

FreeBSD e Dispositivos de Estado Sólido

Resumo

Este artigo aborda o uso de dispositivos de disco de estado sólido no FreeBSD para criar sistemas embarcados.

Os sistemas embarcados têm a vantagem de uma maior estabilidade devido à ausência de partes móveis (discos rígidos). No entanto, é preciso ter em conta que o espaço disponível em disco é geralmente baixo no sistema e a durabilidade do meio de armazenamento.

Tópicos específicos a serem abordados incluem os tipos e atributos das mídias de estado sólido adequadas para uso como disco no FreeBSD, opções de kernel que são de interesse em tal ambiente, os mecanismos `rc.initdiskless` que automatizam a inicialização de tais sistemas e a necessidade de sistemas de arquivos read-only e a construção de sistemas de arquivos a partir do zero. O artigo será concluído com algumas estratégias gerais para ambientes FreeBSD pequenos e read-only.

Índice

1. Dispositivos de Disco de Estado Sólido	1
2. Opções do Kernel	2
3. O Subsistema <code>rc</code> e os Sistemas de Arquivos Read-Only	2
4. Construindo um sistema de arquivos a partir do zero	3
5. Estratégias do Sistema para Ambientes Pequenos e Somente Leitura	5

1. Dispositivos de Disco de Estado Sólido

O escopo deste artigo será limitado a dispositivos de disco de estado sólido feitos de memória flash. A memória flash é uma memória de estado sólido (sem partes móveis) que é não volátil (a memória mantém os dados mesmo depois de todas as fontes de energia terem sido desconectadas). A memória flash pode suportar um enorme choque físico e é razoavelmente rápida (as soluções de memória flash abordadas neste artigo são um pouco mais lentas que um disco rígido EIDE para operações de gravação e muito mais rápidas para operações de leitura). Um aspecto muito importante da memória flash, cujas ramificações serão discutidas mais adiante neste artigo, é que cada setor tem uma capacidade limitada de reescrita. Você só pode gravar, apagar e gravar novamente em um setor de memória flash um certo número de vezes antes que o setor fique permanentemente inutilizável. Embora muitos produtos de memória flash mapeiam automaticamente os blocos defeituosos, e embora alguns até distribuam operações de gravação uniformemente por toda a unidade, a verdade é que existe um limite para a quantidade de escrita que pode ser feita no dispositivo. Unidades competitivas possuem entre 1.000.000 e 10.000.000 gravações por setor em suas especificações. Este valor varia com a temperatura do ambiente.

Especificamente, estaremos discutindo unidades compact-flash compatíveis com ATA, as quais são bastante populares como mídia de armazenamento para câmeras digitais. De particular interesse é o fato de que eles são fixados diretamente no barramento IDE e são compatíveis com o conjunto de comandos ATA. Portanto, com um adaptador muito simples e de baixo custo, esses dispositivos podem ser conectados diretamente a um barramento IDE em um computador. Uma vez implementado desta maneira, sistemas operacionais como o FreeBSD vêem o dispositivo como um disco rígido normal (embora pequeno).

Outras soluções de disco de estado sólido existem, mas seu custo, obscuridade e relativa dificuldade de uso os colocam além do escopo deste artigo.

2. Opções do Kernel

Algumas opções do kernel são de interesse específico para aqueles que criam sistemas FreeBSD embarcados.

Todos os sistemas embarcados FreeBSD que usam memória flash como disco para o sistema estarão interessados em usar discos em memória e sistemas de arquivos em memória. Devido ao número limitado de gravações que podem ser feitas na memória flash, o disco e os sistemas de arquivos no disco provavelmente serão montados como read-only. Nesse ambiente, sistemas de arquivos tais como /tmp e /var são montados como sistemas de arquivos em memória para permitir que o sistema crie logs e atualize contadores e arquivos temporários. Os sistemas de arquivos em memória são um componente crítico para uma implementação bem-sucedida do FreeBSD em dispositivos de estado sólido.

Você deve ter certeza de que as seguintes linhas existem no seu arquivo de configuração do kernel:

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

3. O Subsistema rc e os Sistemas de Arquivos Read-Only

A inicialização pós-boot de um sistema FreeBSD embarcado é controlada por /etc/rc.initdiskless.

O /etc/rc.d/var monta o /var como um sistema de arquivos em memória, cria uma lista configurável de diretórios em /var com o comando [mkdir\(1\)](#) e altera os modos em alguns desses diretórios. Na execução do /etc/rc.d/var, uma outra variável rc.conf entra em jogo – **varsize**. Uma partição /var é criada por /etc/rc.d/var baseado no valor desta variável em rc.conf:

```
varsize=8192
```

Lembre-se de que esse valor é informado em setores, por padrão.

O fato do /var ser um sistema de arquivos read-write é uma distinção importante, pois a partição / (e quaisquer outras partições que você possa ter em sua mídia flash) deve ser montada como read-only. Lembre-se que em [Dispositivos de Disco de Estado Sólido](#) detalhamos as limitações da memória flash - especificamente a capacidade de gravação limitada. A importância de não montar sistemas de arquivos em mídia flash em modo read-write, e a importância de não usar um arquivo de swap, não pode ser exagerado. Um arquivo de swap em um sistema ocupado pode inutilizar uma mídia flash em menos de um ano. Criação de log pesado ou criação e destruição de arquivos temporários podem fazer o mesmo. Portanto, além de remover a entrada **swap** do seu /etc/fstab, você também deve alterar o campo Options para cada sistema de arquivos para **ro** como segue:

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

Alguns aplicativos no sistema começarão a falhar imediatamente como resultado desta alteração. Por exemplo, o cron não será executado corretamente como resultado da falta de crontabs no /var criado pelo /etc/rc.d/var, o syslog e o dhcp também irão encontrar problemas como resultado do sistema de arquivos estar em modo read-only e dos itens ausentes no /var que o /etc/rc.d/var criou. Estes são apenas problemas temporários, embora sejam abordados, juntamente com soluções para a execução de outros pacotes de software comuns em [Estratégias do Sistema para Ambientes Pequenos e Somente Leitura](#).

Uma coisa importante para lembrar é que um sistema de arquivos que foi montado como read-only com o /etc/fstab pode ser colocado em modo read-write a qualquer momento, executando o comando:

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

e pode ser alternado de volta para somente leitura com o comando:

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

4. Construindo um sistema de arquivos a partir do zero

Como os cartões Compact Flash compatíveis com ATA são vistos pelo FreeBSD como discos rígidos IDE normais, você poderia teoricamente instalar o FreeBSD a partir da rede usando o os disquetes do kern e mfsroot ou de um CD.

No entanto, mesmo uma pequena instalação do FreeBSD utilizando procedimentos normais de instalação pode produzir um sistema com tamanho maior que 200 megabytes. Como a maioria das pessoas usará dispositivos de memória flash menores (128 megabytes são considerados razoavelmente grandes - 32 ou até mesmo 16 megabytes são comuns), uma instalação usando mecanismos normais não será possível - simplesmente não há espaço em disco suficiente nem para as menores instalações convencionais.

A maneira mais fácil de superar essa limitação de espaço é instalar o FreeBSD usando meios convencionais em um disco rígido normal. Após a conclusão da instalação, reduza o sistema operacional para um tamanho que caiba na mídia flash e compacte o sistema de arquivos inteiro com o tar. Os passos seguintes irão guiá-lo através do processo de preparação de uma parte da memória flash para o seu sistema de arquivos compactado com o tar. Lembre-se de que não estamos executando uma instalação normal, logo as operações como particionamento, criação dos labels, criação do sistema de arquivos, etc. precisam ser executadas manualmente. Além dos disquetes do kern e mfsroot, você também precisará usar o disquete do fixit.

1. Particionando seu Dispositivo de Mídia Flash

Após inicializar com os disquetes do kern e mfsroot, escolha **custom** no menu de instalação. No menu de instalação personalizada, escolha **partition**. No menu de partições, você deve apagar todas as partições existentes usando a tecla **d**. Depois de excluir todas as partições existentes, crie uma partição usando a tecla **c** e aceite o valor padrão para o tamanho da partição. Quando perguntado sobre o tipo da partição, certifique-se de que o valor esteja configurado para **165**. Agora escreva esta tabela de partições no disco pressionando **w** (esta é uma opção oculta nesta tela). Se você estiver usando um cartão compact flash compatível com ATA, deverá escolher o FreeBSD Boot Manager. Agora pressione **q** para sair do menu de partições. Você verá novamente o menu do gerenciador de inicialização - repita a escolha feita anteriormente.

2. Criando Sistemas de Arquivos em seu Dispositivo de Memória Flash

Saia do menu de instalação personalizada e, no menu de instalação principal, escolha a opção **fixit**. Depois de entrar no ambiente do fixit, digite o seguinte comando:

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

Neste ponto, você terá entrado no editor vi sob os auspícios do comando disklabel. Em seguida, você precisa adicionar uma linha **a:** no final do arquivo. Esta linha **a:** deve ser semelhante a linha abaixo:

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

Onde **123456** é um número o qual é exatamente o mesmo que o número existente na entrada **c:** para o tamanho. Basicamente, você está duplicando a linha **c:** existente como uma linha **a:**, certifique-se de que o fstype seja **4.2BSD**. Salve o arquivo e saia.

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

3. Colocando seu Sistema de Arquivos na Mídia Flash

Monte a mídia flash recém-preparada:

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

Coloque esta máquina na rede para que possamos transferir nosso arquivo tar e extrai-lo em nosso sistema de arquivos de mídia flash. Um exemplo de como fazer isso é:

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

Agora que a máquina está na rede, transfira seu arquivo tar. Você pode se deparar com um pequeno dilema neste ponto - se a sua memória flash tiver por exemplo 128 megabytes, e seu arquivo tar for maior que 64 megabytes, você não poderá ter o seu arquivo tar na mídia flash ao mesmo tempo em que realiza a descompressão - você ficará sem espaço. Uma solução para esse problema, se você estiver usando FTP, é descompactar o arquivo enquanto ele é transferido por FTP. Se você realizar sua transferência desta maneira, você nunca terá o arquivo tar e o conteúdo do tar em seu disco ao mesmo tempo:

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

Se o seu arquivo tar estiver gzipado, você pode fazer isso também:

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

Depois que o conteúdo do seu sistema de arquivos compactado pelo tar estiver no sistema de arquivos da sua memória flash, você poderá desmontar a memória flash e reinicializar:

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

Assumindo que você configurou seu sistema de arquivos corretamente quando ele foi construído no disco rígido normal (com seus sistemas de arquivos montado como read-only, e com as opções necessárias compiladas no kernel) você agora deve inicializar com sucesso seu sistema embarcado FreeBSD.

5. Estratégias do Sistema para Ambientes Pequenos e Somente Leitura

Em [O Subsistema rc e os Sistemas de Arquivos Read-Only](#), foi apontado que o sistema de arquivos /var construído pelo /etc/rc.d/var e a presença de um sistema de arquivos raiz read-only causa problemas com muitos pacotes de software comuns usados com o FreeBSD. Neste artigo, serão fornecidas sugestões para a execução bem-sucedida do cron, do syslog, instalações de ports e do

servidor Web Apache.

5.1. Cron

Na inicialização, o `/var` é preenchido pelo `/etc/rc.d/var` usando a lista disponível em `/etc/mtree/BSD.var.dist`, então o `cron`, o `cron/tabs`, `at`, e alguns outros diretórios padrões são criados.

No entanto, isso não resolve o problema de manter as crontabs entre nas reinicializações. Quando o sistema for reinicializado, o sistema de arquivos `/var` que está na memória desaparecerá e todas as crontabs que você tenha nele também desaparecerão. Portanto, uma solução seria criar crontabs para os usuários que precisam delas, montar seu sistema de arquivos `/` como `read-write` e copiar estas crontabs para algum lugar seguro, como `/etc/tabs`, em seguida, adicione uma linha ao final do `/etc/rc.initdiskless` que copie estes crontabs para `/var/cron/tabs` depois que o diretório for criado durante inicialização do sistema. Você também pode precisar adicionar uma linha que altere modos e permissões nos diretórios criados e nos arquivos copiados com `/etc/rc.initdiskless`.

5.2. Syslog

O `syslog.conf` especifica os locais de certos arquivos de log que existem em `/var/log`. Esses arquivos não são criados pelo `/etc/rc.d/var` na inicialização do sistema. Portanto, em algum lugar do `/etc/rc.d/var`, logo após a seção que cria os diretórios em `/var`, você precisará adicionar algo como isto:

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

5.3. Instalação de Ports

Antes de discutir as alterações necessárias para usar com êxito a árvore de ports, é necessário um lembrete sobre a natureza `read-only` dos seus sistemas de arquivos na mídia flash. Como eles são `read-only`, você precisará montá-los temporariamente para `read-write` usando a sintaxe de montagem mostrada em [O Subsistema `rc` e os Sistemas de Arquivos `Read-Only`](#). Você sempre deve remontar esses sistemas de arquivos no modo `read-only` quando tiver terminado qualquer manutenção - gravações desnecessárias na mídia flash podem reduzir consideravelmente sua vida útil.

Para tornar possível entrar em um diretório do ports e executar com sucesso o comando `make install`, devemos criar um diretório de pacotes em um sistema de arquivos que não esteja localizado na memória o qual manterá o controle dos nossos pacotes entre as reinicializações. Como é necessário montar seus sistemas de arquivos como `read-write` para a instalação de um pacote, é sensato supor que uma área na mídia flash também possa ser usada para que as informações do pacote sejam gravadas.

Primeiro, crie o diretório do banco de dados de pacotes. Ele fica normalmente em `/var/db/pkg`, mas não podemos colocá-lo lá, pois ele irá desaparecer toda vez que o sistema for inicializado.

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

Agora, adicione uma linha ao arquivo `/etc/rc.d/var` que vincule o `/etc/pkg` ao `/var/db/pkg`. Um exemplo:

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

Agora, sempre que montar seus sistemas de arquivos como read-write e instalar um pacote, o `make install` funcionará e as informações do pacote serão gravadas com êxito em `/etc/pkg` (porque o sistema de arquivos será, naquele momento, montado como read-write) que estará sempre disponível para o sistema operacional como `/var/db/pkg`.

5.4. Servidor Web Apache



As etapas nesta seção são necessárias apenas se o Apache estiver configurado para gravar suas informações de pid ou log fora do `/var`. Por padrão, o Apache mantém seu arquivo pid em `/var/run/httpd.pid` e seus arquivos de log em `/var/log`.

Agora é assumido que o Apache mantém seus arquivos de log em um diretório `apache_log_dir` fora do `/var`. Quando esse diretório reside em um sistema de arquivos read-only, o Apache não poderá salvar nenhum arquivo de log e pode ter problemas para funcionar. Se assim for, é necessário adicionar um novo diretório à lista de diretórios em `/etc/rc.d/var` para criar no `/var` e vincular `apache_log_dir` ao `/var/log/apache`. Também é necessário definir permissões e propriedade neste novo diretório.

Primeiro, adicione o diretório `log/apache` à lista de diretórios a serem criados em `/etc/rc.d/var`.

Segundo, adicione estes comandos ao `/etc/rc.d/var` após a seção de criação do diretório:

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

Por fim, remova o diretório `apache_log_dir` existente e substitua-o por um link:

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