

Atualizando seu FreeBSD (sistema e Ports) por código fonte.

Autoria de Christopher Giese
15/06/2006
Última Atualização 29/06/2006

Nesse artigo Christopher Giese e Reginaldo Russinholi, com complementos de Renato Botelho, instruem o leitor durante o processo de atualização do FreeBSD por código fonte, abordando atualização do sistema operacional, das definições da Coleção de Ports local e das próprias aplicações (ports) instaladas localmente, neste caso por código fonte ou através de pacotes pré-compilados.

O artigo aborda o uso do csup ao invés do tradicional cvsup, e do portupgrade, ferramentas mais modernas e rápidas, disponíveis nas últimas versões do FreeBSD e que começam se tornar padrão desde os últimos dois releases do FreeBSD. Leitura recomendada.

FreeBSD

ATUALIZANDO O CÓDIGO DE SEU FREEBSD 6.x VIA CSUP

Inicialmente iremos baixar os source do FreeBSD, neste howto faremos isto utilizando o csup.

Dependendo da sua versão atual do FreeBSD, o csup já pode fazer parte da base do sistema, não necessitando instalação, para checar isso basta verificar se o arquivo /usr/bin/csup

Para instalar o binario do csup caso ele não esteja ainda no sistema, temos aqui 2 opcoes....
escolha a sua preferida:

Por pacote pré-compilado:

```
pkg_add -r csup
rehash
```

Pela Coleção de Ports:

```
cd /usr/ports/net/csup
make install clean
rehash
```

Agora iremos criar um arquivo supfile que servira para o nosso csup como um "arquivo de configuracao". Como o nosso objetivo inicial e' atualizar o nosso FreeBSD para a ultima versao STABLE do FreeBSD 6.X (Quando escrevi este howto era 6.1 STABLE), voce usaria este supfile:

```
#----- INICIO DO ARQUIVO -----#
*default host=cvsup12.FreeBSD.org
*default base=/var/db
*default prefix=/usr
*default release=cvs tag=RELENG_6
*default delete use-rel-suffix

*default compress

src-all
#----- FIM DO ARQUIVO -----#
```

OBS: salve com o nome/local "/usr/local/etc/src-supfile" .

Vale lembrar que podemos escolher o servidor que desejamos "baixar" os sources, neste exemplo foi utilizado o cvsup12.FreeBSD.org que e' um Top Level Domain, ou seja, os primeiros servidores que são atualizados do cvs oficial. Para obter informacoes sobre outros mirrors consulte a pagina oficial do FreeBSD <http://www.freebsd.org> ou instale o fastest_cvsup (cd /usr/ports/sysutils/fastest_cvsup && make install clean), com ele voce conseguira ver qual o servidor mais rapido no momento.

Depois de tudo configurado basta digitar:

```
csup -g -L 2 /usr/local/etc/src-supfile
```

Com este comando sera iniciado o "download" do novo codigo do FreeBSD.

Depois que tudo acabar, voce tera que recompilar o seu FreeBSD, para fazer isso faca:

```
cd /usr/src
make -j4 buildworld (* antes de dar o enter.... leia o * ali para baixo)
make buildkernel KERNCONF=GENERIC
make installkernel KERNCONF=GENERIC
mergemaster -p
make TMPDIR=/var/tmp installworld
mergemaster -Pi
reboot
```

NOTA: sobre o mergemaster

O `mergemaster -p` irá comparar os usuários/grupos atuais da máquina com os necessários pelo sistema. Muitas vezes usuários novos são incorporados ao sistema, e são necessários até mesmo para que o installworld seja executado corretamente. Por isso a necessidade de executá-lo antes do mesmo. Tome muito cuidado pois se for feita a escolha errada você poderá perder todos os usuários/grupos que foram cadastrados após a instalação.

Na maioria dos casos, a opção ideal para ser usada no `mergemaster -p` é a opção "merge". Essa opção permite que você junte as novas alterações necessárias pelo sistema, mais as suas alterações, ou seja, os usuários/grupos que você adicionou. Antes de selecionar a opção "i" para que o seu merge seja instalado, utilize a opção "v" e dê uma conferida no novo arquivo gerado.

O `mergemaster -Pi` irá comparar o seu /etc com o /usr/src/etc, essa comparação não é feita por conteúdo, e sim pela versão do arquivo no CVS. Essa versão pode ser vista em cada arquivo em uma linha, no começo do mesmo, que possui a string "\$FreeBSD:", portanto, se tem um arquivo que você não quer atualizar e não quer que apareça na hora da comparação, como por exemplo, o /etc/pf.conf, basta copiar essa linha do arquivo mais novo (/usr/src/etc/pf.conf) substituindo a do atual (/etc/pf.conf), estando os dois com a mesma versão, serão considerados iguais.

O parâmetro -i faz com que arquivos novos que foram incorporados ao sistema sejam adicionados automaticamente, e o -P faz uma cópia de cada arquivo que você selecionar a opção "i" para instalar o mais novo e guarda isso em "/var/tmp/mergemaster", essa opção é extremamente importante, pois caso você substitua um arquivo que não deveria, você tem um backup.

Execute ***SEMPRE*** o mergemaster, e com muito cuidado, essa é uma parte fundamental da atualização, e, se não for feita corretamente, seu servidor pode não inicializar mais os aplicativos corretamente no próximo boot.

* Aqui a mais importantes de todas as observacoes:

Obs.: O "make -j4 buildworld" demora muito pode ir tomar uma cervejinha

- Se sua maquina for rapida = umas 4 cervejinhas

- Se sua maquina for nao tao rapida = 10 cervejas + bisteca na chapa

- ... agora se sua maquina for lerda.... logo apos o enter..... coloque a COSTELA no celofane ;)

Depois de tudo atualizado, é hora de fazer uma limpeza no sistema, pra remover programas, mans e libs que foram removidas do sistema, mas permanecem na sua máquina.

ATENÇÃO: Se você atualizou da série 5.x pra 6.x por exemplo, as libs do sistema tiveram sua versão alterada, então, antes de executar os procedimentos abaixo, atualiza ***TODOS*** os seus ports (portupgrade -fa) e recompila programas que foram instalado por fora do ports, para garantir que nada irá parar de funcionar.

```
# cd /usr/src
# make check-old
```

Faça um backup de todos os arquivo que o make check-old retornar

```
# make delete-old
# make delete-old-libs
```

Esses dois comandos irão perguntar pra você antes de apagar cada arquivo e/ou cada lib. Caso você tenha

conferido bem a saída do make check-old e tenha certeza de que vai apagar tudo, utilize-os da seguinte maneira:

```
# make -DBATCH_DELETE_OLD_FILES delete-old  
# make -DBATCH_DELETE_OLD_FILES delete-old-libs
```

PARTE 2 - ATUALIZANDO PORTS

Depois de atualizar seu FreeBSD para o último 6-STABLE, aí vão dicas para manutenção dos ports.

Para atualizar a árvore do ports:

Foi adicionada ao FreeBSD uma ferramenta chamada portsnap, ela serve para atualizar a árvore do ports, veja bem, isso não atualiza os ports, apenas a árvore de diretórios.

Na primeira vez é meio demorado, mas depois se torna mais rápido e consome menos recursos que o método mais conhecido, via cvsup, ou agora, csup.

Execute isso na primeira vez

```
# portsnap fetch extract
```

Nas próximas vezes isso basta

```
# portsnap fetch update
```

E se você for colocar no crontab

```
# portsnap cron update
```

Isso atualiza e deixa a sua árvore do ports com tudo zerado, agora, antes de atualizar qualquer port é necessário ler o arquivo /usr/ports/UPDATING. Quando um port necessita de algo diferente na sua atualização, isso fica anotado nesse arquivo, e deve ser lido com atenção, principalmente para servidores em produção, não queremos parar nenhum serviço desnecessariamente.

Antes de instalar qualquer pacote via ports, existe uma ferramenta que deve ser instalada para garantir a segurança dos produtos que vão ser instalados nesse server, é o portaudit. Ele faz consulta a base de Vulnerabilidades (VuXML) e vai te alertar quando você tentar instalar um port que tem uma falha, e, também, se você tem um port instalado e uma vulnerabilidade é encontrada nele, o portaudit também avisa você no email que é enviado diariamente para o root.

Para instalar o portaudit:

```
# cd /usr/ports/security/portaudit  
# make install clean  
# rehash
```

Para baixar a base de dados e checar:

```
# portaudit -Fa
```

O portaudit será incorporado ao /etc/periodic, e baixará diariamente a base atualizada e fará uma checagem.

Uma ferramenta extremamente útil para atualização de ports é o portupgrade, para instalá-lo

```
# cd /usr/ports/sysutils/portupgrade && make install clean
```

Depois de instalado, se você quiser saber quais ports necessitam de atualização:

```
# portversion -vL=
```

Se você quiser atualizar um pacote

```
# portupgrade pacote1 pacote2 pacote3
```

Se voce quiser atualizar tudo

```
# portupgrade -a
```

Atualizar tudo usando pacotes pre-compilados, se for possivel

```
# portupgrade -Pa
```

Usar apenas pacotes pre-compilados

```
# portupgrade -PPa
```

Infinitas opcoes existem no portupgrade, lendo o man voce tera uma nocao mais ampla de tudo o que ele pode fazer, mas o mais interessante e' que quando voce roda um portupgrade pacote e ele vai atualizar o pacote de 1.2 pra 1.3, antes dele instalar a nova versao 1.3, ele faz um pacote de backup do 1.2, se no momento da instalacao der problema, ele restaura o pacote da 1.2 e tudo fica como antes.

Uma outra dica interessante e' a respeito de atualizacao do perl, quando a mudanca for algo como perl-5.8.7 para perl-5.8.7_1, essa dica nao serve de nada, pois a versao do perl e' a mesma, o que mudou foi a versao do port, agora quando for 5.8.7 -> 5.8.8, ai sim anote essa, para nao precisar recompilar todos os ports que salvam libs dentro de /usr/local/lib/perl/5.8.7, para que elas movam tudo para o novo diretorio 5.8.8, foi escrito um script que faz as mudancas sozinho sem precisar recompilar nada, e' o perl-after-upgrade. Entao basta executar:

```
# portupgrade perl  
# perl-after-upgrade -f
```

Bom pessoal e' isso, nao tem segredo.

Espero ter ajudado :-)

```
# Sample By  
# Christopher Giese <skywarrior at bsdux.com.br>  
# Renato Botelho <garga at FreeBSD.org>  
# Thanks to: Reginaldo Russinholi  
# www.bsdux.com.br
```

Este artigo foi submetido à lista FUG-BR por Christopher Giese, e publicado originalmente aqui .