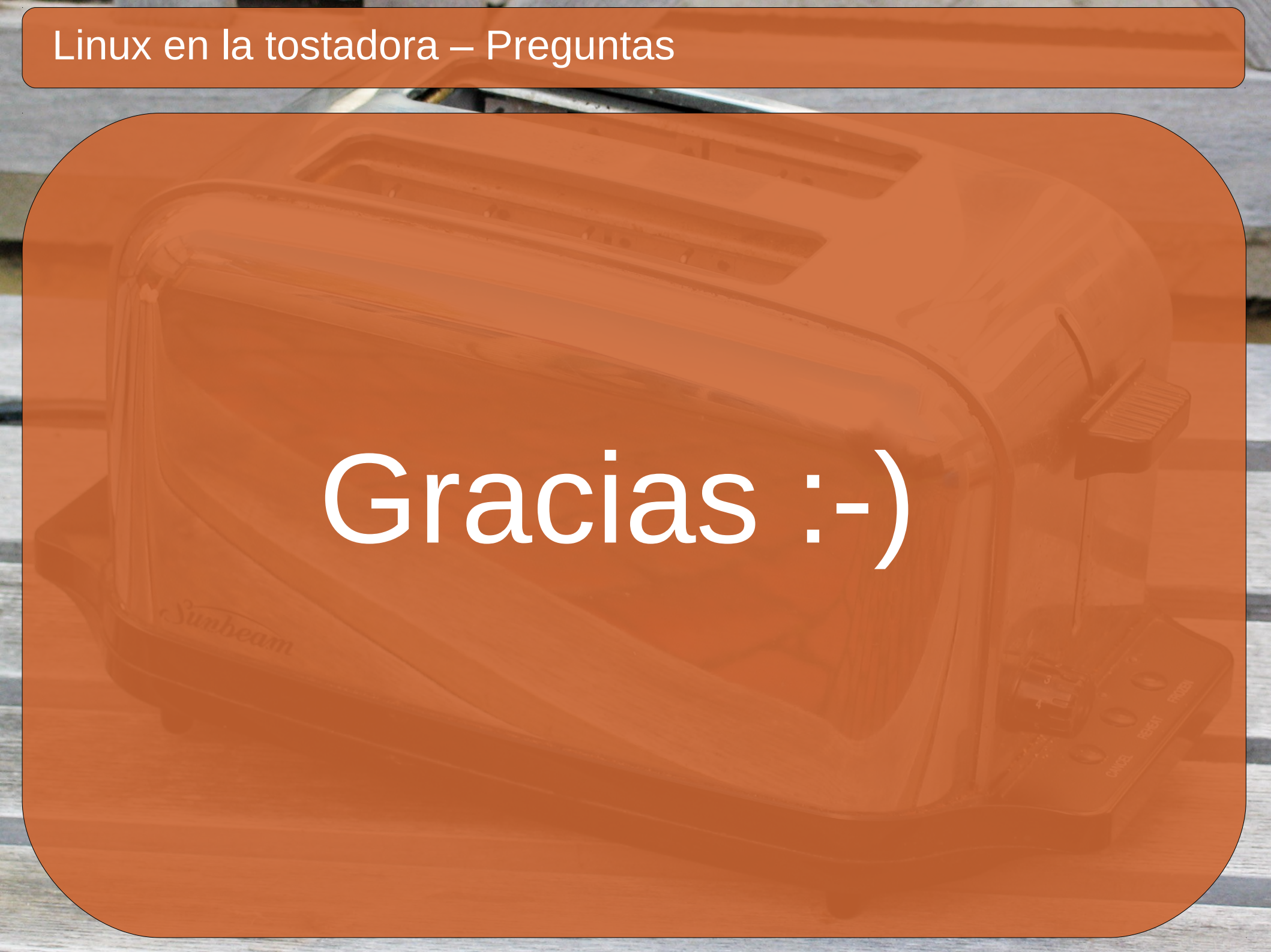
Linux en la tostadora – Referencias

Si bien la web es una fuente enorme de datos, uno o dos libros nunca vienen mal :-)

• Embedded Linux Primer: a practical, real-world approach. Christopher Hallinan, Prentice Hall, ISBN-10: 0-13-167984-8.
• Building embedded Linux Systems, Karim Yaghmour, Jon Masters et al. O'Reilly, ISBN: 978-0-596-52968-0.

Ambos libros cuentan con referencias a fuentes externas para temas particulares.

¿Preguntas?



Gracias :-)

Linux en la tostadora – Copyright y demáses

La imagen de fondo de la tostadora está disponible en http://commons.wikimedia.org/wiki/File:Hot_Dog_Toaster.jpg
Y se encuentra bajo dominio público y licencia CC-BY-SA 3.0

El mismo puede encontrarse en http://perezmeyer.com.ar/files/tostadora/