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

Aprovechar la potencia de nuestro procesador x86 (14618 lectures)

Per **Celso González**, ***PerroVerd*** (http://mitago.net)

Creado el 26/11/2001 00:40 modificado el 26/11/2001 20:46

> Al compilar un programa podemos indicar al compilador (normalmente gcc) una serie de opciones para que genere código optimizado para nuestro procesador. De esta forma podemos indicar al compilador que genere código optimizado para Pentium III (por ejemplo) que será mucho más eficiente que si no indicamos nada.
> Un ejemplo de estas optimizaciones es la distribución Mandrake[1] que tiene todos los paquetes optimizados para 586 o superior, obteniendo una mejora de velocidad aproximada de un 30%

Esto también tiene el problema que si compilamos para una arquitectura en concreto no podemos ejecutar ese programa en una máquina inferior, por ejemplo, no podemos instalar una distribución Mandrake en un 386

Aclarado esto, paso a explicar que la mayoría de distribuciones traen paquetes (rpm o deb) compilados para 386. El objetivo de este artículo es lograr construir paquetes optimizados para nuestra arquitectura

RPM

Aquí no hace falta que explique nada, el amigo **Gorka Olaizola** escribió hace mucho tiempo un mini-howto para generar paquetes rpm
El howto lo podemos encontrar en la página personal de Gorka[2] (la versión 1.1 es más completa), aunque me suena haberlo visto también por el Insflug[3]

DEB

DISCLAIMER: El procedimiento descrito aquí es meramente intuitivo ya que básicamente he intentado trasladar la idea de compilar un rpm y desconozco si es la forma correcta de realizarlo, como mínimo puedo asegurar que compila y se ejecuta ;)

En primer lugar leer el howto escrito para rpm ya que es la fuente de inspiración y hay algunos conceptos que se explican en él que yo no voy a tratar

Debemos bajar las fuentes del paquete que queremos compilar, para esto debemos tener en el fichero `/etc/apt/sources.list` una línea con la siguiente pinta:
`deb-src ftp:/ftp.xx.debian.org/debian/ stable main non-free contrib`
Donde xx es el mirror de debian que empleemos (es, fi, uk...) y stable se puede cambiar por testing o unstable. Si el paquete es non-US el procedimiento es el mismo pero referenciando a un servidor non-US

Después del **apt-get update** de rigor haremos un **apt-get source nombre_del_paquete** en mi caso voy a emplear el paquete hello, que es un paquete para hacer pruebas

Una vez que hemos bajado el paquete tendremos un directorio hello-x.xx y tres ficheros hello-x.xx (.diff.gz, .dsc y .orig.tar.gz), a nosotros nos interesa el fichero **hello-x.xx/debian/rules**, por lo que usando nuestro editor favorito lo abrimos

Ahora viene la parte interesante, se trata de añadir a la variable CFLAGS o CXXFLAGS los paramétros de compilación de nuestra máquina. Para hacer esto debemos localizar la sección **build:** y añadir al make o al configure estas variables

En el paquete hello el fichero rules original tiene esta pinta

```
package=hello

build:
  $(checkdir)
  ./configure --prefix=/usr
  $(MAKE) CFLAGS=-O2 LDFLAGS=
  touch build
...
```

Por ejemplo voy a compilar para un Pentium de forma que el fichero rules me quedará así:

```
package=hello

build:
  $(checkdir)
  ./configure --prefix=/usr
  $(MAKE) CFLAGS="-O2 -fomit-frame-pointer \
                  -mpreferred-stack-boundary=2 -march=i586" -LDFLAGS=
       touch build
...
```

Por último solo nos falta generar el paquete .deb correspondiente, para esto haremos nos iremos al directorio hello-x.xx y escribiremos **dpkg-buildpackage** después hacemos un **cd ..** y encontraremos el paquete hello-x.xx

Notas Finales

- Aunque en los paquetes rpm podemos mezclar i386 con i586 en debian no tenemos forma de saber si el paquete está optimizado para determinado procesador.
- Si vamos a optimizar algo debemos tener un poco de vista :) será mucho más efectivo optimizar el paquete glibc (usado por casi todo) que el paquete man, por poner un ejemplo.
- No esperemos maravillas, aunque podemos ganar algo de velocidad no significa que esta pueda llegar a ser apreciable.
- Normalmente para compilar necesitaremos paquetes extras de desarrollo, por ejemplo si queremos compilar el paquete apache casi seguro que necesitaremos el paquete apache-dev. Para saber que paquetes necesitamos podemos hacer **dpkg-buildpackage -D** de forma que comprueba las dependencias necesarias
- No todos los ficheros son tan fáciles de manejar como el del paquete hello :(normalmente cuanto más complejo es el programa mucho más complejo es el makefile el configure o el rules correspondiente

Parámetros de optimización

Los parámetros de optimización que se emplean son los mismos que usa el kernel 2.4.x.
La lista es la siguiente

- **"486"** for the AMD/Cyrix/IBM/Intel 486DX/DX2/DX4 or SL/SLC/SLC2/SLC3/SX/SX2 and UMC U5D or U5S.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i486

- **"586"** for generic Pentium CPUs, possibly lacking the TSC (time stamp counter) register.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

- **"Pentium-Classic"** for the Intel Pentium.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

- **"Pentium-MMX"** for the Intel Pentium MMX.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

- **"Pentium-Pro"** for the Intel Pentium Pro/Celeron/Pentium II.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i686

- **"Pentium-III"** for the Intel Pentium III and Celerons based on the coppermine core.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i686

- **"Pentium-4"** for the Intel Pentium 4.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i686

- **"K6"** for the AMD K6, K6-II and K6-III (aka K6-3D).
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=k6

- **"Athlon"** for the AMD K7 family (Athlon/Duron/Thunderbird).
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i686
  -malign-functions=4

- **"Crusoe"** for the Transmeta Crusoe series.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i686
  -malign-functions=0 -malign-jumps=0 -malign-loops=0

- **"Winchip-C6"** for original IDT Winchip.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

- **"Winchip-2"** for IDT Winchip 2.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

- **"Winchip-2A"** for IDT Winchips with 3dNow! capabilities.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

- **"CyrixIII"** for VIA Cyrix III or VIA C3.
  -O2 -fomit-frame-pointer -fno-strict-aliasing -fno-common -pipe -mpreferred-stack-boundary=2 -march=i586

Por último, si alguien conoce mejores parámetros de optimización agradecería me los comunicase

---

Lista de enlaces de este artículo:

1. http://www.mandrake-linux.com
2. http://web.jet.es/olsago/docs.html
3. http://www.insflug.org

E-mail del autor: celso _ARROBA_ mitago.net
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=1025