Construire des produits avec FreeBSD

Résumé

Le projet FreeBSD est un projet international, collaboratif et basé sur le volontariat, qui développe un système d'exploitation portable et de grande qualité. Le projet FreeBSD distribue le code source de ses produits avec une licence libérale dans l'intention d'encourager l'utilisation de son code. Collaborer avec le project FreeBSD peut aider les organisations à réduire leur délai de mise sur le marché, leurs coûts de développement, et améliorer la qualité de leurs produits.

Cet article se penche sur les questions relatives à l'utilisation du code de FreeBSD dans les appareils informatiques et les logiciels. Il met en évidence les caractéristiques de FreeBSD qui en font un excellent support pour le développement de produits. Cet article conclut en suggérant quelques "bonnes pratiques" pour les organisations qui collaborent avec le projet FreeBSD.

Version française de Frederic Culot <culot@freebsd.org>.

Table des matières

1. Introduction 1
2. FreeBSD en tant que brique constitutive 3
3. Collaborer avec FreeBSD 7
4. Conclusion 11
Bibliography 11

1. Introduction

FreeBSD est reconnu aujourd'hui comme un système d'exploitation hautes performances pour serveurs. Il est déployé sur des millions de serveurs web et de machines connectées à internet de part le monde. Le code de FreeBSD fait aussi partie intégrante de beaucoup de produits depuis des appareils comme les routeurs réseau, pare-feux, et dispositifs de stockage, jusqu'aux ordinateurs personnels. Des parties de FreeBSD ont également été utilisées dans des produits commerciaux (voir FreeBSD en tant que brique constitutive).

Dans cet article nous nous intéressons au projet FreeBSD en tant que ressource pour la conception logicielle-une collection de briques de base et de processus que vous pouvez utiliser pour construire d'autres produits.

Bien que les sources de FreeBSD soient distribuées librement au public, les organisations ont besoin de *collaborer* avec le projet pour pouvoir pleinement apprécier les bénéfices de ce travail. Dans les sections suivantes de cet article nous présentons les moyens efficaces qui existent afin de

collaborer avec le projet, ainsi que les pièges à éviter.

Avertissement au lecteur. L'auteur pense que les caractéristiques du Projet FreeBSD telles que décrites dans cet article sont en grande partie vraies au moment où cet article a été conçu et rédigé (2005). Cependant, le lecteur doit garder en tête que les pratiques et processus utilisés par les communautés open-source peuvent changer au cours du temps, et que les informations contenues dans cet article devraient donc être considérées comme étant indicatives plutôt que prescriptives.

1.1. Public visé

Ce document pourrait présenter un intérêt pour les groupes de personnes suivants:

- Les preneurs de décisions dans les entreprises qui recherchent à améliorer la qualité de leurs produits, à réduire leur délai de mise sur le marché, et réduire leurs coûts de développement sur le long terme.
- Les consultants en technologie à la recherche de bonnes pratiques pour tirer profit de l'"open-source".
- Les spécialistes de l'industrie intéressés par la compréhension de la dynamique des projets open-source.
- Les développeurs logiciels cherchant à utiliser FreeBSD et désirant contribuer au projet en retour.

1.2. Objectifs de l'article

La lecture de cet article devrait vous apporter:

- Une compréhension des buts du Projet FreeBSD ainsi que de la structure de son organisation.
- Un aperçu des technologies disponibles dans le projet.
- Une compréhension de son modèle de développement et de ses processus d'ingénierie.
- Une compréhension des différences entre les processus de développement conventionnels que l'on retrouve chez les éditeurs de logiciels et ceux utilisés par le projet FreeBSD.
- Une sensibilisation aux canaux de communication utilisés par le projet et le niveau de transparence auquel vous pouvez vous attendre.
- Une connaissance des moyens optimaux qui existent pour travailler avec le projet-comment réduire au maximum les coûts de développement, améliorer le délai de mise sur le marché, gérer les failles de sécurité, et préserver la compatibilité future de votre produit avec les évolutions du projet FreeBSD.

1.3. Structure de l'article

La suite de l'article est structurée de la façon suivante:

- FreeBSD en tant que brique constitutive introduit le projet FreeBSD, présente sa structure organisationnelle, ses technologies clés et ses processus de développement.

- Collaborer avec FreeBSD décrit les moyens de collaborer avec le projet FreeBSD. Les pièges les plus courants rencontrés par les sociétés travaillant avec les projets basés sur le volontariat comme FreeBSD sont également présentés.
- Conclusion conclut.

2. FreeBSD en tant que brique constitutive

FreeBSD représente une excellente base sur laquelle construire des produits:

- Le source code de FreeBSD est distribué avec une licence BSD libérale qui facilite grandement son utilisation dans les produits commerciaux [Mon2005].
- Le projet FreeBSD a d'excellentes pratiques de développement qui peuvent être mises à profit.
- Le projet offre une transparence exceptionnelle eu égard à son fonctionnement, permettant aux companies utilisant son code de planifier efficacement l'avenir.
- La culture du projet FreeBSD, héritée du Groupe de Recherche sur la Science Informatique de l'Université de Berkeley en Californie [McKu1999-1], encourage le travail de grande qualité. Certaines fonctionnalités de FreeBSD sont considérés comme des références.

[GoldGab2005] examine avec plus de détails les raisons commerciales qui justifient l'utilisation de l'open-source. Les bénéfices que les sociétés peuvent tirer de l'utilisation de composants FreeBSD dans leurs produits comprennent un délai réduit de mise sur le marché, ainsi qu'une réduction des coûts et des risques liés au développement.

2.1. Construire avec FreeBSD

Voici quelques utilisations que des sociétés ont faites de FreeBSD:

- Comme source de code testé pour des bibliothèques ou utilitaires.

  En étant "en aval" du projet, les organisations tirent profit des nouvelles fonctionnalités, corrections de bogues et tests dont le code en amont bénéficie.

- En tant que système d'exploitation embarqué (par exemple, pour un routeur OEM ou un appareil servant de pare-feu). Dans ce modèle, les organisations utilisent un noyau FreeBSD adapté ainsi qu'un ensemble de logiciels appropriés conjointement avec une couche propriétaire de gestion de leur appareil. Les OEMs bénéficient des nouveaux supports matériels ajoutés par le projet FreeBSD en amont, ainsi que des tests effectués sur le système de base.

  FreeBSD est diffusé avec un environnement de développement auto-hébergé qui permet de créer facilement de telles configurations.

- En tant qu'environnement compatible UNIX® pour les fonctions de gestion des environnements de stockage et les appareils réseau, fonctionnant sur un "serveur lame" séparé.

  FreeBSD fournit les outils nécessaires pour créer des systèmes d'exploitation dédiés et des images d'applications. Son implémentation basée sur une API BSD UNIX® est mature et testée. FreeBSD peut aussi fournir un environnement de développement croisé stable pour les autres

composants de l'appareil final.

- En tant que moyen d'obtenir une large base de tests et du support de la part d'une équipe de développeurs internationale pour tout ce qui a trait à la "propriété intellectuelle" non critique.

  Dans ce modèle, les organisations apportent un ensemble d'infrastructures utiles au projet FreeBSD (voir par exemple netgraph(3)). L'importante exposition que le code acquiert aide pour l'identification rapide de problèmes de performance et de bogues. L'implication d'excellents développeurs apporte aussi des ajouts utiles à la base existante, ce dont l'organisation contributrice bénéficie également.

- En tant qu'environnement de développement autorisant le développement croisé pour des systèmes embarqués tels que RTEMS et eCOS.

  Il existe une pléthore d'environnements de développement très complets dans le catalogue des 37000 applications portées et empaquetées pour FreeBSD.

- Comme moyen de fournir une API Unix dans un système propriétaire par ailleurs, augmentant ainsi son attractivité pour les développeurs d'applications.

  Dans ce cas des parties du noyau FreeBSD et des applications sont "portées" pour tourner conjointement avec d'autres tâches du système d'exploitation propriétaire. La disponibilité de l'implémentation d'une API Unix™ stable et bien testée peut réduire l'effort nécessaire pour porter des applications populaires sur le système propriétaire. Comme FreeBSD est fournit avec une documentation de grande qualité concernant ses mécanismes internes et assure une gestion efficace des vulnérabilités et des cycles de développement, les coûts pour se maintenir à jour sont bas.

2.2. Technologies

Le projet FreeBSD supporte un grand nombre de technologies dont une sélection est présentée ci-dessous:

- Un système complet qui peut faire de l'auto-hébergement croisé pour les architectures suivantes: alpha (jusqu'à FreeBSD version 6.X), amd64, ia64, i386, sparc64, powerpc (voir build(7)).
- Le support pour les technologies, protocoles et standards suivants: 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™.
- Un noyau modulaire permettant le traitement symétrique multiprocesseurs, avec chargement possible de modules noyau et un système de configuration facile à utiliser.
- Le support pour l'émulation de Linux™ et des binaires SVR4 à vitesse quasi-native et le support pour les pilotes réseau Windows™ (NDIS).
- Des librairies pour de nombreuses tâches liées à la programmation: archivage, support FTP et HTTP, support des processus légers en plus d'un environnement de programmation POSIX™.
- Des dispositifs de sécurité avancés : Mandatory Access Control (mac(9)), jails (jail(2)), ACLs, ainsi

que le support d'un dispositif cryptographique au niveau noyau.

- Des caractéristiques réseau avancées : dispositifs pares-feu, gestion de Qos, communications TCP/IP hautes performances avec support de nombreuses caractéristiques avancées.

  Le système Netgraph (netgraph(4)) présent dans le noyau FreeBSD permet à des modules noyau de gestion des communications réseau d'être interconnectés de manière flexible.

- Le support pour des technologies de stockage avancées: fibre, SCSI, RAID logiciel et matériel, ATA et SATA.

  FreeBSD est capable de gérer plusieurs systèmes de fichiers différents, et son support natif du système de fichiers UFS2 autorise les soft updates, les sauvegardes instantanées, ainsi que les systèmes de fichiers très volumineux (16TB par système) [McKu1999].

  Le système GEOM (geom(4)) présent dans le noyau FreeBSD permet de composer de manière flexible des modules noyau dédiés à la gestion du stockage.

- L'accès à plus de 37000 applications portées, qu'elles soient commerciales ou open-source, gérées grâce à la collection des portages de FreeBSD.

2.3. Structure organisationnelle

La structure organisationnelle de FreeBSD n'est pas hiérarchique.

Il existe essentiellement deux types de contributeurs à FreeBSD, les utilisateurs de FreeBSD, et les développeurs qui ont les droits en écriture (connus sous le terme *committers* dans notre jargon) et peuvent modifier les sources.

Il existe plusieurs milliers de contributeurs dans le premier groupe, la vaste majorité des contributions à FreeBSD proviennent de personnes faisant partie de ce groupe. Les droits de commit (droits d'accès en écriture) sont accordés aux personnes qui contribuent au projet de manière récurrente. Ces droits viennent avec des responsabilités supplémentaires, et les nouveaux committers se voient attribuer des mentors pour les aider à apprendre les bases.

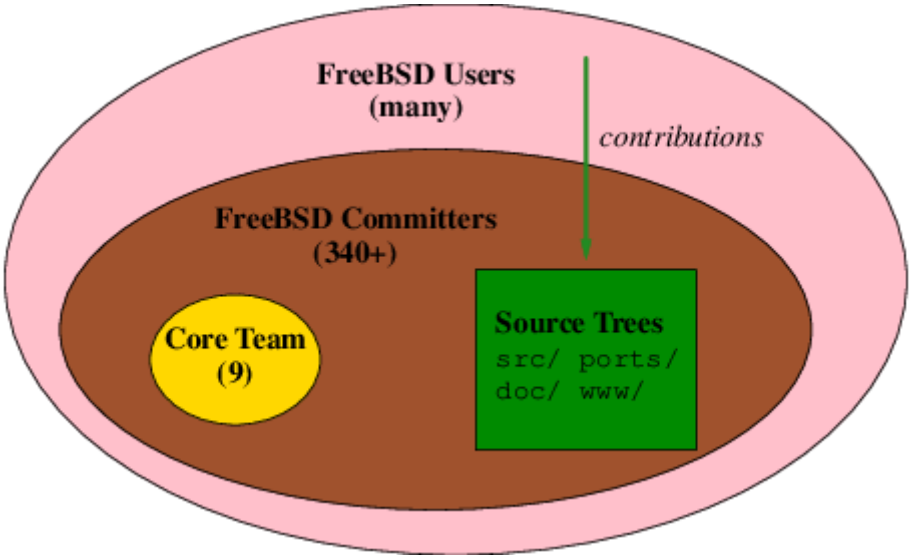

Figure 1. L'organisation FreeBSD

La résolution des conflits est assurée par une équipe ("Core Team") de neuf membres qui est élue par le groupe des committers.

Les committers ne sont pas employés par FreeBSD. Il est exigé de la part des committers qu'ils prennent la responsabilité des changements qu'ils introduisent dans le code. Le Guide du Committer FreeBSD [ComGuide] documente les règles et responsabilités des committers.

Le modèle de projet de FreeBSD est examiné en détails dans [Nik2005].

2.4. Les processus de développement des versions de FreeBSD

Les processus de développement des versions de FreeBSD jouent un rôle majeur en assurant que les versions qui sont délivrées sont de grande qualité. À n'importe quel moment que l'on considère, les volontaires de FreeBSD assurent le développement de plusieurs branches de code (Les branches FreeBSD):

- Les nouvelles fonctionnalités et le code expérimental sont incorporés sur la branche de développement, aussi connue sous le nom de branche *-CURRENT*.
- Les branches *-STABLE* représentent les lignes de code qui sont reprises de la HEAD à des intervalles de temps réguliers. Seul le code testé est autorisé sur une branche -STABLE. Les nouvelles fonctionnalités sont autorisées une fois qu'elles ont été testées et stabilisées sur la branche -CURRENT.
- Les branches *-RELEASE* sont maintenues par l'équipe sécurité de FreeBSD. Seuls les correctifs de bogues pour les problèmes critiques sont autorisés sur les branches -RELEASE.

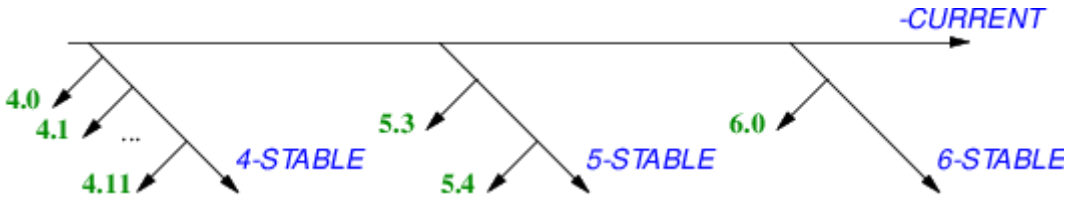

Figure 2. Les branches FreeBSD

Les lignes de code sont maintenues aussi longtemps qu'il existe des utilisateurs et des développeurs qui s'y intéressent.

Les architectures machine sont groupées en "niveaux". Les architectures de premier niveau (*Tier 1*) sont entièrement supportées par l'équipe en charge des versions et l'équipe sécurité. Les architectures de second niveau (*Tier 2*) sont supportées dans la mesure du possible, et les architectures expérimentales représentent le *Tier 3*. La liste des architectures supportées est incluse dans la documentation FreeBSD.

L'équipe en charge des versions publie une feuille de route pour les futures versions de FreeBSD sur la page web du projet. Les dates qui sont mentionnées sur la feuille de route ne sont pas des dates butoirs: les versions de FreeBSD sont délivrées lorsque son code et sa documentation sont prêts.

Les processus de développement des versions de FreeBSD sont décrits dans [RelEngDoc].

3. Collaborer avec FreeBSD

Les projets Open-source tels que FreeBSD offrent un code de très grande qualité [Cov2005]. Des études ont examiné les effets de la disponibilité du code source sur le développement logiciel [Com2004].

Alors que l'accès à du code source de qualité peut réduire les coûts initiaux de développement, les coûts liés à la gestion des changements deviennent prédominants par la suite. Comme les environnements informatiques changent au fil du temps et que de nouvelles failles de sécurité sont découvertes, votre produit lui aussi a besoin de changements et d'adaptations. Utiliser du code open-source se conçoit plus comme un processus *continu dans le temps* que comme quelque chose de ponctuel. Les meilleurs projets avec lesquels collaborer sont ceux qui sont *actifs*, c'est-à-dire ceux qui ont une communauté active, des objectifs clairs et des méthodes de travail transparentes.

- FreeBSD possède une communauté active de développeurs gravitant autour du projet. Au moment de l'écriture de cet article, il existe plusieurs milliers de contributeurs vivant sur tous les continents peuplés de la planète et plus de 300 personnes possédant les droits d'accès aux dépôts des sources du projet.
- Les objectifs du projet FreeBSD sont [Hub1994]:
  - De développer un système d'exploitation de grande qualité pour les architectures informatiques populaires, et
  - Mettre notre travail à la disposition de tous sous couvert d'une licence libérale.
- FreeBSD bénéficie d'une culture ouverte et transparente. Quasiment toutes les discussions au sein du projet se font par courrier électronique, sur les listes publiques qui sont aussi archivées pour la postérité. Les règles et pratiques du projet sont documentées et maintenues en utilisant un système de gestion de versions. Participer au projet est ouvert à tous.

3.1. Comprendre la culture FreeBSD

Afin de pouvoir travailler de manière efficace avec le projet FreeBSD, vous devez comprendre la culture qui règne au sein du projet.

Les projets menés par des volontaires fonctionnent avec des règles différentes de celles utilisées par des organisations commerciales. Une des erreurs récurrentes faite par les entreprises lorsqu'elles s'aventurent dans le monde de l'open-source est de sous-estimer ces différences.

Motivation. La plupart des contributions à FreeBSD sont faites de manière volontaire et aucune rétribution financière n'entre en jeu. Les facteurs qui motivent les contributeurs sont complexes, et parmi ceux-ci on peut citer l'altruisme ou un intérêt pour résoudre les genres de problèmes que FreeBSD tente de résoudre. Dans cette environnement, "l'élégance n'est jamais optionnelle"[Nor1993].

La Vision à Long Terme. Les origines de FreeBSD remontent à presque vingt ans dans le passé avec le travail effectué au Groupe de Recherche en Science Informatique (CSRG) de l'Université de Berkeley en Californie.[1]Certains des développeurs originaux du CSRG sont toujours associés au projet.

Le projet met l'accent sur les perspectives à long terme [Nor2001]. Un acronyme fréquemment rencontré au sein du projet est DTRT qui signifie "Do The Right Thing" (Faites les Choses Correctement).

Les Processus de Développement. Les programmes informatiques sont des outils de communication: à un certain niveau les programmeurs communiquent leurs intentions, en utilisant une notation précise, à un outil (un compilateur) qui traduit ces instructions en code exécutable. À un autre niveau, la même notation est utilisée entre deux programmeurs pour communiquer leurs intentions.

Les spécifications formelles et les documents d'architecture sont rarement utilisés dans le projet. Du code clair et bien écrit ainsi que des rapports de changements (Un example de rapport de modification) eux aussi bien écrits sont utilisés à la place. Le développement de FreeBSD commence par "une ébauche de consensus et en faisant tourner du code"[Carp1996].

```
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
```

Un example de rapport de modification

La communication entre programmeurs est facilitée par l'utilisation d'un standard commun concernant le code style(9).

Les canaux de communication. Les contributeurs FreeBSD sont répartis dans le monde entier. Le courrier électronique (et dans une moindre mesure, l'IRC) est le moyen de communication prépondérant au sein du projet.

3.2. Les meilleures pratiques pour collaborer avec le projet FreeBSD

Nous nous intéressons maintenant à quelques bonnes pratiques utiles pour tirer profit au maximum de l'utilisation de FreeBSD pour le développement de produits.

Plan à long terme

Mettre en place des processus qui simplifient le suivi du développement de FreeBSD. Par example:

Suivre les changements dans le code source de FreeBSD. Le projet rend la copie de son dépôt CVS aisée grâce à l'utilisation de CVSup. Avoir l'historique complet des sources est utile lors du déboguage de problèmes complexes et offre des indications utiles sur les intentions des développeurs. Utilisez un système de contrôle de sources efficace qui vous permette de facilement fusionner les changements entre le code FreeBSD et votre propre code.

La Un listing annoté généré par `cvs annotate` montre une partie d'un listing annoté du fichier dont le rapport de changement de la Un example de rapport de modification fait référence. L'origine de chacune des lignes du code source est clairement affichée. Les listings annotés montrant l'historique de chacun des fichiers faisant partie de FreeBSD sont disponibles sur Internet.

```
#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 */
```

Un listing annoté généré par `cvs annotate`

Utilisez un observateur. Nommez un *observateur* pour surveiller les développements de FreeBSD, pour déceler les changements qui pourraient potentiellement impacter vos produits.

Remontez les bogues en amont. Si vous détectez un bogue dans le code FreeBSD que vous utilisez, remplissez un rapport de bogue. Cette étape permet de faire en sorte que vous n'ayez pas à corriger le même bogue la prochaine fois que vous récupérerez les sources en amont.

Tirez profit des efforts portés sur la gestion des versions

Utilisez du code d'une branche de développement -STABLE de FreeBSD. Ces branches de développement sont officiellement supportées par les équipes en charge des versions et les équipes sécurité de FreeBSD et sont constituées de code testé.

Donnez du code pour réduire les coûts

La majeure partie des coûts associés au développement est liée à la maintenance. En donnant du code non critique au projet, vous bénéficiez du fait que votre code aura une diffusion bien plus importante que celle qu'il aurait eu sans ça. Ceci amène à ce que plus de bogues et de failles de sécurité soient éliminés et que les problèmes de performance soient identifiés et résolus.

Recevez un soutien efficace

Pour les produits avec des dates butoirs rapprochées, il est recommandé d'embaucher ou de s'attacher les services d'un développeur ou d'une firme qui a de l'expérience avec FreeBSD. La liste de diffusion sur les emplois en rapport avec FreeBSD est un canal de communication utile si vous êtes à la recherche de talents dans le domaine. Le projet FreeBSD maintient une liste des consultants et des firmes de consulting assurant des travaux liés à FreeBSD. Le Groupe de Certification FreeBSD propose des certifications pour la majorité des systèmes d'exploitation dérivés de BSD.

Pour les besoins moins critiques, vous pouvez demander de l'aide sur les listes de diffusion du projet. Un guide utile à suivre si vous souhaitez demander de l'aide est celui de [Ray2004].

Faites de la publicité sur votre engagement

Vous n'êtes pas obligé de faire de la publicité sur votre utilisation de FreeBSD, mais le faire permet à la fois de vous aider vous mais aussi le projet.

Faire valoir auprès de la communauté FreeBSD que votre société utilise FreeBSD améliore vos chances de pouvoir attirer des personnes talentueuses. Une longue liste de personnes habilitées à faire du support sur FreeBSD signifie aussi plus d'échanges d'idées entre les développeurs. Ceci permet de construire des fondations plus seines pour votre futur.

Soutenez les développeurs de FreeBSD

Parfois la manière la plus directe pour qu'une fonctionnalité dont on a besoin soit incluse dans FreeBSD est d'aider un développeur qui travaille déjà sur un problème ayant un rapport avec

cette fonctionnalité. Ces aides peuvent prendre plusieurs formes, depuis le don de matériel jusqu'à des donations financières. Dans certains pays, les donations au projet FreeBSD peuvent bénéficier d'avantages au niveau des impôts. Le projet a un interlocuteur dédié pour assister les donateurs. Le projet maintien également une page web sur laquelle les développeurs recensent leurs besoins.

Le projet FreeBSD met un point d'honneur à remercier tous les donnateurs sur son site web.

4. Conclusion

Les objectifs du projet FreeBSD sont de créer et proposer gratuitement le code source d'un système d'exploitation de grande qualité. En travaillant avec le projet FreeBSD vous pouvez réduire vos coûts de développement et améliorer vos délais de mise sur le marché dans un certain nombre de scénarios de développement de produits.

Nous avons passé en revue les caractéristiques du projet FreeBSD qui en font un excellent choix pour faire partie d'une stratégie produit d'une entreprise. Nous avons ensuite présenté la culture du projet et examiné les différents moyens à disposition pour interagir avec ses développeurs. Cet article conclut avec une liste des bonnes pratiques qui peuvent être mises en place par les organisations pour collaborer avec le projet.

Bibliography

[Carp1996] The Architectural Principles of the Internet B. Carpenter. The Internet Architecture Board.The Internet Architecture Board. Copyright® 1996.

[Com2004] How is Open-Source Affecting Software Development? Diomidis Spinellis. Clemens Szyperski.IEEE Computer Copyright® Jan/Feb 2004. IEEE Computer Society.

[ComGuide] Committer's Guide The FreeBSD Project. Copyright® 2005.

[Cov2005] Coverity study on kernel security holes in Linux and FreeBSDCoverity Inc.. Copyright® 2005.

[GoldGab2005] Innovation Happens Elsewhere: Open Source as Business Strategy Ron Goldman. Richard Gabriel. Copyright® 2005. Morgan-Kaufmann.

[Hub1994] Contributing to the FreeBSD Project Jordan Hubbard. Copyright® 1994-2005. The FreeBSD Project.

[McKu1999] Soft Updates: A Technique for Eliminating Most Synchronous Writes in the Fast Filesystem Kirk McKusick. Gregory Ganger. Copyright® 1999.

[McKu1999-1] Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable Marshall Kirk McKusick. Open Sources: Voices from the Open Source Revolution O'Reilly Inc.. Copyright® 1993.

[Mon2005] Why you should use a BSD style license for your Open Source Project Bruce Montague. The FreeBSD Project. Copyright® 2005.

[Nik2005] A project model for the FreeBSD Project Niklas Saers. Copyright® 2005. The FreeBSD Project.

[Nor1993] Tutorial on Good Lisp Programming Style Peter Norvig. Kent Pitman. Copyright® 1993.

[Nor2001] Teach Yourself Programming in Ten Years Peter Norvig. Copyright® 2001.

[Ray2004] How to ask questions the smart way Eric Steven Raymond. Copyright® 2004.

[RelEngDoc] FreeBSD Release Engineering Murray Stokely. Copyright® 2001. The FreeBSD Project.

[1] Le dépôt des sources de FreeBSD contient l'historique du projet depuis sa création, et il existe des CDROMs qui contiennent du code plus ancien en provenance du CSRG.