Construindo Produtos com o FreeBSD

Joseph Koshy, The FreeBSD Project <jkoshy@FreeBSD.org >

Revisão: 43184

FreeBSD is a registered trademark of the FreeBSD Foundation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the “™” or the “®” symbol.

2013-11-13 por hrs.

Sumário

O projeto FreeBSD é um projeto voluntário e colaborativo de âmbito mundial, o qual desenvolve um sistema operacional de alta qualidade, capaz de ser utilizado em diferentes arquiteturas computacionais. O projeto FreeBSD distribui o código fonte do seu produto sob uma licença liberal, com a intenção de incentivar o uso de seu código. Colaborar com o projeto FreeBSD pode ajudar sua empresa a reduzir o tempo necessário para colocar um produto no mercado, a reduzir seus custos de engenharia e a melhorar qualidade de seus produtos.

Este artigo analisa as questões envolvidas no uso do código do FreeBSD em appliances e softwares. Ele também destaca as características do FreeBSD, que o tornam uma excelente base para o desenvolvimento de produtos. O artigo conclui sugerindo um conjunto das “melhores práticas” de organizações que colaboram com o projeto FreeBSD.

Índice

1. Introdução 2
2. O FreeBSD como um conjunto de blocos de construção 3
3. Colaborando com o FreeBSD 8
4. Conclusão 14
Bibliografia 14

1. Introdução

Atualmente o FreeBSD é bem conhecido como um sistema operacional de alto desempenho para servidores. Ele está instalado em milhões de servidores web e em outros hosts conectados diretamente a internet em todo o mundo. O código do FreeBSD também é parte integrante de muitos produtos, que vão desde aparelhos como roteadores de rede, firewalls e dispositivos de armazenamento, até computadores pessoais. Partes do FreeBSD também têm sido utilizadas em softwares comerciais (consulte Seção 2, “O FreeBSD como um conjunto de blocos de construção”).

Neste artigo, vamos olhar para o Projeto FreeBSD como um recurso de engenharia de software — como um conjunto de blocos de construção e de processos os quais você pode utilizar para construir produtos.

Embora o código fonte do FreeBSD seja distribuído gratuitamente ao público, para desfrutar plenamente dos benefícios do trabalho do projeto, as organizações precisam *colaborar* com o mesmo. Nas seções subsequentes do presente artigo discutiremos formas eficazes de colaborar com o projeto, bem como os perigos que precisam ser evitados ao fazê-lo.

Advertência ao Leitor. O autor considera que as características do projeto FreeBSD mencionadas neste artigo eram substancialmente verdadeiras no momento em que o artigo foi concebido e escrito (2005). No entanto, o leitor deve ter em mente que as práticas e processos utilizados por comunidades de código aberto podem mudar ao longo do tempo, e que portanto as informações deste artigo devem ser consideradas apenas como indicativas e não como verdades absolutas.

1.1. Público Alvo

Este documento tem como público alvo os seguintes grupos de pessoas:

- Tomadores de decisão em empresas que estejam em busca de meios para melhorar a qualidade de seus produtos, de reduzir o tempo necessário para lançá-los no mercado e de reduzir seus custos de engenharia no longo prazo.
- Consultores de tecnologia procurando as melhores práticas para alavancar projetos de “código aberto”.
- Observadores da indústria interessados em compreender a dinâmica dos projetos de “código aberto”.
- Desenvolvedores de software que utilizam o FreeBSD e que buscam formas de contribuir com o projeto.

1.2. Objetivos do artigo

Após a leitura deste artigo, você deve ter:

- Uma melhor compreensão dos objetivos do Projeto FreeBSD e de sua estrutura organizacional.
- Uma visão geral das tecnologias disponíveis no projeto.
- Uma melhor compreensão do modelo de desenvolvimento adotado pelo Projeto FreeBSD e dos processos de engenharia envolvidos no lançamento de uma nova versão do sistema.
- Uma compreensão de como os processos convencionais de desenvolvimento de software em uma empresa diferem dos processos utilizados no projeto FreeBSD.
- Consciência dos canais de comunicação utilizados pelo projeto e do nível de transparência que você pode esperar.
- Consciência das melhores formas de se trabalhar com o projeto — a melhor forma de reduzir os custos de engenharia, de reduzir o tempo necessário para levar seu produto ao mercado, de gerir vulnerabilidades de segurança, e de preservar a compatibilidade futura com o seu produto a medida que o Projeto FreeBSD evolui.

1.3. Estrutura do Artigo

O restante deste artigo está estruturado da seguinte forma:

- A Seção 2, “O FreeBSD como um conjunto de blocos de construção” apresenta o projeto FreeBSD, explora sua estrutura organizacional, as principais tecnologias e processos de engenharia envolvidos no lançamento de uma nova versão do sistema.
- A Seção 3, “Colaborando com o FreeBSD” descreve formas de colaborar com o Projeto FreeBSD. Esta seção também aborda as armadilhas que são geralmente encontradas por empresas que trabalham com projetos voluntários como o FreeBSD.
- A Seção 4, “Conclusão” conclui o artigo.

2. O FreeBSD como um conjunto de blocos de construção

O FreeBSD fornece uma excelente base sobre a qual podemos construir produtos:

- O código fonte do FreeBSD é distribuído sob uma licença BSD liberal, o que facilita sua adoção em produtos comerciais com um mínimo de preocupações. [9]
- O Projeto FreeBSD possui excelentes práticas de engenharia as quais podem ser aproveitadas.
- O projeto oferece uma transparência excepcional em seu funcionamento, permitindo que as empresas que utilizam o seu código se planejem de forma eficaz para o futuro.

- A cultura do projeto FreeBSD, herdada do Grupo de Pesquisa de Ciências da Computação da Universidade da Califórnia em Berkeley [8], fomenta trabalhos de alta qualidade. Algumas funcionalidades do FreeBSD definem o estado da arte.

O [5] analisa em maior profundidade os motivos comerciais para se utilizar código fonte aberto. Para as organizações, os benefícios do uso de componentes do FreeBSD em seus produtos incluem a redução do tempo necessário para lançar novos produtos no mercado, menores custos e menores riscos de desenvolvimento.

2.1. Construindo com o FreeBSD

Aqui estão alguns exemplos de como as empresas estão utilizando o FreeBSD:

- Como um provedor (*upstream source*) de códigos testados para bibliotecas e utilitários.

  Sendo o *downstream* do projeto, as organizações se aproveitam das novas funcionalidades, das correções de bugs e dos testes que o código fonte do projeto FreeBSD recebe.

- Como sistema operacional integrado (por exemplo, em um roteador OEM e ou em um dispositivo de firewall). Neste modelo, as empresas utilizam uma versão customizada do kernel e do conjunto de aplicativos do FreeBSD, juntamente com uma camada proprietária de gestão para os seus dispositivos. Os fabricantes de equipamentos originais (OEMs) se beneficiam da adição por parte do FreeBSD de suporte a novos componentes de hardware, bem como se beneficia dos testes que o sistema base recebe.

  O FreeBSD é distribuído com um ambiente de desenvolvimento auto-hospedado o qual permite a fácil criação de tais configurações.

- Como um ambiente Unix compatível para as funções de gerenciamento em dispositivos de armazenamento high-end e em dispositivos de rede, executando em uma lâmina separada.

  O FreeBSD fornece ferramentas para a criação de imagens do sistema operacional dedicadas a executar uma função específica. Sua implementação da API unix BSD é madura e testada. O FreeBSD também pode proporcionar um ambiente de desenvolvimento cruzado estável para os outros componentes de dispositivos topo de linha.

- Como um veículo para obter suporte e testes amplos de uma equipe mundial de desenvolvedores para a sua "propriedade intelectual" não-crítica.

  Neste modelo, as organizações contribuem com frameworks de infra-estrutura úteis ao projeto FreeBSD (por exemplo, veja o netgraph(3)). A ampla exposição que o código obtém ajuda na rápida identificação de bugs e de problemas de desempenho. O envolvimento de desenvolvedores de alta qualidade também resulta

no desenvolvimento de extensões úteis para a infra-estrutura do sistema, e das quais a empresa que está contribuindo com o projeto também se beneficia.

- Como um ambiente de desenvolvimento apoiando desenvolvimento cruzado para sistemas operacionais embarcados como RTEMS e o eCOS.

  Existem muitos ambientes de desenvolvimento completos na forte coleção de mais de 24,000 aplicativos portados e empacotados para o FreeBSD.

- Como forma de suportar uma API estilo Unix em um sistema operacional que de outro modo seria proprietário, aumentando a sua palatabilidade para os desenvolvedores de aplicativos.

  Aqui as partes do kernel do FreeBSD e as aplicações são "portadas" para serem executadas juntamente com outras tarefas no sistema operacional proprietário. A disponibilidade de uma implementação estável e bem testada da API Unix™ pode reduzir o esforço necessário para portar aplicações populares para um sistema operacional proprietário. Como o FreeBSD é distribuído acompanhado de uma documentação de alta qualidade sobre a sua estrutura interna, e possui processos eficazes de engenharia para gerenciamento de vulnerabilidades e para lançamento de novas versões, os custos para mantê-lo atualizado são baixos.

2.2. Tecnologias

Existe um grande número de tecnologias suportadas pelo projeto FreeBSD. Abaixo você encontra uma lista com alguma delas:

- Um sistema completo que pode compilar a si mesmo de forma cruzada para as seguintes arquiteturas: alpha (até o FreeBSD versão 6.X), amd64, ia64, i386, sparc64, powerpc (veja build(7)).

- Suporte para as seguintes tecnologias, protocolos e padrões: ATA, ATAPI, ATM, Bluetooth™, CAM, CardBus™, DHCP, DNS, EISA™, Ethernet™, FDDI, Fibre Channel, GPIB, IEEE 1394, IPv4, IPv6, IPSEC, IPX™, ISDN, MAC, NIS, NFS, OpenSSH, OPIE, PAM, PCI™, PCMCIA, POSIX™, PnP, RAID, RPC, SATA, SCSI, SMB, TCP, USB, VESA, VLAN, VLB, WebNFS™.

- Um kernel modular capaz de multiprocessamento simétrico, com módulos de kernel carregáveis e um sistema de configuração flexível e fácil de usar.

- Suporta a emulação de binários do Linux e do SVR4 com velocidades próximas as que você obtém executando os aplicativos de forma nativa. Suporte para os binários dos drivers de rede do Windows™ (NDIS).

- Bibliotecas para muitas tarefas de programação: arquivos, suporte a FTP e HTTP, suporte a threads, além de um ambiente completo de programação POSIX™ like.

- Funcionalidades avançadas de segurança: Controle de Acesso Obrigatório (mac(9)), jails (jail(2)), ACLs,e suporte no kernel a dispositivos de criptografia.

- Funcionalidades avançadas de rede: firewalls, gerenciamento de QoS, rede TCP/IP de alta performance com suporte a muitos recursos avançados.

  O framework Netgraph (netgraph(4)) presente no kernel do FreeBSD, permite que os módulos de rede possam ser conectados entre si de formas flexíveis.

- Suporte para tecnologias avançadas de armazenamento Fibre Channel, SCSI, RAID por software e hardware, ATA e SATA.

  O FreeBSD suporta um grande numero de sistemas de arquivos, e o seu sistema de arquivos nativo UFS2 suporta `soft updates` , `snapshots` e sistemas de arquivos de tamanho muito grandes (até 16 TB por sistema de arquivos) [7].

  O framework GEOM (geom(4)) presente no kernel do FreeBSD permite que módulos de armazenamento sejam compostos de forma flexível.

- Mais de 24,000 aplicativos portados, tanto comerciais quanto de código aberto, gerenciados através da coleção de ports do FreeBSD.

2.3. Estrutura Organizacional

A estrutura organizacional do FreeBSD não é hierárquica

Existem basicamente dois tipos de colaboradores no projeto FreeBSD, os usuários em geral e os desenvolvedores com acesso de escrita (conhecidos como *committers* no jargão) ao repositório de código fonte.

Existem muitos milhares de colaboradores no primeiro grupo, a grande maioria das contribuições para o FreeBSD vêm de indivíduos desse grupo; A permissão de `commit` (acesso de escrita) no repositório é concedida a pessoas que contribuem de forma consistente para o projeto. O direito de `commit` vem acompanhado de responsabilidades adicionais, e para facilitar o aprendizado das mesmas, um mentor é atribuído a todos os novos `committers` .

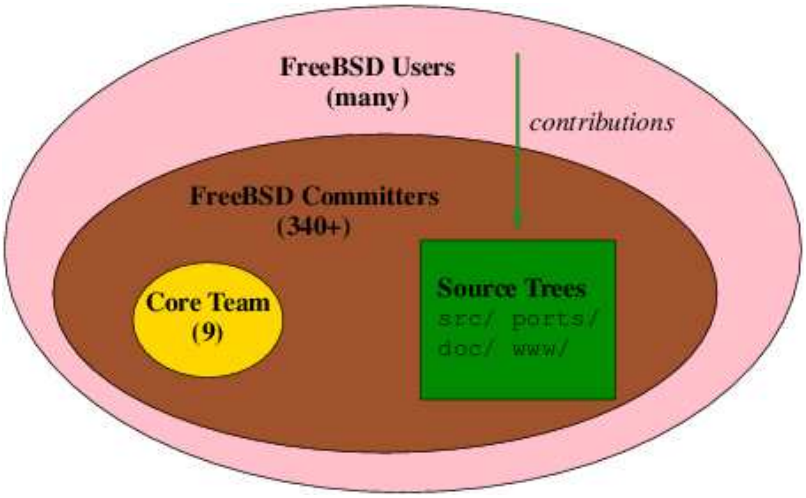

Figura 1. Organização do FreeBSD

A resolução de conflitos é realizada por um "`Core Team`" de 9 pessoas, o qual é eleito a partir do grupo de `committers`.

O FreeBSD não tem `committers` corporativos. Os `committers` são obrigados a assumir de forma individual a responsabilidade pelas mudanças que introduzem no código. O FreeBSD Committer's Guide [3] documenta as regras e responsabilidades que se aplicam aos `committers`.

O modelo do projeto FreeBSD é examinado em detalhes no [10].

2.4. Processos de Engenharia para liberação de novas versões do FreeBSD.

O processo de engenharia para a liberação de uma nova versão do FreeBSD desempenha um papel importante para assegurar que as suas novas versões sejam de alta qualidade. Em qualquer ponto do tempo, os voluntários do FreeBSD suportam múltiplas versões do código sistema (Figura 2, "Ramos de versões do FreeBSD"):

- As novas funcionalidades e os códigos disruptivos entram no ramo de desenvolvimento, também conhecido como ramo *-CURRENT*.
- O ramo *-STABLE* contém linhas de código que são ramificadas a partir do HEAD em intervalos regulares. Apenas código devidamente testado é permitido no ramo -STABLE. Novas funcionalidades são permitidas após terem sido testadas e estabilizadas no ramo -CURRENT.
- O ramo *-RELEASE* é mantido pela equipe de segurança do FreeBSD. Somente correções de bugs críticos são permitidos no ramo -RELEASE.

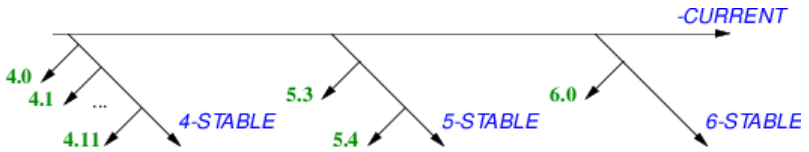

Figura 2. Ramos de versões do FreeBSD

As linhas de código são mantidas vivas enquanto houver interesse dos usuários e dos desenvolvedores nelas.

As arquiteturas de máquina estão agrupadas em "tiers"; As arquiteturas *Tier 1* são totalmente suportadas pelas equipes de engenharia de lançamento e de segurança, as arquiteturas *Tier 2* são suportadas em regime de "melhores esforços", e as arquiteturas experimentais compreendem o *Tier 3*. A lista das arquiteturas suportadas é parte da coleção de documentos do FreeBSD.

A equipe de engenharia de lançamentos publica um `road map` para as versões futuras do FreeBSD no web site do projeto. As datas indicadas no `road map` não são prazos; As novas versões do FreeBSD são liberadas apenas quando o seu código e documentação estão prontos.

O processo de engenharia para a liberação de novas versões do FreeBSD é descrito em detalhes no [14].

3. Colaborando com o FreeBSD

Projetos `open-source` como o FreeBSD oferecem códigos finalizados de altíssima qualidade [4]. Estudos anteriores examinaram o efeito da disponibilidade do código fonte no desenvolvimento de software [2].

Embora o acesso a um código fonte de qualidade possa reduzir o custo inicial de desenvolvimento, a longo prazo, os custos com o gerenciamento de mudanças começam a dominar. A medida que os ambientes computacionais mudam ao longo dos anos e novas vulnerabilidades de segurança são descobertas, o seu produto também precisará mudar e se adaptar. O uso de código open-source não deve ser encarado como uma atividade pontual, mas sim como um *processo contínuo*. Os melhores projetos para se colaborar são os que estão *vivos*, ou seja, aqueles com uma comunidade ativa, que tenha objetivos claros e que possua um estilo de trabalho transparente.

- o FreeBSD tem uma comunidade de desenvolvimento ativa em torno dele. No momento em que este artigo foi escrito, existiam milhares de colaboradores com representantes de praticamente todos os continentes povoados do mundo, e mais de 300 indivíduos com acesso de escrita aos repositórios do projeto.

- Os objetivos do projeto FreeBSD são [6]:

  - Desenvolver um sistema operacional de alta qualidade para o hardware de computadores populares, e,
  - Tornar o nosso trabalho disponível para todos sob uma licença liberal.

- O FreeBSD desfruta de uma cultura aberta e transparente de trabalho. Quase todas as discussões no projeto ocorrem por e-mail, em listas publicas de discussão, que também são arquivadas para a posteridade. As políticas do projeto são documentadas e mantidas sob controle de revisão. A participação no projeto é aberta a todos.

3.1. Compreendendo a cultura do FreeBSD

Para ser capaz de trabalhar de forma eficaz com o projeto FreeBSD, você precisa entender a cultura do projeto.

As regras que regem a operação de um projeto voluntário são diferentes das que regem a operação de uma empresa com fins lucrativos. Um erro comum que as empresas cometem ao se aventurar no mundo open-source é o de desvalorizar essas diferenças.

Motivação. A maioria das contribuições feitas para o FreeBSD são feitas voluntariamente, sem que nenhuma recompensa financeira esteja envolvida. Os fatores que motivam as pessoas são complexos, e vão desde o puro altruísmo até o interesse comum em resolver algum tipo de problema que o FreeBSD esteja tentando resolver. Neste tipo de ambiente, a "elegância jamais é opcional" [11].

Visão de longo prazo. O FreeBSD tem raízes de quase 20 anos para com o trabalho do Grupo de Pesquisa de Ciências da Computação da Universidade da Califórnia, Berkeley.[1] Alguns dos desenvolvedores originais do CSRG permanecem associados com o projeto.

O projeto valoriza perspectivas de longo prazo [12]. Uma sigla encontrada com frequência no projeto é DTRT, a qual significa "Faça a Coisa Certa" (`Do The Right Thing`).

Processo de Desenvolvimento. Programas de computador são ferramentas de comunicacão: em um nível os programadores comunicam as suas intenções usando uma notação precisa para uma ferramenta (um compilador) que traduz as suas instruções para um código executável. Em outro nível, a mesma notação é usada para a comunicação das intenções entre dois programadores.

Especificações formais e documentos de design raramente são utilizados no projeto. Código claro e bem escrito, acompanhado de logs bem escritos para as alterações das (Figura 3, "Um exemplo de entrada no log de alteração"), são usados em seu lugar. O desenvolvimento do FreeBSD acontece por "consenso áspero e por código sendo executado" [1].

[1] O repositório de código fonte do FreeBSD contém a história do projeto desde a sua concepção, e existem CDROMs disponíveis que contém o código anterior do CSRG.

```
bde         2005-10-29 16:34:50 UTC

  FreeBSD src repository

  Modified files:
    lib/msun/src         e_rem_pio2f.c
  Log:
  Use double precision to simplify and optimize arg reduction for ↺
small
  and medium size args too: instead of conditionally subtracting a ↺
float
  17+24, 17+17+24 or 17+17+17+24 bit approximation to pi/2, always
  subtract a double 33+53 bit one.  The float version is now closer ↺
to
  the double version than to old versions of itself — it uses the ↺
same
  33+53 bit approximation as the simplest cases in the double ↺
version,
  and where the float version had to switch to the slow general ↺
case at
  -|x| == 2^7*pi/2, it now switches at -|x| == 2^19*pi/2 the same ↺
as the
  double version.

  This speeds up arg reduction by a factor of 2 for -|x| between ↺
3*pi/4 and
  2^7*pi/4, and by a factor of 7 for -|x| between 2^7*pi/4 and ↺
2^19*pi/4.

  Revision  Changes    Path
  1.14      +22 --97    src/lib/msun/src/e_rem_pio2f.c
```

Figura 3. Um exemplo de entrada no log de alteração

A comunicação entre os programadores é reforçada pelo uso de um style(9) padrão de codificação, comum entre eles.

Canais de Comunicação. Os colaboradores do FreeBSD estão espalhados por todo o mundo. O email (e em menor extensão, o IRC) é o meio de comunicação preferido no projeto.

3.2. Melhores práticas para colaborar com o projeto FreeBSD.

Agora iremos examinar algumas das melhores práticas para se fazer um melhor uso do FreeBSD no desenvolvimento de produtos.

Se planeje para o longo prazo
: Implante processos que o ajudem a monitorar o desenvolvimento do FreeBSD. Por exemplo:

Acompanhe o código fonte do FreeBSD. O projeto facilita o espelhamento do seu repositório CVS usando o CVSup. Ter o histórico completo do código fonte é útil quando se está debugando problemas complexos e oferece informações valiosas sobre as intenções dos desenvolvedores originais. Utilize um sistema de controle de código que lhe permita mesclar facilmente as alterações entre o código original do FreeBSD e o seu próprio código.

A Figura 4, "Código fonte exibindo a listagem de anotações gerada utilizando o `cvs annotate` " mostra as anotações em uma parte do arquivo referenciado pelo log de alterações da Figura 3, "Um exemplo de entrada no log de alteração". A ascendência de cada linha de código é claramente visível. Listagens com as anotações mostrando a história de cada arquivo que faz parte do FreeBSD estão disponíveis na web.

```
#LINE #REV         #WHO      #DATE       #TEXT

62    1.1          (jkh      19-Aug-94):      int32_t ↺
__ieee754_rem_pio2f(float x, float *y)
63    1.1          (jkh      19-Aug-94): {
64    1.14         (bde      29-Oct-05):      double z,w,t,r,fn;
65    1.13         (bde      29-Oct-05):      double tx[3];
66    1.14         (bde      29-Oct-05):      int32_t ↺
e0,i,nx,n,ix,hx;
67    1.1          (jkh      19-Aug-94):
68    1.1          (jkh      19-Aug-94):      ↺
GET_FLOAT_WORD(hx,x);
69    1.1          (jkh      19-Aug-94):      ix = hx&0x7fffffff;
70    1.1          (jkh      19-Aug-94):      ↺
if(ix<=0x3f490fd8)   -/* -|x| ~<= pi/4 -, no need for reduction */
71    1.1          (jkh      19-Aug-94):          {y[0] = x; ↺
y[1] = 0; return 0;}
72    1.14         (bde      29-Oct-05):      -/* 33+53 bit pi ↺
is good enough for special and medium size cases */
73    1.2          (bde      07-Apr-95):      if(ix<0x4016cbe4) ↺
{  -/* -|x| < 3pi/4, special case with n=+-1 */
74    1.14         (bde      29-Oct-05):          if(hx>0) {
75    1.15         (bde      06-Nov-05):              z = x -- ↺
pio2;
76    1.15         (bde      06-Nov-05):              n = 1;
77    1.15         (bde      06-Nov-05):          -} else {
78    1.15         (bde      06-Nov-05):              z = x + ↺
pio2;
79    1.15         (bde      06-Nov-05):              n = 3;
80    1.9          (bde      08-Oct-05):          -}
81    1.15         (bde      06-Nov-05):          y[0] = z;
82    1.15         (bde      06-Nov-05):          y[1] = z -- ↺
y[0];
83    1.15         (bde      06-Nov-05):          return n;
84    1.15         (bde      06-Nov-05):      -}
85    1.15         (bde      06-Nov-05):      if(ix<0x407b53d1) ↺
{  -/* -|x| < 5*pi/4, special case with n=+-2 */
```

Figura 4. Código fonte exibindo a listagem de anotações gerada utilizando o `cvs annotate`

Nomeie um guardião. Nomeie um guardião (*gatekeeper*) para monitorar o desenvolvimento do FreeBSD, para manter-se atento a mudanças que poderiam potencialmente afetar os seus produtos.

Comunique os erros que encontrar de volta para o projeto. Se você encontrar um bug no código do FreeBSD que você está utilizando, envie um relatório de problema. Este procedimento simples irá ajudar a garantir que você não precisará corrigir o

erro novamente da próxima vez que precisar importar novamente do código base do FreeBSD.

Se alavanque nos esforços de engenharia do FreeBSD para lançamento de novas versões.
: Utilize código do ramo de desenvolvimento -STABLE do FreeBSD. Este ramo de desenvolvimento é formalmente suportado pelas equipes de engenharia de lançamento e de segurança, e é formada apenas por código testado.

Doe código para reduzir seus custos
: Uma parte significativa dos custos relacionados ao desenvolvimento de um produto é o de realizar a sua manutenção. Ao doar partes não criticas do seu código para o projeto, você se beneficia por ter o seu código exposto de uma forma ampla, exposição que ele não teria de outra forma. Esta exposição por sua vez leva eliminação de um maior numero de bugs e de vulnerabilidades de segurança, e permite que anomalias de desempenho sejam identificadas e corrigidas.

Obtenha suporte efetivo
: Para produtos com prazos apertados, é recomendado que você contrate o suporte ou consultoria de um desenvolvedor ou empresa com experiência em FreeBSD. A lista de discussão sobre empregos relacionados ao FreeBSD é um canal de comunicação muito útil para se encontrar talentos. O projeto FreeBSD mantém uma galeria de consultores e empresas de consultoria que trabalham com FreeBSD. O Grupo de Certificação BSD oferece certificação para todos os principais sistemas operacionais derivados do BSD.

  Para as necessidades menos importantes, você pode pedir ajuda nas listas de discussão do projeto. Um guia útil para seguir quando precisar pedir está listado em [13].

Divulgue o seu envolvimento
: Você não é obrigado a divulgar que faz uso do FreeBSD, mas ao fazê-lo você estará ajudando ambos os esforços, o seu e o do projeto.

  Dar visibilidade para a comunidade FreeBSD de que a sua empresa utiliza o sistema ajuda a melhorar as suas chances de atrair talentos de alta qualidade. Quanto maior for a lista de organizações que apoiam o FreeBSD maior será a presença do sistema na cabeça (`mind share`) dos desenvolvedores. Ao contribuir para aumentar o numero de desenvolvedores interessados no FreeBSD, você estará gerando uma base saudável para o seu futuro.

Suporte os desenvolvedores do FreeBSD
: Às vezes, o caminho mais direto para ver uma funcionalidade que você deseja implementada no FreeBSD é suportar um desenvolvedor que já esteja olhando um problema relacionado. A ajuda pode variar de uma doação de hardware até uma assistência financeira direta. Em alguns países, as doações para o projeto FreeBSD usufruem de benefícios fiscais. O projeto possui um canal de comunicação dedicado

para assuntos relacionados a doações e para ajudar os doadores. O projeto também mantém uma página web na qual os desenvolvedores podem listar suas necessidades.

Por uma política do projeto, o FreeBSD reconhece todas as contribuições recebidas em seu site web.

4. Conclusão

O Objetivo do projeto FreeBSD é criar e distribuir o código fonte de um sistema operacional de alta qualidade. Ao trabalhar com o projeto FreeBSD você pode reduzir os seus custos de desenvolvimento e melhorar o tempo necessário para lançar seus novos produtos no mercado em vários cenários de desenvolvimento de produtos.

Foram examinadas as características do FreeBSD que o tornam uma excelente opção na estratégia de produto de uma organização. Em seguida, abordamos os aspectos predominantes da cultura do projeto e examinamos formas eficazes de interagir com os seus desenvolvedores. O artigo finaliza com uma lista das melhores práticas que podem ajudar na colaboração da iniciativa privada com o projeto FreeBSD.

Bibliografia

[1] *The Architectural Principles of the Internet*. B. Carpenter. Copyright © 1996.

[2] *How is Open-Source Affecting Software Development?*. Diomidis Spinellis e Clemens Szyperski. *IEEE Computer*. Copyright © Jan/Feb 2004. IEEE Computer Society.

[3] *Committer's Guide*. . Copyright © 2005.

[4] *Coverity study on kernel security holes in Linux and FreeBSD*. . Copyright © 2005.

[5] *Innovation Happens Elsewhere: Open Source as Business Strategy*. Ron Goldman e Richard Gabriel. Copyright © 2005. ISBN 1558608893. Morgan-Kaufmann.

[6] *Contributing to the FreeBSD Project*. Jordan Hubbard. Copyright © 1994—2005. The FreeBSD Project.

[7] *Soft Updates: A Technique for Eliminating Most Synchronous Writes in the Fast Filesystem*. Kirk McKusick e Gregory Ganger. USENIX Annual Technical Conference. . Copyright © 1999.

[8] *Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable*. Marshall Kirk McKusick. *Open Sources: Voices from the Open Source Revolution*. ISBN 1-56592-582-3. O'Reilly Inc.. Copyright © 1993.

[9] *Why you should use a BSD style license for your Open Source Project*. Bruce Montague. The FreeBSD Project. Copyright © 2005.

[10] *A project model for the FreeBSD Project*. Niklas Saers. Copyright © 2005. The FreeBSD Project.

[11] *Tutorial on Good Lisp Programming Style*. Peter Norvig e Kent Pitman. Copyright © 1993.

[12] *Teach Yourself Programming in Ten Years*. Peter Norvig. Copyright © 2001.

[13] *How to ask questions the smart way*. Eric Steven Raymond. Copyright © 2004.

[14] *FreeBSD Release Engineering*. Murray Stokely. Copyright © 2001. The FreeBSD Project.