

wscons en NetBSD

Autor: [Julio Merino \(aka Slink\)](#)

Fecha: 12 de enero de 2002

wscons es el dispositivo que gestiona la consola (monitor, teclado y ratón) en NetBSD. Éste me todo ha sido adaptado también por OpenBSD, aunque es un tanto diferente. Lo que voy a explicaros aquí se refiere específicamente a NetBSD-current (próxima 1.6) aunque creo que también será aplicable a la actual estable, NetBSD 1.5.2

Conceptos básicos

wscons viene incluido por defecto en el kernel, así que no tendréis que preocuparos por incluirlo. Aparte de eso, hemos de conocer los programas que NetBSD nos ofrece para su control; éstos són, resumiendo, `wscnscfg` y `wscnctl`.

`wscnscfg` no lo usaremos directamente, aunque en algún caso puede interesarnos. Éste programa es el que "ejecuta" la mayor parte del archivo `/etc/wscons.conf`. Además se usa para escoger qué teclado, pantalla virtual o ratón usaremos en cada momento, ya que podríamos tener varios de cada. Para más información sobre él, consultar `wscnscfg(8)`.

`wscnctl` es el programa más interesante de los dos, y el que nos permitirá configurar los valores específicos de los dispositivos conectados por `wscnscfg`. Su funcionamiento es similar al de `sysctl`. Consultar `wscnctl(8)`.

Podéis obtener un listado completo de todas las variables que se pueden configurar ejecutando:

```
# wscnctl -a
```

Para modificar cualquiera de las variables que nos muestra el comando anterior haremos lo siguiente:

```
# wscnctl -w nombre_variable=valor
```

Configurar el teclado

Lo que más necesitaremos en un primer momento es configurar el tipo de nuestro teclado. Podemos usar dos tipos para el español: el `es` y el `es.metaesc`. El primero no permite el uso de la tecla `[ALT]` correctamente, mientras que el segundo sí. Por lo tanto, usaremos éste último.

Para seleccionar el teclado manualmente podemos hacer:

```
# wscnctl -w keyboard.encoding=es.metaesc
```

Aunque seguramente nos interesará que se configure cada vez que arranquemos el sistema. Para ello, añadiremos lo siguiente al archivo `/etc/wscons.conf`:

```
encoding es.metaesc
```

Bien, así ya tenemos el teclado configurado. De todos modos aún podemos cambiar otros detalles. Personalmente me molesta la velocidad de repetición de las pulsaciones que viene por defecto (demasiado lenta). Ésto podemos cambiarlo haciendo lo siguiente (ajustar los valores como queráis):

```
# wscnctl -w repeat.dell=400
# wscnctl -w repeat.deln=50
```

Cuidado con poner valores demasiado bajos en `repeat.dell`; si lo hacéis ¡no podréis escribir con normalidad!

Configurar las consolas virtuales

Por defecto, NetBSD lleva muy pocas consolas virtuales, sólo tres. A mi me gusta disponer de unas cuantas más, así que vamos a configurar ocho.

Tenemos dos modos de configuración. Podemos decirle al kernel cuántas queremos o bien podemos configurarlas con `wscnscfg`. Vamos a usar el segundo método ya que así podremos cambiarlas cuando queramos. Tampoco es necesario preocuparse demasiado ya que este programa puede modificar los valores del kernel.

Para empezar vamos a configurar el kernel para que sólo habilite una consola virtual. Ésto no es necesario pero nos evitaremos molestos mensajes al inicio. Editamos nuestro archivo de configuración del kernel y le añadimos:

```
options          WSDISPLAY_DEFAULTSCREENS=1
```

Una vez modificado, recompilamos e instalamos el nuevo kernel. De momento no reiniciéis si no queréis.

Ahora tenemos que modificar el archivo `/etc/wscnscfg.conf`. Buscaremos las siguientes líneas:

```
screen 1      -      vt100
screen 2      -      vt100
screen 3      -      vt100
screen 4      -      -
```

La segunda columna especifica el número de terminal virtual. La tercera, el tipo de pantalla (podemos especificar su tamaño, aunque lo dejaremos tal y como está). Y la cuarta especifica el tipo de terminal, que tampoco modificaremos. Vamos a añadir soporte para 4 consolas virtuales más; ésto sólo habilitará más consolas, pero aún serán inusables. Modificaremos éstas líneas y añadiremos las que falten, de manera que nos quede algo como:

```
screen 1      -      vt100
screen 2      -      vt100
screen 3      -      vt100
screen 4      -      vt100
screen 5      -      vt100
screen 6      -      vt100
screen 7      -      vt100
```

Si os fijáis veréis la consola 0 deshabilitada. Ésto es normal ya que el kernel se encarga de configurarla.

Bien, ahora sólo nos falta habilitar el login en las nuevas consolas. Editaremos el `/etc/ttyx`. Por defecto encontraremos lo siguiente:

```
console "/usr/libexec/getty Pc"          vt100    on secure
ttyE0   "/usr/libexec/getty Pc"          vt220    off secure
ttyE1   "/usr/libexec/getty Pc"          vt220    off secure
ttyE2   "/usr/libexec/getty Pc"          vt220    off secure
ttyE3   "/usr/libexec/getty Pc"          vt220    off secure
```

Que tenemos que dejar así:

```
console "/usr/libexec/getty Pc"          vt100    off secure
ttyE0   "/usr/libexec/getty Pc"          vt220    on secure
ttyE1   "/usr/libexec/getty Pc"          vt220    on secure
ttyE2   "/usr/libexec/getty Pc"          vt220    on secure
ttyE3   "/usr/libexec/getty Pc"          vt220    on secure
ttyE4   "/usr/libexec/getty Pc"          vt220    on secure
ttyE5   "/usr/libexec/getty Pc"          vt220    on secure
ttyE6   "/usr/libexec/getty Pc"          vt220    on secure
ttyE7   "/usr/libexec/getty Pc"          vt220    on secure
```

Fijaros otra vez que aquí deshabilitamos console y la configuramos como ttyE0.

Si queré is ver los cambios inmediatamente, sin reiniciar, podé is ejecutar:

```
# /etc/rc.d/wscons start
# kill -HUP 1
```

Otras variables para el kernel

Si miramos el archivo de configuración genérico del kernel veremos que wscons admite varias variables de configuración. Aquí os muestro las que uso yo:

```
options          WSEMUL_VT100                # VT100 / VT220
options          WS_KERNEL_FG=WSCOL_GREEN
options          WS_KERNEL_BG=WSCOL_BLACK
options          WSDISPLAY_COMPAT_PCVT        # emulate some i
options          WSDISPLAY_COMPAT_SYSCONS    # emulate some i
options          WSDISPLAY_COMPAT_USL        # VT handling
options          WSDISPLAY_COMPAT_RAWKBD     # can get raw sc
options          WSDISPLAY_DEFAULTSCREENS=1
options          PCKBD_LAYOUT="(KB_ES)"
options          PCDISPLAY_SOFTCURSOR
options          XSERVER                # X server supp
```

No voy a comentarlas aquí ya que su funcionamiento es obvio. Sino, consultar [/sys/arch/\[ARCH\]/conf/GENERIC](#), siendo *[ARCH]* el nombre de tu arquitectura, que seguramente será *i386*.
