

Posting Patches

-Kame

Kamezawa.hiroyu@jp.fujitsu.com



- I recommend you to subscribe the mailing-list.
 - ◆ Read it every day.
 - ◆ Of course, you don't have to read all traffics.

- There are several **good** articles
 - ◆ /Documentation/SubmittingPatches
 - ◆ /Documentation/CodingStyle
 - ◆ Andrew Morton's "The perfect patch"
<http://www.zip.com.au/~akpm/linux/patches/stuff/tpp.txt>
 - ◆ Jeff Garzik, "Linux kernel patch submission format."
<http://linux.yyz.us/patch-format.html>

Today's topic

- I'd like to talk about my experience in these 2.5 years
 - ◆ Bug/Test reports
 - ◆ Bug fix patch
 - ◆ How to work
 - ◆ Cleanup patch
 - ◆ Strategy and tactics
 - ◆ Start discussion from very small patch
 - ◆ Patch manners

.

When reporting bug (without patch...)

- Information that you should show
 1. Show kernel version.
 2. Summarize BUG.
 3. Attach your console log, if your kernel panics.(if you can)
 4. Attach your .config
 5. Describe what you was doing.
- Sometimes followings will be needed.
 1. Why do you consider it is bug ?
 2. What it should be (What do you want)
- Adding CC to Maintainers is good. If it's newly-released kernel's bug, replying to the kernel release e-mail is good

- Good start point for positing patch. It's always welcomed!
- Subject should be
“[BUG][PATCH] kernel-version brief description of bug”
- Adding CC to Maintainers
- Describe precisely as BUG report.
- Patch to the newest version is welcomed.
- If no response, maybe “CC” is wrong. Or pepoles are on vacation.....please **retry**
- If you post a patch, you may have responsibility for the Fix.
- BUG-Fix patch is tend to be accepted.

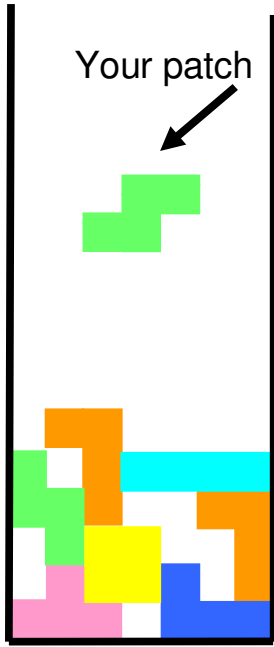
You can learn many things.

How to work

- Time difference
 - ◆ You may have receive responses in the midnight.
 - ◆ Sun, Mon, Tue, Wed, Thu, Fri, Sat
- How to wait
 - ◆ Maintainers receive many e-mails and have to read them.
 - ◆ When you intend to post patch against –mm/stock kernel, subscribing mm-commits mailing list is useful.
- Please be online for a while when your patch is merged.
- *KONJO* for retrying is one of the most important natures
- You can learn following things only by trying. So please try.
 - ◆ Make patches to be acceptable.
 - ◆ Don't hesitate to modify existing code, if necessary and reasonable

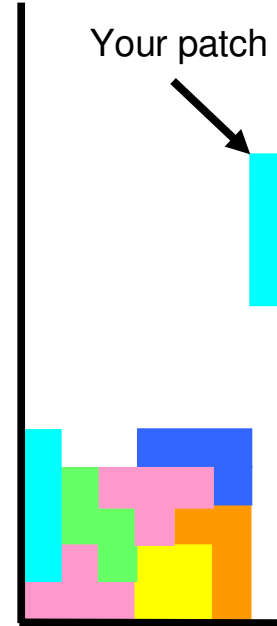
Strategy and Tactics

- You may have to write several patches before positing what you really want....



Your patch is not acceptable

Several cleanups



Your patch is acceptable

You maybe have to learn how to write a series of patches.

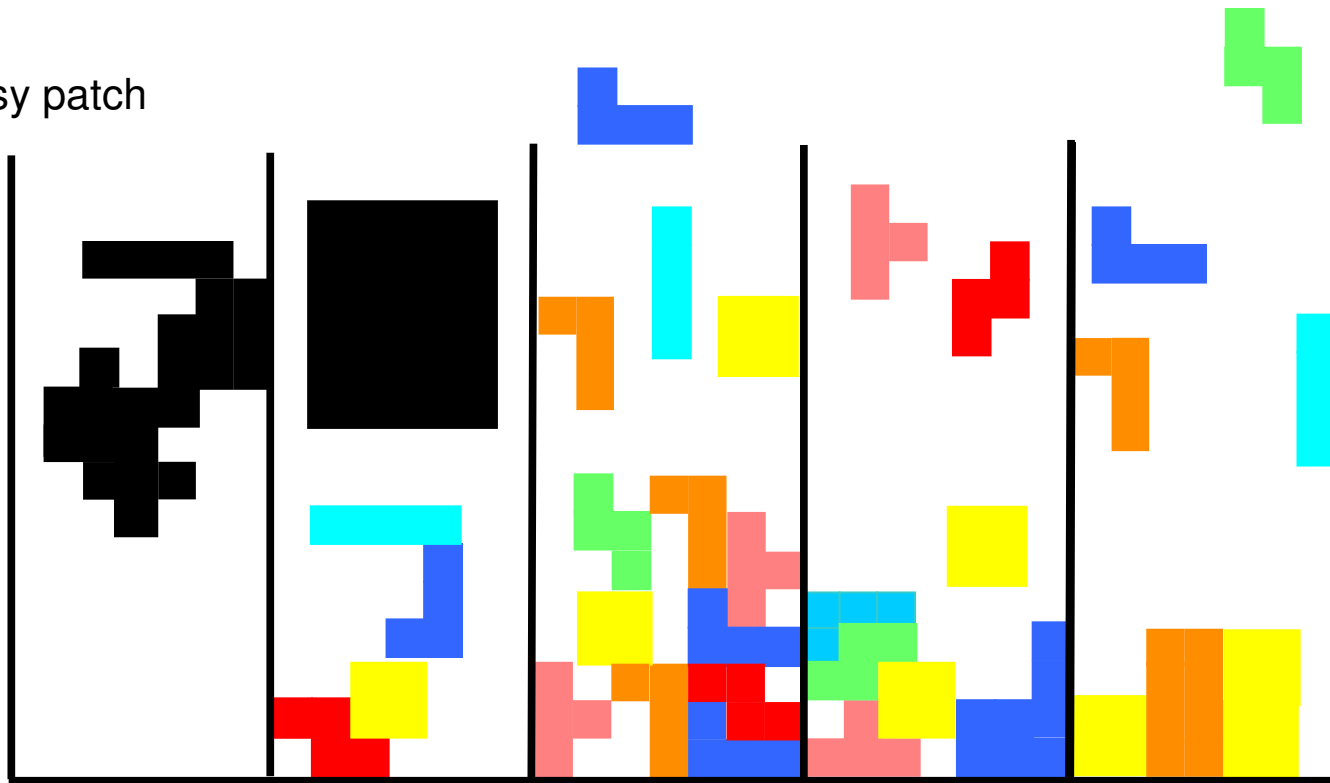
Neat or Die

no plan

good plan

Big messy patch

Clean patch set



Plan A

Plan B

Plan C

Plan D

Plan E

Cleanup patch

- One of the most important kinds of patch
- There are many cleanup-patch traffics in a day..
- Be careful not to change the logic **implicitly**.
- When you clean-up arch-specific-code, sometimes you will be asked to do clean up all other archs...even if you don't have it.
- To know **how-to-cleanup** is important. Because you have to write clean-up patch often before posting the patch which you really want.
- Showing your overview and purpose is sometimes good

Start discussion from very small patch

- Before writing very big and complex patch set, it's good to post your small patch (idea) to mailing list. And see response.
(if no patch, sometimes ignored....)
- When you have a chance to meet maintainers and kernel hackers, posting patches before meeting will be good.

[PATCH 01/10] version description take X : this patch name

- Please read “The Perfect Patch”
- Reflect comments AMAP when you retry
- All patches must have different name
- When you post a patch series
 - ◆ [0/X] Write the overview and description
 - ◆ [1/X][10/X] each patch's name should be different from each other
- Patch should be able to be applied independently to other patches.
- Add Change log
- Please don't omit your purpose in retry-patches.