問題回報(PR)的處理原則

摘要

這篇文章主要在講：由 FreeBSD PR 維護小組所提出的一些 FreeBSD 問題回報(PR) 建議，希望大家在弄 PR 時都能遵守。

目

1. 前言 . . . 1
2. 問題回報(PR)的生命週期 . . . 1
3. 問題回報(PR)的狀態 . . . 2
4. 問題回報(PR)的種類 . . . 2
5. Unassigned PRs . . . 3
6. Assigned PRs . . . 5
7. 重複的 PR . . . 5
8. Stale PRs . . . 5
9. Misfiled PRs . . . 6
10. Followups misfiled as new PRs . . . 6
11. PRs misfiled because of missing fields . . . 7
12. Misfiled PRs that are not really problem reports . . . 8
13. 延伸閱讀 . . . 8

1. 前言

GNATS 是 FReeBSD 計劃所採用的一套專門管理錯誤(回報bug) 系統。 由於對 FreeBSD 品質保證而言，是否能準確掌握各項錯誤回報與進度是十分重要的， 因此，如何正確有效使用 GNATS 也就必須注意。

Access to GNATS is available to FreeBSD developers, as well as to the wider community. 為了讓 GNATS 資料庫使用上儘量一致，於是就產生了怎麼處理像是：followup(回文)、關閉PR等的參考原則。

2. 問題回報(PR)的生命週期

- 首先，回報者(originator)以 send-pr(1) 送出 PR，然後會收到一封確認信。
- 然後，committer 們就會有人(假設叫做 Joe)發掘有興趣的 PR 並將該 PR 指派給自己來處理。 或者 bugbuster 會有人(假設叫做 Jane) 就會下決定：她覺得 Joe 比較適合處理，就將該 PR 指派(assign)給他
- Joe 會先與有問題的回報者作些意見交流(以確定這問題有進入 audit 追蹤流程內) 以及判斷問題點。 然後再確定問題點有寫入 audit 追蹤流程之後，然後把該 PR 狀態設為 "analyzed(已分析)"。
- Joe 開始徹夜找出問題解法，然後將 patch 送到 follow-up(回文用)，並請回報者協助測試是否正常。 然後，他就會將 PR 狀態設為 "feedback" 囉。
- 如此重複 analyzed、feedback 幾趟之後，直到 Joe 與回報者雙方都相當滿意 patch 結果， 於是就會將 patch 給 commits 進入 -CURRENT (或者若 -CURRENT 上面沒這問題的話，就直接送到 -STABLE)，在 commit log 內要把相關 PR 寫上去 (同時回報者若有送完整或部分 patch 的話，就順便記載)，然後，若沒什麼事的話，就開始準備 MFC 哩。 (譯註：MFC意指 Merged From CURRENT ，也就是把 -CURRENT 上的東西併入 -STABLE。
- 若該 patch 不需要 MFC 的話，Joe 就會關掉(close)該 PR 了。
- 若該 patch 需要 MFC 的話，Joe 會把 PR 狀態改為 "patched(已修正)"， 直到已經 MFC 完畢，才會

close(關掉)。

很多送出來的 PR 都很少附上問題的相關訊息，而有些則是相當複雜難搞，
或只是提到部分表面問題而已；
遇到這種情況時，是非常需要得到所有相關訊息以便解決問題。
若遇到這種無解的問題或再次發生的話，就必須要 re-open(重新開啟) 該 PR，以待解決。

PR 上所附的 "email address" 可能因某些原因而無法收信時，遇到這種狀況，通常就是 followup 該 PR ，並(在 followup 時)請回報者重新提供可正常收信的 email address。
當系統上的 mail 系統關閉或沒裝的時候，這通常是在使用 send-pr(1) 的替代方案。

3. 問題回報(PR)的狀態

若 PR 有任何變化的話，請務必記得更新 PR 的『狀態(state)』。 『狀態』應該要能正確反映該 PR 的目前進度才是。

例 1. 以下是更改 PR 狀態的小例子：

當有可以修正問題的 PR 出現，而相關負責的 developer(s) 也覺得這樣的修正可以接受，他們會 followup 該 PR，並將其狀態改為
"feedback"。同時，回報者應重新評估最終的修正結果，並回應：所回報的錯誤是否已成功修正。

每份 PR 通常會有下面這幾種狀態之一：

open
: PR 最初的狀態：這個問題被提出來，並在等待處理中。

analyzed
: 已經開始處理這問題，並且有找到疑似解決的方法。

feedback
: 需要回報者提供更詳細的相關資料，正如教學要因材施教，治病也要因人下藥，越多相關訊息，才能有最佳效果。

patched
: 已經送相關 patch 了，但仍因某些原因(MFC，或來自回報者的確認結果異常)因此尚未完畢。

suspended(暫緩)
: 因為沒附上相關訊息或參考資料，所以還沒辦法處理這問題。 This is a prime candidate for somebody who is looking for a project to take on. If the problem cannot be solved at all, it will be closed, rather than suspended. The documentation project uses suspended for wish-list items that entail a significant amount of work which no one currently has time for.

closed
: A problem report is closed when any changes have been integrated, documented, and tested, or when fixing the problem is abandoned.

The "patched" state is directly related to feedback, so you may go directly to "closed" state if the originator cannot test the patch, and it works in your own testing.

4. 問題回報(PR)的種類

While handling problem reports, either as a developer who has direct access to the GNATS database or as a contributor who browses the database and submits followups with patches, comments, suggestions or change requests, you will come across several different types of PRs.

- Unassigned PRs
- Assigned PRs
- 重複的 PR
- Stale PRs
- Misfiled PRs

The following sections describe what each different type of PRs is used for, when a PR belongs to one of these types, and what treatment each different type receives.

5. Unassigned PRs

When PRs arrive, they are initially assigned to a generic (placeholder) assignee. These are always prepended with freebsd-. The exact value for this default depends on the category; in most cases, it corresponds to a specific FreeBSD mailing list. Here is the current list, with the most common ones listed first:

表 1. Default Assignees - most common

Type	Categories	Default Assignee
base system	bin, conf, gnu, kern, misc	freebsd-bugs
architecture-specific	alpha, i386, ia64, powerpc, sparc64	freebsd-arch
ports collection	ports	freebsd-ports-bugs
documentation shipped with the system	docs	freebsd-doc
FreeBSD web pages (not including docs)	www	freebsd-www

表 2. Default Assignees - other

Type	Categories	Default Assignee
advocacy efforts	advocacy	freebsd-advocacy
Java Virtual Machine™ problems	java	freebsd-java
standards compliance	standards	freebsd-standards
threading libraries	threads	freebsd-threads
usb(4) subsystem	usb	freebsd-usb

Do not be surprised to find that the submitter of the PR has assigned it to the wrong category. If you fix the category, do not forget to fix the assignment as well. (In particular, our submitters seem to have a hard time understanding that just because their problem manifested on an i386 system, that it might be generic to all of FreeBSD, and thus be more appropriate for kern. The converse is also true, of course.)

Certain PRs may be reassigned away from these generic assignees by anyone. For assignees which are mailing lists, please use the long form when making the assignment (e.g., freebsd-foo instead of foo); this will avoid duplicate emails sent to the mailing list.

Here is a sample list of such entities; it is probably not complete. In some cases, entries that have the short form are aliases, not mailing lists.

表 3. Common Assignees - base system

Type	Suggested Category	Suggested Assignee
problem specific to the ARM® architecture	kern	freebsd-arm
problem specific to the MIPS® architecture	kern	freebsd-mips
problem specific to the PowerPC® architecture	kern	freebsd-ppc
problem with Advanced Configuration and Power Management (acpi(4))	kern	freebsd-acpi
problem with Asynchronous Transfer Mode (ATM) drivers	kern	freebsd-atm
problem with FireWire® drivers	kern	freebsd-firewire
problem with the filesystem code	kern	freebsd-fs
problem with the geom(4) subsystem	kern	freebsd-geom
problem with the ipfw(4) subsystem	kern	freebsd-ipfw
problem with Integrated Services Digital Network (ISDN) drivers	kern	freebsd-isdn
problem with Linux® or SVR4 emulation	kern	freebsd-emulation
problem with the networking stack	kern	freebsd-net
problem with PicoBSD	kern	freebsd-small
problem with the pf(4) subsystem	kern	freebsd-pf
problem with the scsi(4) subsystem	kern	freebsd-scsi
problem with the sound(4) subsystem	kern	freebsd-multimedia
problem with sysinstall(8)	bin	freebsd-qa
problem with the system startup scripts (rc(8))	kern	freebsd-rc

表 4. Common Assignees - Ports Collection

Type	Suggested Category	Suggested Assignee
problem with the ports framework (not with an individual port!)	ports	portmgr
port which is maintained by apache@FreeBSD.org	ports	apache
port which is maintained by eclipse@FreeBSD.org	ports	freebsd-eclipse
port which is maintained by gnome@FreeBSD.org	ports	gnome
port which is maintained by haskell@FreeBSD.org	ports	haskell

Type	Suggested Category	Suggested Assignee
port which is maintained by java@FreeBSD.org	ports	freebsd-java
port which is maintained by kde@FreeBSD.org	ports	kde
port which is maintained by openoffice@FreeBSD.org	ports	freebsd-openoffice
port which is maintained by perl@FreeBSD.org	ports	perl
port which is maintained by python@FreeBSD.org	ports	freebsd-python
port which is maintained by x11@FreeBSD.org	ports	freebsd-x11

Ports PRs which have a maintainer who is a ports committer may be reassigned by anyone (but note that not every FreeBSD committer is necessarily a ports committer, so you cannot simply go by the email address alone.)

For other PRs, please do not reassign them to individuals (other than yourself) unless you are certain that the assignee really wants to track the PR. This will help to avoid the case where no one looks at fixing a particular problem because everyone assumes that the assignee is already working on it.

6. Assigned PRs

If a PR has the responsible field set to the username of a FreeBSD developer, it means that the PR has been handed over to that particular person for further work.

Assigned PRs should not be touched by anyone but the assignee. If you have comments, submit a followup. If for some reason you think the PR should change state or be reassigned, send a message to the assignee. If the assignee does not respond within two weeks, unassign the PR and do as you please.

7. 重複的 PR

If you find more than one PR that describe the same issue, choose the one that contains the largest amount of useful information and close the others, stating clearly the number of the superseding PR. If several PRs contain non-overlapping useful information, submit all the missing information to one in a followup, including references to the others; then close the other PRs (which are now completely superseded).

8. Stale PRs

A PR is considered stale if it has not been modified in more than six months. Apply the following procedure to deal with stale PRs:

- If the PR contains sufficient detail, try to reproduce the problem in -CURRENT and -STABLE. If you succeed, submit a followup detailing your findings and try to find someone to assign it to. Set the state to "analyzed" if appropriate.
- If the PR describes an issue which you know is the result of a usage error (incorrect configuration or otherwise), submit a followup explaining what the originator did wrong, then close the PR with the reason "User error" or "Configuration error".
- If the PR describes an error which you know has been corrected in both -CURRENT and -STABLE, close it with a message stating when it was fixed in each branch.

- If the PR describes an error which you know has been corrected in -CURRENT, but not in -STABLE, try to find out when the person who corrected it is planning to MFC it, or try to find someone else (maybe yourself?) to do it. Set the state to "feedback" and assign it to whomever will do the MFC.
- In other cases, ask the originator to confirm if the problem still exists in newer versions. If the originator does not reply within a month, close the PR with the notation "Feedback timeout".

9. Misfiled PRs

GNATS is picky about the format of a submitted bug report. This is why a lot of PRs end up being "misfiled" if the submitter forgets to fill in a field or puts the wrong sort of data in some of the PR fields. This section aims to provide most of the necessary details for FreeBSD developers that can help them to close or refile these PRs.

When GNATS cannot deduce what to do with a problem report that reaches the database, it sets the responsible of the PR to gnats-admin and files it under the pending category. This is now a "misfiled" PR and will not appear in bug report listings, unless someone explicitly asks for a list of all the misfiled PRs. If you have access to the FreeBSD cluster machines, you can use query-pr to view a listing of PRs that have been misfiled:

```
% query-pr -x -q -r gnats-admin
52458 gnats-ad  open    serious  medium   Re: declaration clash f
52510 gnats-ad  open    serious  medium   Re: lots of sockets in
52557 gnats-ad  open    serious  medium
52570 gnats-ad  open    serious  medium   Jigdo maintainer update
```

Commonly PRs like the ones shown above are misfiled for one of the following reasons:

- A followup to an existing PR, sent through email, has the wrong format on its Subject: header.
- A submitter sent a Cc: to a mailing list and someone followed up to that post instead of the email issued by GNATS after processing. The email to the list will not have the category/PRnumber tracking tag. (This is why we discourage submitters from doing this exact thing.)
- When completing the send-pr(1) template, the submitter forgot to set the category or class of the PR to a proper value.
- When completing the send-pr(1) template, the submitter set Confidential to yes. (Since we allow anyone to mirror GNATS via CVSup, our PRs are public information. Security alerts should therefore not be sent via GNATS but instead via email to the Security Team.)
- It is not a real PR, but some random message sent to bug-followup@FreeBSD.org or freebsd-gnats-submit@FreeBSD.org.

10. Followups misfiled as new PRs

The first category of misfiled PRs, the one with the wrong subject header, is actually the one that requires the greatest amount of work from developers. These are not real PRs, describing separate problem reports. When a reply is received for an existing PR at one of the addresses that GNATS "listens" to for incoming messages, the subject of the reply should always be of the form:

```
Subject: Re: category/number: old synopsis text
```

Most mailers will add the “Re:” part when you reply to the original mail message of a PR. The “category/number:” part is a GNATS-specific convention that you have to manually insert to the

subject of your followup reports.

Any FreeBSD developer, who has direct access to the GNATS database, can periodically check for PRs of this sort and move interesting bits of the misfiled PR into the audit trail of the original PR (by posting a proper followup to a bug report to the address bug-followup@FreeBSD.org). Then the misfiled PR can be closed with a message similar to:

```
Your problem report was misfiled.  Please use the format
"Subject: category/number: original text" when following
up to older, existing PRs.  I've added the relevant bits
from the body of this PR to kern/12345
```

Searching with query-pr for the original PR, of which a misfiled followup is a reply, is as easy as running:

```
% query-pr -q -y "some text"
```

After you locate the original PR and the misfiled followups, use the -F option of query-pr to save the full text of all the relevant PRs in a UNIX® mailbox file, i.e.:

```
% query-pr -F 52458 52474 > mbox
```

Now you can use any mail user agent to view all the PRs you saved in mbox. Copy the text of all the misfiled PRs in a followup to the original PR and make sure you include the proper Subject: header. Then close the misfiled PRs. When you close the misfiled PRs remember that the submitter receives a mail notification that his PR changed state to "closed". Make sure you provide enough details in the log about the reason of this state change. Typically something like the following is ok:

```
Followup to ports/45364 misfiled as a new PR.
This was misfiled because the subject did not have the format:

  Re: ports/45364: ...
```

This way the submitter of the misfiled PR will know what to avoid the next time a followup to an existing PR is sent.

11. PRs misfiled because of missing fields

The second type of misfiled PRs is usually the result of a submitter forgetting to fill all the necessary fields when writing the original PR.

Missing or bogus "category" or "class" fields can result in a misfiled report. Developers can use edit-pr(1) to change the category or class of these misfiled PRs to a more appropriate value and save the PR.

Another common cause of misfiled PRs because of formatting issues is quoting, changes or removal of the send-pr template, either by the user who edits the template or by mailers which do strange things to plain text messages. This does not happen a lot of the time, but it can be fixed with edit-pr too; it does require a bit of work from the developer who refiles the PR, but it is relatively easy to do most of the time.

12. Misfiled PRs that are not really problem reports

Sometimes a user wants to submit a report for a problem and sends a simple email message to GNATS. The GNATS scripts will recognize bug reports that are formatted using the send-pr(1) template. They cannot parse any sort of email though. This is why submissions of bug reports that are sent to freebsd-gnats-submit@FreeBSD.org have to follow the template of send-pr, but email reports can be sent to FreeBSD problem reports 郵遞論壇.

Developers that come across PRs that look like they should have been posted to FreeBSD problem reports 郵遞論壇 or some other list should close the PR, informing the submitter in their state-change log why this is not really a PR and where the message should be posted.

The email addresses that GNATS listens to for incoming PRs have been published as part of the FreeBSD documentation, have been announced and listed on the web-site. This means that spammers found them. Spam messages that reach GNATS are promptly filed under the "pending" category until someone looks at them. Closing one of these with edit-pr(1) is very annoying though, because GNATS replies to the submitter and the sender’ s address of spam mail is never valid these days. Bounces will come back for each PR that is closed.

Currently, with the installation of some antispam filters that check all submissions to the GNATS database, the amount of spam that reaches the "pending" state is very small.

All developers who have access to the FreeBSD.org cluster machines are encouraged to check for misfiled PRs and immediately close those that are spam mail. Whenever you close one of these PRs, please do the following:

- Set Category to junk.
- Set Confidential to no.
- Set Responsible to yourself (and not, e.g., freebsd-bugs, which merely sends more mail).
- Set State to closed.

Junk PRs are not backed up, so filing spam mail under this category makes it obvious that we do not care to keep it around or waste disk space for it. If you merely close them without changing the category, they remain both in the master database and in any copies of the database mirrored through CVSup.

13. 延伸閱讀

下面這是在寫、處理 PR 時，可以參考的資料。當然很明顯，這份清單仍須補充。

- How to Write FreeBSD Problem Reports-給 PR 回報者用的參考原則。