

Midiendo Debian Sid, Gentoo 1.4, kernel 2.4, 2.6

David Zubillaga - Diciembre 2003

Índice de contenido

1 Motivación.....	3
2 Competidores.....	3
3 Hardware.....	3
4 Pruebas: Visión global.....	3
5 Metodología.....	4
6 Detalles sobre las instalaciones.....	5
7 Pruebas.....	5
7.1 Condiciones iniciales.....	5
7.2 Resultados.....	7
7.2.1 Xfree86.....	7
7.2.2 OpenGL.....	8
7.2.3 Visión de JPEG.....	9
7.2.4 Redimensión JPEG.....	10
7.2.5 Vídeo.....	10
7.2.6 Codificación MP3.....	11
7.2.7 Descompresión de archivos.....	11
7.2.8 Rendimiento Disco Duro, aplicación.....	12
7.2.9 Rendimiento Disco Duro, bajo nivel.....	13
7.2.10 Compilación de un kernel Linux.....	14
8 Conclusiones.....	14
8.1 Debian y Gentoo.....	14
8.2 Linux 2.4.23 y Linux 2.6.0.....	15
9 Sobre las pruebas.....	15

1 Motivación

Soy usuario de distribuciones Debian GNU/Linux desde Hamm (Debian 2.0, ¿1999?). A día de hoy sigo usando Debian (Sid) a diario aunque como cualquier *geek* me encanta probar todo tipo de sistemas operativos, en especial aquellos con licencias libres.

Poco tiempo después de enterarme de la existencia de Gentoo Linux empecé a tomarla en consideración no ya para probarla, sino para sustituir mi vieja y digna de confianza Debian.

¿Razones? Sistema de *ports* a-la-BSD pensado para compilar todo el sistema y sus aplicaciones haciendo uso, si uno lo quiere, de todas las optimizaciones existentes en el “nuevo” gcc3. Tener la sensación de que todo es lo mejor que puede ser. Tener un sistema construido por ti mismo casi desde cero, sin llegar a los extremos de Linux From Scratch.

Pero me pregunto: ¿merece la pena? Claro que merece la pena **probar** un sistema así, pero, ¿tanto como para darme todo el trabajo de cambiar mi actual sistema por uno nuevo? Pues nada, vamos a ponerlos en una pista, y a ver quién corre más, y cuanto más.

Reconozco que me gustan los números, los *benchmarks*. Claro, no conozco a nadie que no tenga mil cosas que decir en contra de los benchmarks: No valen para nada, no hay ninguno bien hecho, hay demasiadas variables como para que los resultados tengan valor...

Ya sé que muchos pensarán lo mismo de estas pruebas. No intento convencer a nadie de nada, porque hay muchos más criterios para elegir un sistema u otro además del rendimiento. He hecho esto “just-for-fun” y para ayudarme a MI a decidir, haciendo pruebas de aplicaciones que YO uso.

Continuemos...

2 Competidores

En principio mi intención era medir el tiempo que tardaban en realizar diversas tareas cotidianas algunos programas ejecutándose sobre Debian Sid y sobre Gentoo 1.4, ambas lo más actuales posible.

Felizmente, mientras escribía los scripts que conforman el test, apareció Linux 2.6.0, con lo que ya de paso también compararía mi instalación de Debian Sid 2.4 con la 2.6.

En resumen, finalmente he medido el rendimiento de las siguientes combinaciones Distribución/Kernel:

- Debian GNU/Linux unstable / 2.4.23
- Gentoo Linux 1.4 / 2.4.23
- Debian GNU/Linux unstable / 2.6.0

3 Hardware

CPU: Athlon XP 1700+ (1463 Mhz, 128Kb Cache L1, 256Kb Cache L2)

RAM: 256Mb DDR 266 MHz

Placa: [DFI](#) AD70-SC North Bridge VIA KT266A South Bridge VT8233

Disco Duro: IBM IC35L040AVVA07-0 (IDE 40Gb 7200rpm ATA100 2Mb Cache)

Tarjeta gráfica: Nvidia GeForce 2 MX 32Mb (AGP 4x)

4 Pruebas: Visión global

La mayor parte de pruebas que he hecho han sido *pruebas de aplicación* (Ver [Linux Benchmarking HOWTO](#)), aunque también incluyo alguna prueba sintética para comprobar el rendimiento de, por ejemplo, Xfree86.

Bueno, estas son las pruebas con las que he “castigado” a mi PC:

- **Redimensión JPEG:** Redimensión de 50 JPEG de 1600x1200 a 720x540 pixels mediante una de las utilidades incluidas en [Imagemagick](#).
- **Codificación MP3:** Codificación de un fichero de sonido PCM en formato WAV extraído de un CD-Audio a MP3 usando [lame](#).

- **Descompresión de archivos:** Descomprimir un fichero tar.bz2.
- **Vídeo:** Reproducción de un vídeo en formato AVI codificado mediante el codec DivX5 usando [mplayer](#).
- **OpenGL:** Ejecución de glxgears.
- **Xfree86:** Test sintético sobre diversas operaciones típicas en Xfree86 usando x11perf.
- **Visión de JPEG:** Visión de 50 JPEG de 1600x1200 pixels en un display X de 1024x768 usando el algoritmo de escalado de mejor calidad con el programa [GQView](#).
- **Rendimiento Disco Duro, bajo nivel:** Ejecución de hdparm para medir la transferencia en Mb/s del disco duro con y sin caché.
- **Rendimiento Disco Duro, aplicación:** bonnie++ realiza un test del rendimiento del disco duro más sofisticado que hdparm, simulando el comportamiento de diferentes tipos de aplicaciones.
- **Compilación de un kernel Linux:** No podía faltar.

5 Metodología

Todas las pruebas se ejecutan automáticamente lanzando un shellscrip (zubmark.sh) que va llamando una por una las pruebas que encuentre en el directorio “tests”. En este caso, las pruebas que escribí, fueron:

```
~/zubmark$ ls tests
bonnie++  gqview  imagemagick  lame      tarxjf
glxgears  hdparm  kernel       mplayer   x11
```

Cada uno de esos directorios listados debe contener dos scripts, uno que se encargue de generar el número de versión de la aplicación u otras informaciones relevantes sobre el test y otro que haga ejecutar el o los comandos que conforman el test, generando el o los ficheros necesarios con los resultados.

Posteriormente el script inicial (zubmark.sh) recopilará la información generada por cada test y la almacenará en varios ficheros “listos” para ser analizados y, si se tiene el valor suficiente, publicados.

Una vez llegué a una versión estable de este “sistema”, es decir, todos los test generan ficheros con sentido, hice un fichero .tar.gz con todo el directorio zubmark/, que contiene, además de los ficheros que ejecutan los test, los diversos ficheros asociados (imágenes, vídeos, sonidos, etc).

El siguiente paso, por cada distribución y kernel es crear un usuario nuevo y vaciar su directorio home de cualquier fichero de configuración y descomprimir ahí el .tar.gz con el test.

Sólo queda reiniciar el ordenador arrancando la distribución o kernel a probar, hacer login con el usuario de prueba, arrancar Xfree86, abrir un xterm, ejecutar zubmark.sh y esperar.

He procurado tener el mínimo número de demonios ejecutándose en cada una de las tres ejecuciones de zubmark.sh.

Las mediciones de tiempo y memoria en aquellas pruebas que las necesiten las hace el programamemtime, que calcula los tiempos de usuario, de sistema y real, así como la cantidad de memoria estimada que ha usado el proceso.

6 Detalles sobre las instalaciones

La fecha de ejecución de las pruebas fue el 19/12/2003 y aunque los sistemas no estaban actualizados de ese mismo día, sí de una semana antes, aproximadamente.

Las dos distribuciones están instaladas en diferentes particiones del mismo disco duro. Con la diferencia de que la Debian monta /home desde otra partición diferente a la raíz. En Gentoo todo está instalado sobre la misma partición.

Ambas comparten la misma partición swap de 256Mb aproximadamente.

Otro punto en común para las dos es el kernel. Las dos arrancan con mismo kernel 2.4.23 estándar compilado con mis opciones favoritas y recibiendo prácticamente los mismos parámetros desde LILO (ver diferencias en los dmesgs). Los módulos que se cargan también son los mismos en los dos casos.

En lo que respecta al kernel 2.6.0, las opciones son equivalentes a las del 2.4.23.

Gentoo está instalada desde Stage1 a partir de esta imagen: x86-basic-1.4-20030911.iso

En su /etc/make.conf puse las siguientes líneas:

```
USE="X gtk gnome alsa acpi 3dnow avi bonobo dvd gphoto2 gtk2 mmx nls oggvorbis -qt  
sse truecrypt xosd"
```

```
CHOST="i686-pc-linux-gnu"
```

```
CFLAGS="-march=athlon-xp -m3dnow -msse -mfpmath=sse -mmmx -fforce-addr -funroll-  
loops -ffast-math -O3 -pipe -fomit-frame-pointer -falign-functions=4"
```

Y sí, ya sé que no está la colección completa de optimizaciones experimentales.

7 Pruebas

7.1 Condiciones iniciales

Antes y después de ejecutar los tests, el script `zubmark.sh` guarda en un fichero de texto los siguientes datos:

- Fecha (date)
- Informe de memoria ocupada (free)
- Carga del sistema (uptime)
- Listado de los procesos del sistema (`ps -e l`)

Aquí va un resumen de las condiciones iniciales en cada una de las tres pruebas:

Debian Sid 2.4.23

```
vie dic 19 17:52:23 CET 2003  
                total      used      free      shared    buffers    cached  
Mem:            257072     54060     203012         0        4556     26600  
-/+ buffers/cache:  22904     234168  
Swap:           265032         0     265032  
17:52:23 up 1 min,  2 users,  load average: 0.17, 0.09, 0.03  
UID    PID  PRI  NI   VSZ  RSS  COMMAND  
  0      1   8   0  1484  488  init [2]  
  0      2   9   0    0    0  [keventd]  
  0     19  19  19    0    0  [ksoftirqd_CPU0]  
  0     19   9   0    0    0  [kswapd]  
  0     19   9   0    0    0  [bdflush]  
  0     19   9   0    0    0  [kupdated]  
  0     19   9   0    0    0  [kjournald]  
  0     83   9   0    0    0  [kjournald]  
  0     84   9   0    0    0  [kjournald]  
  0    184   9   0  1536  612  /sbin/syslogd  
  0    188   9   0  2160 1284  /sbin/klogd  
  0    232   9   0  1516  520  /usr/sbin/inetd  
1003   269   9   0  4440 1652  -bash  
  0    270   9   0  1484  476  /sbin/getty 38400 tty2  
  0    271   9   0  1484  476  /sbin/getty 38400 tty3  
  0    272   9   0  1484  476  /sbin/getty 38400 tty4  
1003   297   9   0  4124 1244  /bin/sh /usr/bin/X11/startx  
1003   310   9   0  2360  644  xinit /home/dres/.xinitrc -- /usr/X11R6/lib/X11/xinit/xserverrc  
  0    311   8 -10 279952 14120 /usr/bin/X11/X -dpi 100  
1003   315  10   0  3048 1680  icewm  
1003   316  11   0  6756 2612  /usr/bin/X11/xterm  
1003   317  11   0  5056 2324  bash  
1003   330  16   0  4116 1216  /bin/sh ./zubmark.sh  
1003   334  19   0  3316 1588  ps -e l
```

Gentoo 1.4 2.4.23

```
Fri Dec 19 18:51:26 CET 2003  
                total      used      free      shared    buffers    cached  
Mem:            257072     50768     206304         0        2336     28184  
-/+ buffers/cache:  20248     236824  
Swap:           265032         0     265032  
18:51:26 up 1 min,  2 users,  load average: 0.14, 0.05, 0.02  
UID    PID  PRI  NI   VSZ  RSS  COMMAND  
  0      1   9   0  1328  492  init [3]  
  0      2   9   0    0    0  [keventd]  
  0     19  19  19    0    0  [ksoftirqd_CPU0]  
  0     19   9   0    0    0  [kswapd]
```

```

0      5   9   0   0   0   [bdflush]
0      6   9   0   0   0   [kupdated]
0      7   9   0   0   0   [kjournald]
0    367   9   0   0   0   [kjournald]
0    368   9   0   0   0   [kjournald]
0    372   9   0   0   0   [khubd]
0    917   9   0 2016  952  login -- zub
0    918   9   0 1448  652  /sbin/agetty 38400 tty2 linux
0    919   9   0 1448  652  /sbin/agetty 38400 tty3 linux
0    920   9   0 1448  652  /sbin/agetty 38400 tty4 linux
0    921   9   0 1448  652  /sbin/agetty 38400 tty5 linux
0    922   9   0 1448  652  /sbin/agetty 38400 tty6 linux
1000  923   9   0 2232 1256  -bash
1000  928   9   0 2032  972  /bin/sh /usr/X11R6/bin/startx
1000  940   9   0 2776  736  xinit /home/zub/.xinitrc -- -nolisten tcp -deferglyphs 16
0    941  12   0 280352 1434 /etc/X11/X :0 -nolisten tcp -deferglyphs 16
1000  945  14   0  6084 3632  icewm
1000  946  16   0  6008 2500  xterm
1000  947  17   0  2228 1264  bash
1000  949  19   0  2032  948  /bin/sh ./zubmark.sh
1000  953  18   0  2496  640  ps -e l

```

Debian Sid 2.6.0

```

vie dic 19 17:00:50 CET 2003
total      used      free      shared      buffers      cached
Mem:      256436      53688      202748      0      3820      25808
-/+ buffers/cache:      24060      232376
Swap:      265032      0      265032
17:00:50 up 1 min,  2 users,  load average: 0.33, 0.16, 0.06
UID      PID      PRI      NI      VSZ      RSS      COMMAND
0         1        16      0      1516      512      init [2]
0         2        34      19      0         0      [ksoftirqd/0]
0         3         5     -10      0         0      [events/0]
0         4         5     -10      0         0      [kblockd/0]
0         5        25      0      0         0      [khubd]
0         6        25      0      0         0      [pdflush]
0         7        15      0      0         0      [pdflush]
0         8        25      0      0         0      [kswapd0]
0         9        10     -10      0         0      [aio/0]
0        10       22      0      0         0      [kseriod]
0        11       15      0      0         0      [kjournald]
0        86       15      0      0         0      [kjournald]
0        87       18      0      0         0      [kjournald]
0       194       16      0     1576      628      /sbin/syslogd
0       198       15      0     2404     1416      /sbin/klogd
0       245       18      0     1556      536      /usr/sbin/inetd
1003    282       17      0     4476     1664      -bash
0       283       17      0     1520      468      /sbin/getty 38400 tty2
0       284       17      0     1520      468      /sbin/getty 38400 tty3
0       285       17      0     1520      468      /sbin/getty 38400 tty4
1003    287       18      0     4152     1248      /bin/sh /usr/bin/X11/startx
1003    300       17      0     2392      656      xinit /home/dres/.xinitrc -- /usr/X11R6/lib/X11/xinit/xserverrc
0       301         5     -10    279984    14128      /usr/bin/X11/X -dpi 100
1003    308       15      0     3080     1676      icewm
1003    309       16      0     6784     2632      /usr/bin/X11/xterm
1003    310       16      0     5044     2284      bash
1003    324       18      0     4148     1224      /bin/sh ./zubmark.sh
1003    328       19      0     2316      660      ps -e l

```

Puede verse que los procesos en ejecución son básicamente los mismos en los tres casos, salvo syslogd, klogd e inetd que sólo están en los Debian. La memoria libre/ocupada es también muy similar, si bien da la impresión de que los mismos procesos ocupan menos memoria en el ps correspondiente a Gentoo.

7.2 Resultados

Aquí van por fin los resultados cada uno de los tres sistemas. El orden en el que aparecen aquí es el orden de ejecución.

Sólo he tenido paciencia para poner 2 repeticiones de cada uno de los test. Viendo que en general no varían mucho, supongo que sea suficiente. Al ser sólo dos valores no me molestaré en hacer ningún cálculo estadístico como medias, valores máximos, mínimos, y demás.

Al final de cada test escribiré una conclusión que contendrá opiniones personales, probablemente equivocadas.

7.2.1 Xfree86

Tests

```
x11perf -repeat 1 -scroll100
x11perf -repeat 1 -putimage100
x11perf -repeat 1 -shmput100
x11perf -repeat 1 -circulate
x11perf -repeat 1 -move
x11perf -repeat 1 -resize
```

Configuración

La configuración del servidor X es la misma en los tres casos. La resolución usada es 1024x768 y 24 bits de color.

Resultado

	Debian 2.4.23	Gentoo	Debian 2.6.0
scroll100	20300.0/sec	20300.0/sec	20300.0/sec
	20300.0/sec	20300.0/sec	20300.0/sec
putimage100	5740.0/sec	5540.0/sec	5680.0/sec
	5750.0/sec	5470.0/sec	5670.0/sec
shmput100	5750.0/sec	5750.0/sec	5750.0/sec
	5750.0/sec	5750.0/sec	5750.0/sec
circulate	27600.0/sec	29900.0/sec	28200.0/sec
	25600.0/sec	29400.0/sec	29800.0/sec
move	27100.0/sec	23800.0/sec	25800.0/sec
	25900.0/sec	23700.0/sec	26600.0/sec
resize	31800.0/sec	29300.0/sec	30900.0/sec
	30700.0/sec	29300.0/sec	31500.0/sec

Versiones

Pego las líneas relevantes de /var/log/XFree86.0.log. En los tres casos uso el driver propietario de Nvidia en su última versión estable el día del test. En el caso del kernel 2.6.0 con su patch correspondiente.

Debian

```
XFree86 Version 4.3.0.1 (Debian 4.3.0-0pre1v4 20031024231956 aigarius@namatevs.lv)
Release Date: 15 August 2003
X Protocol Version 11, Revision 0, Release 6.6
Build Operating System: Linux 2.4.22 i686 [ELF]
Build Date: 25 October 2003
(II) NVIDIA XFree86 Driver 1.0-4496 Wed Jul 16 19:06:03 PDT 2003
```

```
x11perf - X11 performance program, version 1.5
```

Gentoo

```
XFree86 Version 4.3.0
Release Date: 27 February 2003
X Protocol Version 11, Revision 0, Release 6.6
Build Operating System: Linux 2.4.22 i686 [ELF]
Build Date: 24 October 2003
(II)NVIDIA XFree86 Driver 1.0-4496 Wed Jul 16 19:06:03 PDT 2003
```

```
x11perf - X11 performance program, version 1.5
```

Conclusión

Los test *scroll100* y *shmput100* dan resultados idénticos en los tres sistemas lo que quiere decir que el rendimiento depende sobre todo del hardware acelerador de la tarjeta gráfica, con lo que no importa mucho ni la distribución ni el kernel.

En el test *putimage100* Gentoo queda un 4% por detrás de Debian con 2.4. El kernel 2.6 da unos resultados muy similares al 2.4 en Debian, aunque marginalmente peores.

Gentoo se adelanta en *circulate* un 10% sobre Debian con 2.4 y un 2% sobre 2.6.

Debian con cualquiera de los dos kernel es un 10% más rápida en *move* que Gentoo.

Igual situación en *resize*, aunque esta vez sólo con un 5% de ventaja.

Debian con kernel 2.4 es el más rápido en 3 test, algo por delante de ella misma con kernel 2.6, que parece se comporta de forma algo más consistente.

Hay que decir que Debian arranca el servidor X con un valor nice -10, esto es, la prioridad del servidor X será mayor en general en Debian que en mi instalación de Gentoo.

No parece que sea muy ventajoso en este caso el haber compilado XFree86 con optimizaciones.

7.2.2 OpenGL

Test

glxgears

Resultado

La salida de glxgears es de una línea cada 5 segundos con el número de frames dibujados. Cada línea de la siguiente tabla es el número máximo de frames obtenido en cada una de las dos repeticiones del test.

Debian 2.4.23	Gentoo	Debian 2.6.0
913,600	910,800	913,000
913,600	910,200	913,000

Versiones

En los tres casos la información que da glxgears -info es la misma:

GL_RENDERER = GeForce2 MX/AGP/SSE/3DNOW!

GL_VERSION = 1.4.0 NVIDIA 44.96

Conclusión

2.4 se comporta marginalmente mejor que 2.6. Gentoo se ve mínimamente penalizada de nuevo, en teoría, por su diferente valor de nice por defecto. No parece que las optimizaciones hayan valido para nada aquí. La mayoría del trabajo es hardware.

7.2.3 Visión de JPEG

Test

gqview -s -f jpeg/

El test muestra un slideshow a toda pantalla de las 50 imágenes jpeg 1600x1200 contenidas en el directorio jpeg/.

Configuración

Previamente a la ejecución de los tests se copió el mismo directorio con una configuración de gqview en los homes de los usuarios. Se establece que el tiempo que tiene que mostrarse cada imagen es 1s y que la calidad del algoritmo de escalado sea la mayor posible.

Evidentemente el tiempo "elapsed" en este caso no tiene importancia y es aproximadamente igual a los 50 segundos que se desperdician esperando. No encontré la forma de poner un valor de 0 al retardo en el slideshow.

Resultado

Esta tabla, al igual que otras que usaré más adelante, muestran la salida del comando memtime en las dos repeticiones del test.

Debian 2.4.23					Gentoo					Debian 2.6.0				
user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS
26.30	1.51	53.56	46168	40700	25.24	1.33	52.69	44224	40228	26.78	1.39	53.03	46208	40636
26.38	1.20	52.80	46168	40620	24.84	1.44	52.69	44216	40304	26.59	1.34	51.93	46208	40752

Versiones

Debian

GQview 1.3.5

Versión GTK2: 2.2.4

Versión libjpeg: libjpeg.so.62 (libc6) => /usr/lib/libjpeg.so.62

Gentoo

GQview 1.3.5

Versión GTK2: 2.2.4

Versión libjpeg: libjpeg.so.62 (libc6) => /usr/lib/libjpeg.so.62

Conclusión

Gentoo es un 5% más rápida aquí que Debian con 2.4.23. Algo más aún con respecto al kernel 2.6.

7.2.4 Redimensión JPEG

Test

```
convert -geometry "388800@" -quality 80
```

Se usa el anterior comando (una de las utilidades de Imagemagick) como parte de un script para convertir 50 jpeg 1600x1200 a 720x540.

Resultado

Debian 2.4.23					Gentoo					Debian 2.6.0				
user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS
58.15	1.61	63.25	35152	28391	46.16	1.4	50.05	33641	27860	59.44	1.69	63.74	35003	28384
58.28	1.43	62.69	35152	28391	46.12	1.41	49.5	33641	27860	59.35	1.69	63.21	35003	28384

Versiones

Debian

Version: ImageMagick 5.5.7 06/10/03 Q16 <http://www.imagemagick.org>

Gentoo

Version: ImageMagick 5.5.6 04/01/03 Q16 <http://www.imagemagick.org>

Conclusión

Gentoo se destaca claramente en este test, tardando 12 segundos menos que la más rápida de las Debian, lo que viene a ser en torno a 20% más rápido.

El mismo test tarda 1 segundo más en completarse bajo el kernel 2.6.0 que sobre 2.4.23.

7.2.5 Vídeo

Test

```
mplayer -vo xv -benchmark -nosound -quiet
```

El fichero con el vídeo reproducido es un AVI de 6:09 minutos con estas características:

```
$ file video.avi
```

```
video.avi: RIFF (little-endian) data, AVI, 360 x 240, 29.97 fps, video: DivX 5, audio: MPEG-1 Layer 3 (stereo, 44100 Hz)
```

Configuración

El directorio de configuración de mplayer es el mismo para cada uno de los usuarios del test.

Resultado

Mplayer con la opción -benchmark da además información acerca del tiempo que emplea en descodificar (VC) y mostrar (VO) el vídeo.

Debian 2.4.23					Gentoo					Debian 2.6.0				
user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS
5.69	0.44	10.12	28908	13212	5.39	0.37	9.45	20384	11280	5.85	0.46	10.10	28672	13212
5.72	0.41	8.90	28908	13212	5.40	0.38	8.57	20384	11280	5.93	0.38	9.19	28672	13212
VC		VO			VC		VO			VC		VO		
5,590s		2,877s			5.324s		2.837s			5,803s		2,921s		
5,492s		2,849s			5.192s		2.833s			5,766s		2,898s		

Versiones

En este caso el binario usado en las pruebas Debian no es una versión precompilada.

Debian

MPlayer 1.0pre3-3.3.3 (C) 2000-2003 MPlayer Team

CPU: Advanced Micro Devices Athlon 4 /Athlon MP/XP Palomino 1463 MHz (Family: 6, Stepping: 2)

Detected cache-line size is 64 bytes

CPUflags: MMX: 1 MMX2: 1 3DNow: 1 3DNow2: 1 SSE: 1 SSE2: 0

Compiled for x86 CPU with extensions: MMX MMX2 3DNow 3DNowEx SSE

Gentoo

MPlayer 1.0pre3-3.2.3 (C) 2000-2003 MPlayer Team

CPU: Advanced Micro Devices Athlon 4 /Athlon MP/XP Palomino 1463 MHz (Family: 6, Stepping: 2)

Detected cache-line size is 64 bytes

CPUflags: MMX: 1 MMX2: 1 3DNow: 1 3DNow2: 1 SSE: 1 SSE2: 0

Compiled for x86 CPU with extensions: MMX MMX2 3DNow 3DNowEx SSE

Conclusión

Gentoo vuelve a ser quien más rápido termina con la tarea, en torno a un 5% más rápido que Debian con 2.4. La versión con 2.6.0 es la más lenta, aunque sólo un 1% por detrás de 2.4.23.

Parece que la tarea en la que Gentoo se adelanta es en la decodificación del vídeo, con 0.3 segundos menos que la Debian más rápida. El tiempo empleado en la salida parece totalmente dependiente del hardware.

7.2.6 Codificación MP3

Test

```
lame -S -b 192 -h
```

El fichero PCM del que se obtendrá un mp3 es un wav de 44Mb del que file dice lo siguiente:

```
$ file -L test.wav
```

```
test.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 44100 Hz
```

Resultado

Debian 2.4.23					Gentoo					Debian 2.6.0				
user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS
18.99	0.34	20.02	3680	2196	19.57	0.15	20.23	3612	2208	19.48	0.32	19.89	3716	2216
19.05	0.22	19.36	3680	2196	19.55	0.20	19.80	3612	2208	19.41	0.23	19.73	3716	2216

Versiones

En las tres pruebas la versión es:

LAME version 3.93 MMX (<http://www.mp3dev.org/>)

Como en el caso de mplayer, lame también ha sido compilado desde fuente en los Debian.

Conclusión

Gentoo queda en último lugar esta vez, aunque por un margen mínimo de sobre 1%-2%.

No hay diferencias significativas en este test entre las diversas combinaciones distribución-kernel.

7.2.7 Descompresión de archivos

Test

```
tar xjf
```

El fichero descomprimido en este test ha sido “linux-2.4.23.tar.bz2”, de 29Mb.

Resultado

Debian 2.4.23					Gentoo					Debian 2.6.0				
user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS
25.96	2.42	46.14	3908	800	25.42	2.11	44.21	1988	704	26.33	2.50	37.54	3632	796
25.72	2.44	40.36	3908	800	25.10	2.43	37.94	1988	704	26.33	2.53	33.86	3632	796

Versiones

Misma versión en los tres casos:

```
tar (GNU tar) 1.13.25
```

```
bzip2, a block-sorting file compressor. Version 1.0.2, 30-Dec-2001.
```

Conclusión

El ganador en tiempo absoluto (elapsed) ha sido Debian con kernel 2.6.0, y con una buena diferencia sobre las demás. Calculando a partir de los tiempos medios: 5.3 y 7.5 segundos más rápida que Gentoo y Debian con 2.4, respectivamente. Aproximadamente un 15% más rápido.

Sin embargo la velocidad de acceso a disco es muy importante en esta prueba, y aunque en elapsed Debian con 2.6 es la que mejor se comporta, el tiempo de CPU usado por el usuario es menor tanto en Debian con 2.4 como, especialmente, en Gentoo, indicativo de que si no fuera por el acceso a disco probablemente lo harían más rápido.

7.2.8 Rendimiento Disco Duro, aplicación

Test

```
bonnie++
```

Resultado

Debian (2.4.23)

```
Version 1.03      -----Sequential Output----- --Sequential Input- --Random-
-Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine      Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
gothmog      496M 15007 96 21320 10 10168 3 16199 77 22883 5 197.2 0
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
16 1624 99 +++++ +++ +++++ +++ 1693 99 +++++ +++ 4170 98
```

Gentoo

```
Version 1.03      -----Sequential Output----- --Sequential Input- --Random-
-Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine      Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
goth         496M 10278 85 23579 14 9289 4 15492 77 22152 6 192.0 0
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
16 1335 98 +++++ +++ +++++ +++ 1527 99 +++++ +++ 3917 96
```

Debian (2.6.0)

```
Version 1.03      -----Sequential Output----- --Sequential Input- --Random-
-Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine      Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
gothmog      496M 18429 94 23927 13 9520 4 15126 72 22587 5 159.0 0
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
```

Versiones

En los tres casos la versión de bonnie++ es:

Version: 1.03

Conclusión

No parece haber diferencias significativas a igualdad de hardware aquí.

7.2.9 Rendimiento Disco Duro, bajo nivel

Test

```
hdparm -tT /dev/hda
```

Configuración

La configuración del disco duro es la misma en los tres casos. Este es un “resumen” de la salida de hdparm -Iv:

```
/dev/hda:
multcount      = 16 (on)
IO_support     = 1 (32-bit)
unmaskirq     = 1 (on)
using_dma      = 1 (on)
keepsettings   = 0 (off)
readonly      = 0 (off)
readahead     = 8 (on)
geometry      = 5005/255/63, sectors = 80418240, start = 0

ATA device, with non-removable media
powers-up in standby; SET FEATURES subcmd spins-up.
  Model Number:      IC35L040AVVA07-0
  Serial Number:     VNC220A2G3GJ3C
  Firmware Revision: VA20A50K

Standards:
  Used: ATA/ATAPI-5 T13 1321D revision 1
  Supported: 5 4 3 2 & some of 6

Configuration:
  Logical          max      current
  cylinders        16383  16383
  heads           16        16
  sectors/track   63        63
  --
  CHS current addressable sectors: 16514064
  LBA user addressable sectors: 80418240
  device size with M = 1024*1024: 39266 MBytes
  device size with M = 1000*1000: 41174 MBytes (41 GB)

Capabilities:
  LBA, IORDY(can be disabled)
  bytes avail on r/w long: 52      Queue depth: 32
  Standby timer values: spec'd by Standard, no device specific minimum
  R/W multiple sector transfer: Max = 16 Current = 16
  Advanced power management level: unknown setting (0x0000)
  Recommended acoustic management value: 128, current value: 254
  DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 *udma5
  Cycle time: min=120ns recommended=120ns
  PIO: pio0 pio1 pio2 pio3 pio4
  Cycle time: no flow control=240ns IORDY flow control=120ns
```

Resultado

Debian (2.4.23)

```
/dev/hda:
Timing buffer-cache reads: 1060 MB in 2.00 seconds = 530.00 MB/sec
Timing buffered disk reads: 134 MB in 3.04 seconds = 44.08 MB/sec
```

```
/dev/hda:
Timing buffer-cache reads: 1044 MB in 2.00 seconds = 522.00 MB/sec
Timing buffered disk reads: 136 MB in 3.01 seconds = 45.18 MB/sec
```

Gentoo

```
/dev/hda:
```

```
Timing buffer-cache reads: 1076 MB in 2.00 seconds = 538.00 MB/sec
Timing buffered disk reads: 134 MB in 3.02 seconds = 44.37 MB/sec
```

/dev/hda:

```
Timing buffer-cache reads: 1012 MB in 2.00 seconds = 506.00 MB/sec
Timing buffered disk reads: 132 MB in 3.00 seconds = 44.00 MB/sec
```

Debian (2.6.0)

/dev/hda:

```
Timing buffer-cache reads: 1040 MB in 2.01 seconds = 518.52 MB/sec
Timing buffered disk reads: 136 MB in 3.02 seconds = 45.07 MB/sec
```

/dev/hda:

```
Timing buffer-cache reads: 1040 MB in 2.00 seconds = 519.04 MB/sec
Timing buffered disk reads: 134 MB in 3.01 seconds = 44.57 MB/sec
```

Versiones

En los tres casos:

hdparm v5.4

Conclusión

Sólo un 1% de diferencia entre las medias. Como en el caso del test anterior, las diferencias no son significativas.

7.2.10 Compilación de un kernel Linux

Test

```
make oldconfig dep bzImage
```

Se ejecuta ese comando dentro de un directorio que contiene las fuentes del kernel 2.4.23 y un fichero .config con una configuración.

Resultado

Debian 2.4.23					Gentoo					Debian 2.6.0				
user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS	user	system	elapsed	VSize	RSS
238.62	19.61	273.76	3856	924	227.61	17.26	255.92	1996	848	236.00	23.63	273.92	3572	900
238.05	20.09	260.23	3856	924	227.20	17.73	251.59	1996	848	236.03	23.54	261.07	3572	900

Versiones

Debian

GNU Make 3.80

gcc (GCC) 3.3.3 20031206 (prerelease) (Debian)

Gentoo

GNU Make 3.80

gcc (GCC) 3.2.3 20030422 (Gentoo Linux 1.4 3.2.3-r3, propolice)

Conclusión

Gentoo compila el kernel un 5% más rápido que cualquiera de las dos Debian aquí probadas, tardando 13 segundos menos.

La versión de gcc usada es diferente en las dos distribuciones, pero una prueba posterior para calcular la diferencia de prestaciones entre 3.2 y 3.3 arrojó un resultado 0.5 segundos a favor de 3.2 en una compilación de kernel. Sobre un tiempo total de 330 segundos, la diferencia es lo bastante pequeña como para ser despreciable.

8 Conclusiones

8.1 Debian y Gentoo

En general parece que Gentoo consigue terminar antes con la mayoría de las tareas en comparación con Debian y sus binarios precompilados para i386. Algo así como un 5% antes.

Parece que sólo se obtiene un aumento espectacular del rendimiento si el programa incluye código que explícitamente use instrucciones MMX, 3DNow!, SSE, partes de código ensamblador i586...

En ese caso, si en Debian se compilan esas “partes críticas”, el comportamiento es similar al de Gentoo, aunque aún así, tal vez un poco por debajo.

8.2 *Linux 2.4.23 y Linux 2.6.0*

En la mayoría de los test Debian con el kernel 2.6 ha sido el más lento aunque con poca diferencia sobre 2.4, un 1%.

Aunque cronómetro en mano los programas se comporten (un poco) peor, compilado con la opción “Preemptible Kernel”, como es el caso de esta prueba, el sistema responderá mejor cuando se esté manejando una aplicación interactiva en situaciones de carga alta. En resumen: es un poco más lento, pero parece más rápido.

9 Sobre las pruebas

El presente documento está disponible en formatos swx (Documento de texto OpenOffice), pdf y html.

Tengo disponibles para quien lo quiera todos los scripts (bash) y resultados de estos tests en forma de ficheros de texto, con información que he omitido para no hacer aún más aburrido este documento.

Si alguien tiene la (enfermiza) curiosidad necesaria puede pedirme esos ficheros llenos de números, salidas redireccionadas de comandos y dmesgs en *dzub europe.com* (Cambiar espacio por @).

Claro está también puedes usar esa dirección para enviarme los comentarios, críticas, flames y amenazas que no puedas evitar expresar.