FreeBSD und Solid State-Geräte

Zusammenfassung

Dieser Artikel behandelt die Verwendung von Solid State Festplatten in FreeBSD, um eingebettete Systeme zu erstellen.

Eingebettete Systeme haben den Vorteil, dass sie eine höhere Stabilität aufgrund des fehlens von beweglichen Bauteilen (Festplatten) besitzen. Es muss jedoch beachtet werden, dass generell weniger Speicherplatz zur Verfügung steht und die Lebensdauer des Speichermediums geringer ist.

Spezielle Themen die behandelt werden beinhalten die Typen und Attribute von Solid State Datenträgern, die in FreeBSD verwendet werden, Kernel-Optionen die in solch einer Umgebung von Interesse sind, der rc.initdiskless-Mechanismus, welcher die Initalisierung solcher Systeme automatisiert, sowie die Notwendigkeit von Nur-Lese-Dateisystemen und das Erstellen von Dateisystemen von Grund auf. Der Artikel schliesst mit ein paar generellen Strategien für kleine und Nur-Lese-FreeBSD Umgebungen.

Übersetzt von Benedict Reuschling.

Inhaltsverzeichnis

1. Solid State Festplattengeräte 1
2. Kerneloptionen 2
3. Die `rc`-Subsysteme und nur-Lese Dateisysteme 2
4. Ein Dateisystem von Grund auf neu erstellen 3
5. Systemstrategien für kleine und Nur-Lese-Umgebungen 6

1. Solid State Festplattengeräte

Der Umfang des Artikels ist auf Solid State Geräte beschränkt, die auf Flash-Speicher basieren. Flash-Speicher ist ein Solid State Speicher (keine beweglichen Teile), der nicht flüchtig ist (der Speicher behält die Daten sogar nachdem alle Energiequellen ausgeschaltet wurden). Flash-Speicher kann grossen physischen Schock aushalten und ist vergleichsweise schnell (die Flash-Speicher Lösungen, die in diesem Artikel behandelt werden, sind nur wenig langsamer als eine EIDE-Festplatte in der Schreibgeschwindigkeit und viel schneller bei Lese-Operationen). Ein sehr wichtiger Aspekt von Flash-Speichern, dessen Auswirkungen später in diesem Artikel diskutiert werden, ist, dass jeder Sektor eine begrenzte Kapazität zur Wiederbeschreibung besitzt. Man kann nur eine bestimmte Anzahl von Schreib-, Lösch- und Wiederbeschreibungsvorgängen durchführen, bevor der Sektor permanent unbrauchbar wird. Obwohl viele Flash-Speicher Produkte automatisch schlechte Blöcke markieren und manche Geräte Schreiboperationen gleichmässig verteilen, bleibt weiterhin die Anzahl der durchführbaren Schreibvorgänge begrenzt. Verschiedene Geräteeinheiten besitzen zwischen 1,000,000 und 10,000,000 Schreibzyklen pro Sektor in ihren

Spezifikationen. Diese Zahlen variieren aufgrund der Umgebungstemperatur.

Im Speziellen werden wir ATA-kompatible Compact-Flash-Karten betrachten, welche als Speichermedien für Digitalkameras ziemlich populär sind. Von besonderem Interesse ist der Umstand, dass diese direkt an den IDE-Bus angeschlossen werden und kompatibel zu den ATA-Befehlen sind. Aus diesem Grund können diese Geräte mit einem einfachen und billigen Adapter direkt an den IDE-Bus eines Computers angeschlossen werden. Auf diese Weise sehen Betriebssysteme wie FreeBSD diese Geräte dann als normale (wenn auch sehr kleine) Festplatten an.

Andere Solid State Plattenlösungen existieren, jedoch platzieren deren Kosten, Obskurität und ihre Unhandlichkeit sie aussserhalb des Umfangs dieses Artikels.

2. Kerneloptionen

Ein paar Kerneloptionen sind von besonderem Interesse für diejenigen, welche diese eingebetteten FreeBSD-Systeme erstellen möchten.

Alle eingebetteten FreeBSD-Systeme, die Flash-Speicher als Systemplatte verwenden, sind interessant für Dateisysteme im Hauptspeicher und RAM-Disks. Wegen der begrenzten Anzahl von Schreibzyklen, die auf Flash-Speicher durchgeführt werden können, werden die Platte und die Dateisysteme darauf mit grosser Wahrscheinlichkeit nur lesend eingehängt werden. In dieser Umgebung werden Dateisysteme wie /tmp und /var als RAM-Disks eingebunden, um dem System zu erlauben, Logdateien anzulegen und Zähler sowie temporäre Dateien zu aktualisieren. RAM-Disks sind eine kritische Komponente für eine erfolgreiche Solid State Umsetzung in FreeBSD.

Sie sollten dafür sorgen, dass die folgenden Zeilen in Ihrer Kernelkonfigurationsdatei vorhanden sind:

```
options         MFS             # Memory Filesystem
options         MD_ROOT         # md device usable as a potential root device
pseudo-device   md              # memory disk
```

3. Die `rc`-Subsysteme und nur-Lese Dateisysteme

Die Initialisierung nach dem Bootvorgang eines eingebetteten FreeBSD-Systems wird von /etc/rc.initdiskless kontrolliert.

/etc/rc.d/var hängt /var als RAM-Disk ein, erstellt eine konfigurierbare Liste von Verzeichnissen in /var mittels des mkdir(1)-Kommandos und ändert die Attribute von ein paar dieser Verzeichnisse. Bei der Ausführung von /etc/rc.d/var kommt eine andere rc.conf-Variable ins Spiel: `varsize`. Die Datei /etc/rc.d/var erstellt eine /var-Partition basierend auf dem Wert dieser Variable in rc.conf:

```
varsize=8192
```

Standardmässig wird dieser Wert in Sektoren angegeben.

Der Fakt, dass es sich bei /var um ein nur-Lese Dateisystem handelt, ist eine wichtige Unterscheidung, da die /-Partition (und jede andere Partition, die Sie auf Ihrem Flash-Medium haben) nur lesend eingehängt wird. Erinnern Sie sich, dass in Solid State Festplattengeräte die Beschränkungen von Flash-Speichern erläutert wurden, speziell deren begrenzte Kapazität zum Schreiben. Die Notwendigkeit, Dateisysteme auf Flash-Speichern nur lesend einzubinden und keine Swap-Dateien zu verwenden, kann nicht oft genug erwähnt werden. Eine Swap-Datei auf einem ausgelasteten System kann in weniger als einem Jahr den gesamten Flash-Speicher aufbrauchen. Häufige Protokollierung oder das Erstellen und Löschen von temporären Dateien kann das gleiche verursachen. Aus diesem Grund sollten Sie zusätzlich zum entfernen des `swap`-Eintrags aus ihrer /etc/fstab-Datei auch noch die Optionsfelder für jedes Dateisystem auf `ro` wie folgt stellen:

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

Ein paar Anwendungen im normalen System werden sofort nach dieser Änderung ausfallen. Beispielsweise wird Cron nicht richtig funktionieren, aufgrund von fehlenden Cron-Tabellen in /var die von /etc/rc.d/var erstellt wurden. Syslog und DHCP werden ebenfalls Probleme durch das nur-Lese Dateisystem und fehlende Elemente im Verzeichnis /var verursachen, die /etc/rc.d/var erstellt hat. Diese Probleme sind jedoch nur vorübergehend und werden zusammen mit Lösungen zur Ausführung von anderen gebräuchlichen Softwarepaketen in Systemstrategien für kleine und Nur-Lese-Umgebungen angesprochen.

Eine wichtige Sache, an die man sich erinnern sollte, ist, dass ein Dateisystem, welches als nur lesend in /etc/fstab eingebunden wurde, jederzeit als schreibend durch das folgende Kommando eingehängt werden kann:

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

und auch wieder zurück auf nur lesend durch den Befehl:

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

4. Ein Dateisystem von Grund auf neu erstellen

Wenn ATA-kompatible Compact-Flash-Karten von FreeBSD als normale IDE-Festplatten erkannt werden, könnten Sie theoretisch FreeBSD aus dem Netzwerk mittels der Kern- und mfsroot-Floppies oder einer CD installieren.

Jedoch kann selbst eine kleine Installation von FreeBSD durch die normale Installationsprozedur ein System erzeugen, dass grösser als 200 MB ist. Da die meisten Leute kleinere Flash-Speichermedien einsetzen (128 MB wird hier als gross angesehen - 32 oder sogar 16 MB sind

gebräuchlich) ist eine gewöhnliche Installation mit normalen Methoden nicht möglich, da es einfach nicht genug freien Plattenplatz gibt, selbst für die kleinste Installationsart.

Der einfachste Weg, diese Speicherlimitierung zu umgehen, ist, FreeBSD auf konventionelle Weise auf eine normale Festplatte zu installieren. Nachdem die Installation abgeschlossen wurde, kürzen Sie das Betriebssystem auf das nötigste, bis Sie eine Grösse erreicht hat, die auf das Flash-Medium passt und benutzen Sie dann tar auf dem gesamten Dateisystem. Die folgenden Schritte werden Sie durch den Prozess der Vorbereitung eines Flash-Mediums für ihr getartes Dateisystem führen. Beachten Sie, dass Operationen wie Partitionierung, Benennung, Erstellung von Dateisystemen, etc. von Hand durchgeführt werden müssen, da eine normale Installation nicht möglich ist. Zusätzlich zu den Kern- und mfsroot-Disketten benötigen Sie auch die fixit-Floppy.

1. Partitionierung Ihrer Flash-Medien

   Wählen Sie nach dem Starten der Kern- und mfsroot-Disketten, `custom` aus dem Installationsmenü. In diesem Menü wählen Sie dann `partition` aus. Dort sollten Sie alle bestehenden Partitionen mit Hilfe der Taste `d` löschen. Nachdem alle bestehenden Partitionen gelöscht wurden, erstellen Sie mittels der Taste `c` eine Partition und akzeptieren Sie den Standardwert für die Grösse der Partition. Wenn Sie nach dem Typ der Partition gefragt werden, stellen Sie sicher, dass der Wert auf `165` eingestellt ist. Schreiben Sie jetzt diese Partitionstabelle auf die Platte durch betätigen der Taste `w` (dies ist die versteckte Option auf diesem Bildschirm). Wenn Sie eine ATA-kompatible Compact Flash-Karte verwenden, sollten Sie den FreeBSD Bootmanager auswählen. Drücken Sie nun die Taste `q`, um das Partitionsmenü zu verlassen. Sie werden das Menü des Bootmanagers noch ein weiteres Mal gezeigt bekommt. In diesem Fall wiederholen Sie die Auswahl von vorher.

2. Anlegen von Dateisystemen auf Ihrem Flashspeicher-Gerät

   Verlassen Sie das Installationsmenü und wählen Sie aus dem Hauptinstallationsmenü die Option `fixit`. In der fixit-Umgebung angelangt, geben Sie den folgenden Befehl ein:

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

   Zu diesem Zeitpunkt sollten Sie sich im vi-Editor unter der Herrschaft des disklabel-Kommandos befinden. Als nächstes müssen Sie die eine `a:`-Zeile an das Ende der Datei hinzufügen. Diese `a:`-Zeile sollte wie folgt aussehen:

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

   Wobei *123456* eine Zahl darstellt, die exakt der gleichen Zahl in der bestehenden Zeile mit dem `c:`-Eintrag entspricht. Sie kopieren quasi die bestehende Zeile `c:` als eine neue Zeile `a:` und stellen sicher, dass fstype `4.2BSD` entspricht. Speichern Sie die Datei und verlassen Sie den Editor.

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

```
# newfs /dev/ad0a
```

3. Schreiben des Dateisystems auf Ihr Flash-Medium

   Hängen Sie das neu erstellte Flash-Medium ein:

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

   Verbinden Sie diese Maschine mit dem Netzwerk, um die tar-Datei zu übertragen und extrahieren Sie es auf das Dateisystem des Flash-Mediums. Ein Beispiel dazu wäre folgendes:

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

   Jetzt da die Maschine ans Netzwerk angeschlossen ist, kopieren Sie die tar-Datei. An diesem Punkt werden Sie möglicherweise mit einem Dilemma konfrontiert - sollte Ihr Flash-Speicher beispielsweise 128 MB gross sein und Ihre tar-Datei grösser als 64 MB, können Sie ihre tar-Datei auf dem Flash-Speicher nicht entpacken - Ihnen wird vorher der Speicherplatz ausgehen. Eine Lösung für dieses Problem, sofern Sie FTP verwenden, ist, dass Sie die Datei entpacken können, während es von FTP übertragen wird. Wenn Sie die Übertragung auf diese Weise durchführen, haben Sie niemals die tar-Datei und deren Inhalt zur gleichen Zeit auf Ihrem Medium:

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

   Sollte Ihre tar-Datei gezippt sein, können Sie dies ebenso bewerkstelligen:

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

   Nachdem der Inhalt Ihrer tar-Datei auf dem Dateisystem des Flash-Mediums abgelegt wurden, können Sie den Flash-Speicher aushängen und neu starten:

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

   In der Annahme, dass Sie Ihr Dateisystem richtig konfiguriert haben, als es noch auf der gewöhnlichen Festplatte gebaut wurde (mit Ihren Nur-Lese-Dateisystemen und den nötigen Optionen im Kernel), sollten Sie nun erfolgreich von Ihrem FreeBSD Embedded-System starten können.

5. Systemstrategien für kleine und Nur-Lese-Umgebungen

In Die rc-Subsysteme und nur-Lese Dateisysteme wurde darauf hingewiesen, dass das /var -Dateisystem von /etc/rc.d/var konstruiert wurde und die Präsenz eines Nur-Lese-Wurzeldateisystems Probleme mit vielen in FreeBSD gebrächlichen Softwarepaketen verursacht. In diesem Artikel werden Vorschläge für das erfolgreiche Betreiben von cron, syslog, Installationen von Ports und dem Apache-Webserver unterbreitet.

5.1. cron

Während des Bootvorgangs wird /var von /etc/rc.d/var anhand der Liste aus /etc/mtree/BSD.var.dist gefüllt, damit cron, cron/tabs, at und ein paar weitere Standardverzeichnisse erstellt werden.

Jedoch löst das noch nicht das Problem, Crontabs über Neustarts des Systems hinaus zu erhalten. Wenn das System neu gestartet wird, verschwindet das /var-Dateiystem, welches sich im Hauptspeicher befunden hat und jegliche Crontabs, die Sie hatten werden ebenfalls verschwinden. Aus diesem Grund besteht eine Lösung darin, Crontabs für diejenigen Benutzer zu erstellen, die diese auch benötigen. Dazu sollte das /-Dateisystem lesend und schreibend eingehängt und diese Crontabs an einen sicheren Ort kopiert werden, wie beispielsweise /etc/tabs. Fügen Sie dann eine Zeile an das Ende der Datei /etc/rc.initdiskless hinzu, die diese Crontabs in /var/cron/tabs kopiert, nachdem dieses Verzeichnis während der Systeminitialisierung erstellt wurde. Sie werden auch eine Zeile hinzufügen mössen, welche die Besitzer und Berechtigungen auf diesen Verzeichnissen, die Sie erstellen und den dazugehörigen Dateien, die Sie mittels /etc/rc.initdiskless kopieren, setzen.

5.2. syslog

Die Datei syslog.conf spezifiziert den Ort von bestimmten Logdateien, welche in /var/log existieren. Diese Dateien werden nicht von /etc/rc.d/var während der Systeminitialisierung erstellt. Aus diesem Grund müssen Sie irgendwo in /etc/rc.d/var nach dem Abschnitt, der die Verzeichnisse in /var erstellt, eine Zeile ähnlich der folgenden hinzufügen:

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

5.3. Installation von Ports

Bevor die notwendigen Änderungen erkärt werden, einen Ports-Baum zu verwenden, ist es notwendig, Sie an die Nur-Lese-Besonderheit Ihres Dateisystems auf dem Flash-Medium zu erinnern. Da dieses nur lesend verfügbar ist, müssen Sie es vorübergehend mit Schreibrechten ausstatten, indem Sie die mount-Syntax, wie in Die rc-Subsysteme und nur-Lese Dateisysteme dargestellt wird, verwenden. Sie sollten immer diese Dateisysteme erneut mit nur-Lese-Rechten einhängen wenn Sie damit fertig sind - unnötige Schreibvorgänge auf dem Flash-Medium kann dessen Lebenszeit erheblich verkürzen.

Um es zu ermöglichen, in das Ports-Verzeichnis zu wechseln und erfolgreich make `install` auszuführen, müssen wir ein Paketverzeichnis auf einem Nicht-Hauptspeicherdateisystem erstellen, welches die Pakete über Neustarts hinweg im Auge behält. Weil es sowieso nötig ist, Ihre Dateisysteme mit Lese-Schreibrechten für die Installation eines Pakets einzuhängen, ist es sinnvoll anzunehmen, dass ein Bereich Ihres Flash-Mediums ebenfalls für Paketinformationen, die darauf abgespeichert werden, verwendet wird.

Erstellen Sie zuerst ein Verzeichnis für die Paketdatenbank. Normalerweise ist dies /var/db/pkg, jedoch können wir es dort nicht unterbringen, da es jedesmal verschwinden wird, wenn das System neu gestartet wird.

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

Fügen Sie nun eine Zeile in /etc/rc.d/var hinzu, welche das /etc/pkg-Verzeichnis mit /var/db/pkg verknüpft. Ein Beispiel:

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

Nun wird jedes Mal, wenn Sie Ihre Dateisysteme mit Lese-Schreibrechten einbinden und ein Paket installieren, der Befehl make `install` funktionieren und Paketinformationen werden erfolgreich nach /etc/pkg geschrieben (da zu diesem Zeitpunkt das Dateisystem mit Lese-Schreibrechten eingebunden ist), welche dann stets dem Betriebssystem als /var/db/pkg zur Verfügung stehen.

5.4. Apache Webserver

Die Anweisungen in diesem Abschnitt sind nur notwendig, wenn Apache so eingerichtet ist, dass dieser seine PID oder Protokollierungsinformationen ausserhalb von /var ablegt. Standardmässig bewahrt Apache seine PID-Datei in /var/run/httpd.pid und seine Protokolldateien in /var/log auf.

Es wird nun davon ausgegangen, dass Apache seine Protokolldateien in einem Verzeichnis namens apache_log_dir ausserhalb von /var speichert. Wenn dieses Verzeichnis auf einem nur-Lese-Dateisystem existiert, wird Apache nicht in der Lage sein, Protokolldateien zu speichern und wird vermutlich nicht richtig funktionieren. Wenn dies der Fall ist, muss ein neues Verzeichnis zu der Liste der Verzeichnisse in /etc/rc.d/var hinzugefügt werden, um dieses in /var zu erstellen und um apache_log_dir nach /var/log/apache zu verknüpfen. Es ist auch nötig, Berechtigungen und Besitzer auf diesem neuen Verzeichnis zu setzen.

Fügen Sie zuerst das Verzeichnis `log/apache` zu der Liste von Verzeichnissen hinzu, die in /etc/rc.d/var angelegt werden sollen.

Danach tragen Sie die folgenden Befehle in /etc/rc.d/var nach dem Abschnitt zum Erstellen der Verzeichnisse ein:

```
# chmod 0774 /var/log/apache
```

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

Schliesslich löschen Sie das bestehende apache_log_dir Verzeichnis und ersetzen es mit einer Verknüpfung:

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