FreeBSD y las unidades de estado sólido (SSD)

John Kozubik <john@kozubik.com>

Revision: 74683b1030

2019-06-16 22:20:04 +0000 by Sergio Carlavilla Delgado.

Resumen

Este artículo trata sobre el uso de discos de estado sólido en FreeBSD para crear sistemas embebidos.

Los sistemas embebidos tienen la ventaja de una mayor estabilidad por la falta de partes móviles (discos duros). Sin embargo, se debe tener en cuenta que generalmente el espacio disponible para el sistema y la durabilidad del medio de almacenamiento son menores.

Los temas específicos que se abordarán incluyen los tipos y atributos de los dispositivos de estado sólido adecuados para su uso como disco en FreeBSD, las opciones del kernel que son interesantes para dicho entorno, los mecanismos de `rc.initdiskless` que automatizan el inicio de dichos sistemas, la necesidad de sistemas de archivos de solo lectura y hacer sistemas de archivos desde cero. El artículo concluirá con algunas estrategias generales para entornos pequeños y de solo lectura de FreeBSD.

Tabla de contenidos

1. Dispositivos de disco de estado sólido

El alcance de este artículo se limitará a dispositivos de estado sólido basados en memoria flash. La memoria flash es una memoria de estado sólido (sin partes móviles) que no es volátil (la memoria mantiene los datos incluso después de que se hayan desconectado todas las fuentes de alimentación). La memoria flash puede soportar un enorme impacto físico y es bastante rápida (las soluciones de memoria flash que se tratan en este artículo son un poco más lentas que un disco duro EIDE en operaciones de escritura y mucho más rápidos en operaciones de lectura). Un aspecto muy importante de la memoria flash, cuyas repercusiones se tratarán más adelante, es que cada sector tiene una capacidad de reescritura limitada. Solo puede escribir, borrar y volver a escribir en un sector de la memoria flash varias veces antes de que quede permanentemente inutilizable. Aunque muchos productos de memoria flash mapean automáticamente los bloques defectuosos y algunos incluso distribuyen las operaciones de escritura de manera uniforme en toda la unidad, la verdad es que hay un límite en la cantidad de escrituras que se pueden hacer al dispositivo. Las unidades más competitivas tienen entre 1.000.000 y 10.000.000 millones de escrituras por sector en sus especificaciones. Esta cifra varía debido a la temperatura del ambiente.

Específicamente, discutiremos las unidades flash compactas compatibles con ATA, las cuales son bastante populares como medios de almacenamiento para cámaras digitales. Es de particular interés el hecho de que se conecten directamente al bus IDE y sean compatibles con el conjunto de comandos ATA. Por lo tanto, con un adaptador muy simple y de bajo coste, estos dispositivos se pueden conectar directamente al bus IDE en un ordenador. Una vez implementado de esta forma, los sistemas operativos como FreeBSD ven el dispositivo como un disco duro normal (aunque sea pequeño).

Existen otras soluciones de disco de estado sólido, pero su coste, opacidad y su relativa dificultad de uso los colocan más allá del alcance de este artículo.

2. Opciones del kernel

Algunas opciones del kernel son de especial interés para aquellos que crean un sistema FreeBSD embebido.

Todos los sistemas FreeBSD embebidos que utilizan memorias flash como disco para el sistema estarán interesados en utilizar discos y sistemas de archivos cargados en memoria. Debido al número limitado de escrituras que se pueden hacer en la memoria flash, el disco y los sistemas de archivos probablemente se montarán como de solo lectura. En este entorno, los sistemas de archivos como `/tmp` y `/var` se montan como sistemas de archivos en memoria para permitir que el sistema cree registros y actualice los contadores y los archivos temporales. Los sistemas

de archivos en memoria son un componente crítico para una implementación exitosa de FreeBSD en dispositivos de estado sólido.

Asegúrese de que existen las siguientes líneas en el archivo de configuración del kernel:

```
options         MFS             # sistema de archivos de memoria
options         MD_ROOT         # el dispositivo md puede ser usado potencialmente como
 dispositivo root
pseudo-device   md              # disco de memoria
```

3. El subsistema rc y los sistemas de archivos de solo lectura

La inicialización posterior al arranque de un sistema FreeBSD embebido es controlada por `/etc/rc.initdiskless`.

`/etc/rc.d/var` monta `/var` como sistema de archivos en memoria, crea una listado configurable de directorios en `/var` con el comando mkdir(1) y cambia los modos en algunos de esos directorios. En la ejecución de `/etc/rc.d/var`, otra variable de `rc.conf` entra en juego: `varsize`. `/etc/rc.d/var` crea una partición `/var` basándose en el valor de la variable en `rc.conf`:

```
varsize=8192
```

Recuerde que por defecto este valor está en sectores.

El hecho de que `/var` sea un sistema de archivos de lectura y escritura es una distinción importante, ya que la partición `/` (y cualquier otra partición que pueda tener en su medio flash) se debe montar como solo lectura. Recuerde que en la Sección 1, “Dispositivos de disco de estado sólido” detallamos las limitaciones de la memoria flash, específicamente, la capacidad de escritura limitada. La importancia de no montar sistemas de archivos en medios flash de lectura-escritura, y la importancia de no usar swap, no es exagerada. Un archivo swap en un sistema concurrido puede deteriorar un medio flash en menos de un año. Un logging intenso o la creación y destrucción de archivos temporales puede hacer lo mismo. Por lo tanto, además de quitar la entrada `swap` de su `/etc/fstab`, también debe cambiar el campo Options para cada sistema de archivos a `ro` de la siguiente forma:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1a             /               ufs     ro              1       1
```

Algunas aplicaciones en el sistema comenzarán a fallar inmediatamente como resultado de este cambio. Por ejemplo, cron no se ejecutará correctamente al faltar las cron tabs en `/var` creadas por `/etc/rc.d/var`, además, syslog y dhcp encontrarán problemas como resultado de montar el sistema de archivos como solo lectura y la falta de elementos en `/var` que ha creado `/etc/rc.d/var`. Sin embargo, esto son solo problemas temporales y se tratan, junto con las soluciones para la ejecución de otros programas de uso común en la Sección 5, “Estrategias para entornos pequeños y de solo lectura”.

Una cosa importante a recordar es que un sistema de archivos que fue montado como solo lectura con `/etc/fstab` puede ser montado como lectura-escrita en cualquier momento ejecutando el comando:

```
# /sbin/mount -uw partition
```

y se puede cambiar de nuevo a solo lectura con el comando:

```
# /sbin/mount -ur partition
```

4. Construyendo un sistema de archivos desde cero

Como las tarjetas compact-flash compatibles con ATA son vistas por FreeBSD como discos duros IDE estándar, en teoría se podría instalar FreeBSD desde la red usando floppies kern y mfsroot o desde un CD.

Sin embargo, incluso una pequeña instalación de FreeBSD que utilice procedimientos normales de instalación puede producir un sistema con un tamaño superior a 200 megabytes. Como la mayoría de la gente utilizará dispositivos de memoria flash más pequeños (128 megabytes se consideran razonablemente grandes - 32 o incluso 16 megabytes

son comunes), una instalación utilizando mecanismos normales no será posible - simplemente no hay suficiente espacio en el disco incluso para las instalaciones convencionales más pequeñas.

La forma más fácil de superar esta limitación de espacio es instalar FreeBSD utilizando medios convencionales en un disco duro normal. Una vez finalizada la instalación, reduzca el sistema operativo a un tamaño que se ajuste a su medio flash, y comprima el sistema de archivos completo en un fichero tar. Los siguientes pasos le guiarán en el proceso de preparación de una memoria flash para su sistema de archivos comprimido en un fichero tar. Recuerde que no estamos ejecutando una instalación normal, luego las operaciones como particionado, etiquetado, creación del sistema de archivos, etc. deben ejecutarse manualmente. Además de los disquetes kern y mfsroot, también necesitará usar el disquete fixit.

1. Particionando su dispositivo flash

   Después de arrancar con los disquetes kern y mfsroot, seleccione `custom` en el menú de instalación. En el menú de instalación personalizado, seleccione `partition`. En el menú de particiones, debe borrar todas las particiones existentes mediante la tecla d. Después de eliminar todas las particiones existentes, cree una partición utilizando la tecla c y acepte el valor predeterminado para el tamaño de la partición. Cuando se le pregunte el tipo de partición, asegúrese de que el valor esté establecido en `165`. Ahora escriba la tabla de particiones en el disco presionando w (es una opción oculta en esta pantalla). Si está utilizando una tarjeta compact flash compatible con ATA, debe elegir el FreeBSD Boot Manager. Ahora presione q para salir del menú de partición. Verá de nuevo el menú del gestor de arranque - repita la opción hecha anteriormente.

2. Creación de sistemas de archivos en su dispositivo de memoria flash

   Salga del menú de instalación personalizado y, en el menú de instalación principal, elija la opción `fixit`. Después de entrar en el entorno de fixit, escriba el siguiente comando:

   ```
   # disklabel -e /dev/ad0c
   ```

   En este punto, habrá accedido al editor vi guiado por el comando disklabel. A continuación, debe agregar una línea `a:` al final del archivo. La línea `a:` debería ser similar a la siguiente:

   ```
   a:      123456  0       4.2BSD  0       0
   ```

   Donde *123456* es exactamente el mismo número que la entrada `c:`. Básicamente, está duplicando la línea `c:` como `a:`, asegúrese de que el fstype es `4.2BSD`. Guarde el archivo y ciérrelo.

   ```
   # disklabel -B -r /dev/ad0c
   # newfs /dev/ad0a
   ```

3. Colocando su sistema de archivos en el medio flash

   Monte el medio flash recién preparado:

   ```
   # mount /dev/ad0a /flash
   ```

   Coloque esta máquina en la red para poder transferir nuestro archivo tar y extraerlo en nuestro sistema de archivos del medio flash. Un ejemplo de cómo hacerlo es:

   ```
   # ifconfig xl0 192.168.0.10 netmask 255.255.255.0
   # route add default 192.168.0.1
   ```

   Ahora que la máquina está en la red, transfiera su archivo tar. Es posible que se enfrente a un pequeño dilema en este punto - si su memoria flash tiene por ejemplo 128 megabytes, y su archivo tar tiene más de 64 megabytes, no podrá tener el archivo tar en el medio de flash al mismo tiempo que realiza la descompresión - se quedará sin espacio. Una solución a este problema, si está utilizando FTP, es descomprimir el archivo mientras se transfiere por FTP. Si realiza la transferencia de esta forma, nunca tendrá el archivo tar y los contenidos en el disco al mismo tiempo:

   ```
   ftp> get tarfile.tar "| tar xvf -"
   ```

Si su archivo tar está comprimido en gzip, puede hacerlo de esta forma:

```
ftp> get tarfile.tar "| zcat | tar xvf -"
```

Una vez que el contenido de su sistema de archivos comprimido por tar está en el sistema de archivos de la memoria flash, puede desmontar la memoria flash y reiniciar:

```
# cd /
# umount /flash
# exit
```

Suponiendo que configuró correctamente su sistema de archivos cuando lo construyó en su disco duro normal, (con sus sistemas de archivos montados en modo solo lectura, y con las opciones necesarias compiladas en el kernel) ahora se debería iniciar con éxito su sistema embebido FreeBSD.

5. Estrategias para entornos pequeños y de solo lectura

En la Sección 3, "El subsistema `rc` y los sistemas de archivos de solo lectura", se indicó que el sistema de archivos `/var` construido por `/etc/rc.d/var` y la presencia de un sistema de archivos raíz montado en modo solo lectura causa problemas con muchos paquetes de software utilizados en FreeBSD. En este artículo, se proporcionarán sugerencias para ejecutar con éxito cron, syslog, la instalación de ports y el servidor web Apache.

5.1. Cron

Tras el arranque, `/var` será llenado con `/etc/rc.d/var` usando la lista disponible en `/etc/mtree/BSD.var.dist`, por lo que `cron`, `cron/tabs`, `at` y algunos otros directorios estándar son creados.

Sin embargo, esto no resuelve el problema de mantener las cron tabs entre los reinicios. Cuando el sistema se reinicie, el sistema de archivos `/var` cargado en memoria desaparecerá y todas las cron tabs que tenga también desaparecerán. Por lo tanto, una solución sería crear las cron tabs para los usuarios que las necesiten; monte su sistema de archivos raíz `/` como lectura-escritura y copie las cron tabs a un lugar seguro, como `/etc/tabs`, a continuación, añada una entrada al final de `/etc/rc.initdiskless` que copie estas crontabs a `/var/cron/tabs` después de que el directorio se cree durante el inicio del sistema. Es posible que también deba añadir una entrada que cambie los modos y permisos en los directorios creados y en los archivos copiados con `/etc/rc.initdiskless`.

5.2. Syslog

`syslog.conf` especifica las ubicaciones de ciertos ficheros de log que hay en `/var/log`. Estos archivos no son creados por `/etc/rc.d/var` durante la inicialización del sistema. Por lo tanto, en algún lugar de `/etc/rc.d/var`, justo después de la sección que crea los directorios en `/var`, tendrá que añadir algo como esto:

```
# touch /var/log/security /var/log/maillog /var/log/cron /var/log/messages
# chmod 0644 /var/log/*
```

5.3. Instalación de ports

Antes de analizar los cambios necesarios para utilizar con éxito el árbol de ports, es necesario recordar que su sistema de archivos en el medio flash es de solo lectura. Dado que es de solo lectura, necesitará montarlo temporalmente en modo lectura-escritura utilizando la sintaxis que se muestra en la Sección 3, "El subsistema `rc` y los sistemas de archivos de solo lectura". Siempre debe volver a montar estos sistemas de archivos en modo solo lectura cuando haya terminado cualquier mantenimiento - las escrituras innecesarias en el medio flash podrían acortar considerablemente su vida útil.

Para que sea posible entrar en el directorio de ports y ejecutar con éxito el comando `make install`, debemos crear un directorio para los paquetes en un sistema de archivos que no se encuentre en la memoria para que mantenga nuestros paquetes durante los reinicios. Como es necesario montar sus sistemas de archivos en modo lectura-es-

critura para la instalación de un paquete, es apropiado suponer que también se puede usar un área en el medio flash para escribir la información del paquete.

Primero, cree el directorio para la base de datos de los paquetes. Normalmente se encuentra en `/var/db/pkg` , pero no podemos colocarlo allí ya que desaparecerá cada vez que se inicie el sistema.

```
# mkdir /etc/pkg
```

Ahora, agregue una línea al archivo `/etc/rc.d/var` que enlace `/etc/pkg` a `/var/db/pkg` . Un ejemplo:

```
# ln -s /etc/pkg /var/db/pkg
```

Ahora, cada vez que monte su sistema de archivos en modo lectura-escritura e instale un paquete, el comando `make install` funcionará, y la información del paquete se escribirá correctamente en `/etc/pkg` (porque el sistema de archivos, en ese momento, estará montado en modo lectura-escritura) que siempre estará disponible para el sistema operativo como `/var/db/pkg` .

5.4. Servidor Web Apache

Nota

Los pasos de esta sección solo son necesarios si Apache está configurado para escribir su pid o registro log fuera de `/var`. Por defecto, Apache guarda su archivo pid en `/var/run/httpd.pid` y sus registros de log en `/var/log` .

Se supone que Apache guarda sus archivos de logs en un directorio *`apache_log_dir`* fuera de `/var`. Cuando este directorio reside en un sistema de archivos de solo lectura, Apache no puede guardar ningún archivo de log y puede tener problemas para funcionar. Si es así, debe agregar un nuevo directorio al listado de directorios en `/etc/rc.d/var` a crear en `/var` y vincular *`apache_log_dir`* a `/var/log/apache` . También es necesario establecer permisos y propietarios a este nuevo directorio.

En primer lugar, agregue el directorio `log/apache` a la lista de directorios que se crearán en `/etc/rc.d/var` .

En segundo lugar, agregue estos comandos a `/etc/rc.d/var` después de la sección de creación del directorio:

```
# chmod 0774 /var/log/apache
# chown nobody:nobody /var/log/apache
```

Por último, elimine el directorio *`apache_log_dir`* y reemplácelo por un enlace:

```
# rm -rf apache_log_dir
# ln -s /var/log/apache  apache_log_dir
```