

CREACIÓN DE UN SISTEMA MÍNIMO CON GNU/LINUX

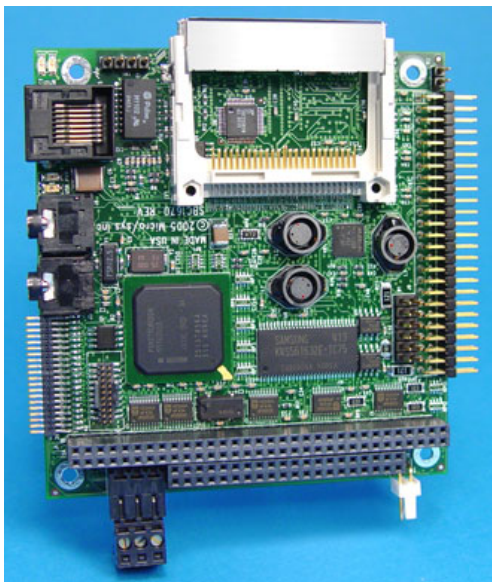
Dr. Luis Gerardo de la Fraga

Departamento de Computación
Cinvestav
E-mail: fraga@cs.cinvestav.mx

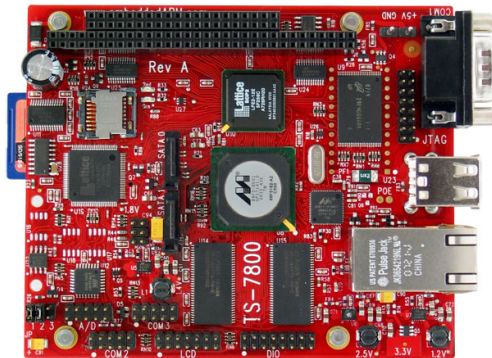
25-26 Octubre, 2011

SISTEMA MÍNIMO

- ▶ En vez de quitar todos los programas que no se usan, se puede arrancar una máquina con el software mínimo.
- ▶ Esto es recomendable hacerlo para crear un cortafuegos.
- ▶ Esto se debe de realizar para una Computadora en una Sola Tarjeta (Single Board Computer, SBC)



<http://www.embeddedsys.com/subpages/products/sbc1670.shtml>



[http://www.embeddedarm.com/products/
board-detail.php?product=TS-7800](http://www.embeddedarm.com/products/board-detail.php?product=TS-7800)

PASOS A SEGUIR

- ▶ Obtener un núcleo ya compilado
- ▶ Crear un sistema de archivos mínimo para que arranque ese núcleo
- ▶ Compilar algunos comandos básicos (se usará BusyBox)
- ▶ Configurar el sistema y arrancarlo
- ▶ Instalarle más software (DropBear, por ejemplo)

Podríamos usar el núcleo de esta computadora:

```
$ uname -a
```

```
Linux ceres 2.6.35-22-powerpc #33-Ubuntu Mon Sep 20 \
  00:42:22 UTC 2010 ppc GNU/Linux
```

El núcleo es el archivo

```
$ cd /boot
```

```
$ file vmlinux-2.6.35-22-powerpc
```

```
vmlinux-2.6.35-22-powerpc: ELF 32-bit MSB executable,  
PowerPC or cisco 4500, version 1 (SYSV),
```

```
statically linked,
```

```
with unknown capability 0x41000000 = 0x13676e75,
```

```
with unknown capability 0x10000 = 0xb0402,
```

```
not stripped
```

Podríamos crear un árbol de archivos para arrancar ese núcleo con un sistema operativo mínimo:

- ▶ /dev Archivos de dispositivos para E/S
- ▶ /proc Directorio para montar la información del núcleo
- ▶ /etc Archivos de configuración del sistema
- ▶ /sbin Binarios para el mantenimiento del sistema
- ▶ /bin Binarios del sistema
- ▶ /lib Bibliotecas compartidas en tiempo de ejecución
- ▶ /mnt Un punto de montaje para otros sistemas de archivos
- ▶ /usr Utilerias para usuarios
- ▶ /var Archivos de auditoría del sistema

El sistema de archivos debe llenarse con la siguiente información:

- ▶ El ejecutable de BusyBox (que contiene todas las utilerías necesarias)
- ▶ Las bibliotecas compartidas en /lib
- ▶ Los archivos de E/S en /dev
- ▶ Los archivos de configuración en /etc

BIBLIOTECAS

Las biblioteca estándar está compilada con información para trazar muchas cosas, intentar

```
$ LD_DEBUG=help /bin/ls
```

Para saber las bibliotecas que está usando un comando también se puede usar directamente el comando `ldd`

```
$ ldd /bin/ls
linux-vdso32.so.1 => (0x00100000)
libselinux.so.1 => /lib/libselinux.so.1 (0x0ffbf000)
librt.so.1 => /lib/librt.so.1 (0x0ff96000)
libacl.so.1 => /lib/libacl.so.1 (0x0ff6d000)
libc.so.6 => /lib/libc.so.6 (0x0fdd9000)
libdl.so.2 => /lib/libdl.so.2 (0x0fdb5000)
/lib/ld.so.1 (0x2024e000)
libpthread.so.0 => /lib/libpthread.so.0 (0x0fd7a000)
libattr.so.1 => /lib/libattr.so.1 (0x0fd55000)
```

BUSYBOX

- ▶ Busybox está disponible en <http://www.busybox.net>
- ▶ se considera a sí mismo la “navaja suiza” para sistemas empotrados (o embebidos).
- ▶ Busybox contiene una gran cantidad de utilerías para realizar un minisistema operativo.

- ▶ La última versión es `busybox-1.19.2.tar.bz2`
- ▶ Las instrucciones de su instalación se encuentran en `busybox-1.19.2/INSTALL`
- ▶ Para compilar todo el paquete que viene por defecto se hace

```
make defconfig
make
```
- ▶ Para instalarlo, por ejemplo, en `/home/pepito/SistemaMinimo`

```
sudo make CONFIG_PREFIX=/home/pepito/SistemaMinimo/ \
install
```

VERIFICACIÓN DEL SISTEMA

- ▶ Se puede usar el comando `chroot` para verificar si nuestro sistema de archivos está funcionando.
- ▶ Tal vez sea necesario copiar el binario del shell `bash` (al directorio `/bin`) y agregar las bibliotecas necesarias para que pueda ejecutarse (se checan con `ldd /bin/bash`).

CREACIÓN DE LA IMAGEN

Para crear la imagen que cargará el núcleo se realiza, en el directorio donde se tiene el árbol de archivos creado:

```
find . -print | cpio -ov -H newc > /home/pepito/imagen
```

Finalmente, se puede comprimir la imagen creada:

```
cd  
gzip imagen
```

ARRANQUE

- ▶ Para probar nuestra imagen podemos arrancar desde disco duro. El arranque lo maneja grub
- ▶ Como mi laptop es una iBook G4 (con Ubuntu 10.10) maneja yaboot en vez de grub.
- ▶ Para hacer arrancable una memoria USB hay que instalar también grub en ella.

La línea para arrancar el sistema mínimo en yaboot o grub es

```
image=/boot/vmlinux
    label="Mi sistema minimo"
    initrd=/boot/imagen.gz
    read-write
    append="root=/dev/ram0"
```

CONFIGURACIÓN DE DROPBEAR

Las llaves van en el directorio `/etc/dropbear` y el script para arrancar el servidor podría ser

```
$ cat /etc/arrancaDropBear
#!/bin/sh

touch /var/log/lastlog
echo -n "Setting dropbear server..."
/usr/sbin/dropbear -w
```

CONFIGURACIÓN DE dnsmasq (1/2)

Este podría ser el script para arrancar dnsmasq

```
#!/bin/sh
```

```
echo "Setting dnsmasq server..."
```

```
mkdir -p /var/lib/dhcp
```

```
/sbin/dnsmasq
```

CONFIGURACIÓN DE DNSMASQ (1/2)

Este podría ser el archivo de configuración de nuestro servidor dhcp:

```
$ cat /etc/dnsmasq.conf
dhcp-range=192.168.10.100,192.168.10.200,24h
dhcp-host=00:B0:D0:CF:66:C1,cliente01,192.168.10.91
#
dhcp-leasefile=/var/lib/dhcp/dnsmasq.leases
```

¡ SUERTE !