



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Placa EPIA con procesador VIA C3 y... linux (21976 lectures)

Per **Celso González**, [PerroVerd](http://mitago.net) (<http://mitago.net>)

Creado el 13/12/2002 10:29 modificado el 13/12/2002 10:29

Esta placa se ha puesto muy de moda últimamente debido a su bajo precio, su tamaño reducido y ya que tengo una me he decidido contar mis experiencias con ella y los trucos que he usado para hacerla funcionar en mi linux.

Introducción al cacharro

El chisme en cuestión es una placa de 17x17 cm (formato mini-itx).

Incorporara un procesador Via C3 a 800 Mhz con un ventilador ball-bearing (muy silencioso), controladora de disco ata 100, tarjeta de red 100, tarjeta gráfica AGP 2x y de sonido, puerto serie, paralelo y usb también trae una ranura de expansión PCI. En resumen, que trae de todo en una mierdecilla de espacio y a un precio muy competitivo (en el momento de escribir esto su precio está en torno a los 130 E)

El consumo es muy pequeño (fuente ATX normal y corriente) y lo único que necesitamos añadir a la placa para que funcione es un chip de memoria (DIMM de 133) y después si queremos un disco, un cdrom, un dvd o similar.

Respecto al rendimiento, funciona como un Celeron 600 (más o menos) que resulta más que suficiente para ver un divx, por ejemplo.

En mi caso, lo estoy utilizando como router server y está funcionando muy deshagado y sin ningún problema.

Configuración con linux

Si alguien quiere compilar el kernel estas son algunas de las opciones que se deberían marcar. (Esta configuración es para un kernel 2.4.20-pre1 aunque es muy similar para casi todos los 2.4.x)

Processor type and features

- >(CyrixIII/VIA-C3,VIA-C5) Processor family
- >/dev/cpu/*/msr - Model-specific register support
- >MTRR (Memory Type Range Register) support

General setup

- >ACPI support (solo para los valientes)
- ATA/IDE/MFM/RLL support
- >VIA82xxx chipset support
- Network device support
- >Ethernet (10 or 100Mbit)
- >Via Rhine support
- >Use MMIO instead of PIO (EXPERIMENTAL)

Character devices

- >/dev/agpgart (AGP support)
- >VIA chipset support

Console drivers

- >Frame-buffer support
- >Trident support (EXPERIMENTAL)

Sound

- >VIA 86C686 Audio codec

USB support

- >USB Host Controller Drivers
- >UHCI Alternate driver (JE) support



En los kernel 2.5 podemos añadir una opción muy interesante que nos va a permitir un ahorro de energía, ya que baja la frecuencia del procesador cuando este se encuentra ocioso.

Processor type and features

->CPU Frequency scaling

-->Via Cyrix III Longhaul

Si alguien quiere más detalles de la configuración o de algún aspecto que me envíe un comentario. De todas formas esto no es lo único que hay que saber de este bicho, aún hay más, pasa la página :)

Optimizaciones del procesador

Hay una serie de discusiones sobre la familia a la que tiene que pertenecer este procesador. Intentaré aclarar un poco las cosas

La salida de `/proc/cpuinfo` me da la siguiente información:

```
processor : 0
vendor_id : CentaurHauls
cpu family : 6
model : 7
model name : VIA Ezra
stepping : 8
cpu MHz : 800.047
cache size : 64 KB
fdiv_bug : no
hlt_bug : no
f00f_bug : no
coma_bug : no
fpu : yes
fpu_exception : yes
cpuid level : 1
wp : yes
flags : fpu de tsc msr cx8 mtrr pge mmx 3dnow
bogomips : 1595.80
```

Lo más importante que vemos es que la `cpu family` es 6 (i686), vemos que además disponemos del juego de instrucciones de `mmx` y el de `3dnow`, la cosa promete, pero ahora es cuando aparecen los problemas

El problema del `cmov`

Hay una instrucción opcional en el juego de instrucciones de un i686 que es la `cmov`.

Esta instrucción funciona en determinados casos (de forma muuuuy lenta) y en otros sencillamente nos da un error de `Illegal instruction`.

El compilador `gcc3.2` hace uso de estas instrucciones y vemos que si compilamos para arquitectura i686 podemos encontrarnos con este problema: o bien rendimiento muy lento o bien que nuestro programa peta alegremente.

Benchmarks

Aparte de todo esto una serie de benchmarks parece que indican que el código generado para procesadores i486 es más rápido que para i586 o i686 en esta máquina.

Esto ha provocado que las optimizaciones de compilador en el kernel 2.4.20 hallan "degradado" este procesador a un misero i486 (con `mmx` y `3dnow`, eso sí)

Hay una discusión abierta en la lista del kernel y la de `gcc` referente a este tema, y todavía parece que no se han puesto de acuerdo ;)

Problemas con `debian`.

Hay una librería en `debian sid` (`openssl`) que por cuestiones de optimización lleva mucho código ensamblador incluido. Este código está optimizado según la arquitectura, y la máquina en tiempo de ejecución decide sobre que arquitectura



enlazar.

El problema es que el código para i686 hace un uso intensivo de la instrucción... cmov, como no :) y la detección automática dice que estamos en un i686, con lo que tenemos montones de Illegal Instruction en paquetes tan poco importantes como el ssh, por ejemplo.

La gente de openssl dice que el problema no es suyo, que es de la gente del kernel, o bien de la gente del gcc. La cuestión es que aún no hay un parche oficial.

Socorro, ¿qué hago, como voy a renunciar a mi debian :)

No problema, tengo unos paquetes altamente no oficiales para libssl, openssl y libssl-dev que se pueden descargar añadiendo esta línea al sources.list antes de las líneas de los paquetes oficiales

Openssl compilado para viac3:

```
deb http://bulma.net/~perroverde/debian /
```

Y esto es todo

E-mail del autor: celso _ARROBA_ mitago.net

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1623>