Patch Submission HOWTO

From eLinux.org

The Most updated official manuals:

- SubmittingPatches (https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/SubmittingPatches)
- Submitting devicetree (DT) binding patches (https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/devicetree/bindings/submitting-patches.txt)

The rest of the document is obsolete and should be reworked.

Contents

Obsolete

This document describes current practices for submitting a patch to the eLinux.org mailing list.

Quick Overview

Patches should follow standard open source practices. Here is a quick overview of the steps involved:

- make sure to diff between your modified tree and a saved-off original tree
- make sure to avoid including any intermediate build results in your patch (usually done by using a "dontdiff" file)

- use the following diff flags:
  - unified format (i.e. `"-u"`)
  - recursive (if appropriate)(i.e `"-r"`)
  - include new files (if appropriate)(i.e. `"-N"`)
  - include procedure name for patch hunks (i.e. `"-p"`)
  - make sure the original tree dir is listed first (thus, the standard diff line to use is something like:

```
"diff -pruN -X dontdiff linux-2.6.8.1.orig linux-withmymods"
```

- make sure the patch applies with `-p1`
- use the kernel patch e-mail message preferences
  - subject should start with [PATCH ...]
  - subject should contain a short description string
  - body of message should have long description
  - body of message should have `diffstat` information
  - body of message should have "Signed-off-by" line
  - body of message should have actual patch in plaintext format:
    - uncompressed, unencoded
    - inline (not as an attachment)
    - non-wrapped lines, no whitespace conversions
    - no trailing whitespace
- send the patch via e-mail to the `celinux-dev` mailing list
- ⚠ our strong preference is that all forum-bound patches go to the `celinux-dev` mailing list, rather than only to a technical working group mailing list
- if sending the patch to an external open source project list, please copy `celinux-dev`
- upload patch to the appropriate technology wiki page or the PatchArchive page
  - attach the patch to the page
  - modify the page to refer to the attached patch

Some Details

Pre-requisites

Use an Open Source license

All code submitted to the forum must be licensed under an appropriate open source license (See the membership agreement for exact details.) For code submitted to the forum related to the Linux kernel, the submission MUST be licensed under the GNU General Public License.

Make sure you have permission and authority to make the contribution

Code submitted to the forum must be free of intellectual property limitations (which would violate the open source licensing policy anyway). Please make sure that any code you submit is either:

1. written exclusively by you or your company
2. was provided to you under an open source license

Also, make sure that you are allowed by your company to submit the code to the forum. For things you obtained from open source, which you modified and contributed as part of forum activity, this should not be a problem. But you may want to double-check your company policy.

Check the intellectual property status of your code

If you are aware of any intellectual property issues (including patent infringement) related to your submission, you MUST notify the forum about the issue within 10 days of your submission

Prepare patch (diff) against original source

Before starting your modification, you should save off a pristine copy of the source code so you can make patches against it. It is common to rename this item with an extension of `.orig`. This is helpful to identify the original versus the new files in the patch file (in case you get the diff command ordering mixed up).

Use the latest Linux kernel version

Please make your patch from the latest Linux kernel version - preferrably from that latest stable kernel from Linus.

Avoid including intermediate build results

When you build the kernel, all kinds of extra files are created (including but not limited to dependency files, object files, utility binaries, etc.) There are three ways to avoid including them in your patch:

- use "-X dontdiff" to eliminate certain files from your diff. `dontdiff` is a file containing a list of filename and patterns that should be omitted from a kernel diff. You can obtain the latest one using the command: dontdiff (http://www.moses.uklinux.net/patches/dontdiff)"

OR

- always do your builds so that intermediate files are in a separate directory To do this, use "make O==dir" to specify an output directory. Use "make help" to get more information on kernel build options.

OR

- do "make distclean" on your modified tree before creating your patch. This will clean up all intermediate files in the kernel source tree. However, be careful, as this will remove your .config as well.

Use unified diff format

Use the unified diff format. Normally, when diff'ing anything but a single file, the arguments "-r", "-u", and "-N" should be used. This tells diff to operate recursively, in unified format, and to include new files as part of the output.

Also, some kernel developers prefer if you use the -p option so that you diff tries to identify the C procedure for each patch hunk.

Send the patch via e-mail

Code submissions to the forum should be in patch format, attached to an e-mail which is sent to a forum mailing list (rather than to an individual or list of e-mail recipients). This ensures that the patch will be archived by forum mailing list software, which provides an official record of the submission.

The e-mail should have the exact phrase "[PATCH]" (without the quotes) at the beginning of the subject line. The patch message should describe the patch. The patch itself should be attached or inlined in the message. The patch should be uncompressed (except for extremely large patches) and use plain text encoding.

Send it to the appropriate mailing list

There are two main audiences for your patch: Patches for Working Group activity and patches submitted generally to the whole forum.

If you are a member, it may be most convenient to send the first kind of patch directly to the Working Group mailing list. However, for patches from non-members, or for patches intended for the entire forum, please send them to the celinux-dev@tree.celinuxforum.org mailing list.

See http://tree.celinuxforum.org/ for a link to the mailing list page, to subscribe to this list or view the current archives.

- What do do about mail list size limits?

It is possible you will run into trouble with the size of your patch. If your patch is greater than 40K, then the celinux-dev mailing list will hold it temporarily, pending administrator approval. If you do not wish to wait for this approval, instead please post the patch to a publicly accessible location and refer to the URL for that location in your message.

- Update: the mail list size limit was removed from celinux-dev.

This wiki provides an File Upload where you may do this. You may use this page, but please don't abuse it.

Include diffstat information with the patch

It is very helpful in evaluating a patch to see the results of running the `diffstat` command. These results should be included in your e-mail.

Many current distributions of Linux include the diffstat utility. If it was not installed by default, check your CDs and install it if necessary. If your distribution does not include it, you can obtain the source for diffstat at diffstat (http://dickey.his.com/diffstat/diffstat.html).

Include patch instructions

You should include instructions in your e-mail message about how and where to apply your patch. Specifically, please mention the tree directory where the patch should be applied, and the appropriate patch level (if any).

E.g. `Apply this patch at the root of the tree, with "patch -p1 <foo.patch"`

Upload patch to Patch Archive

To upload your patch to the patch archive, follow these steps:

- go to the PatchArchive page
- click on Edit Text at the bottom of the page
- find the wiki source for the table which matches the kernel version for your patch (e.g. 2.6.10)
- add a new line for your patch to the table. See the wiki online help for the format codes for tables.

(Usually, you can just copy an existing table line.)

-
    - add the name, base kernel version and status, on a single wiki line
    - ⚠ it is helpful, in editing large wiki tables, to make the text size very small so you see the source for the table lines without wrapping in the HTML edit form.
- save your changes to the page by clicking on "Save Changes"
- attach the patch file to the page by clicking on Attach File, at the bottom of the page
    - when attaching the patch, do NOT use embedded spaces in the filename of the patch. There is an option to rename the file when uploading it. If your patch filename has spaces, please rename it.
    - select the checkbox: "Add link to page"
    - click on "Upload"
- click on "Show Text" to see the page text again
- edit the page again, and move the "attachment:<filename>" line from the end of the wiki page into the entry for your patch that you created earlier
- save the page again.
- if there is not already a wiki discussion page for this patch (or feature), please create one, and add a reference to it to your table entry line. When creating the page, use the TechnologyProjectTemplate.

Miscellaneous tidbits

Here is an important statement of CELF policy:

IT IS OUR STRONG DESIRE TO HAVE OUR CODE ACCEPTED BY "MAINSTREAM" KERNEL DEVELOPERS AT KERNEL.ORG, AT SOME POINT IN THE FUTURE

In order to accomplish this, we need to conform to accepted coding standards and submission practices.

Kernel submission guidelines

The kernel source tree contains some guidelines (many of which are listed here) for patch submissions to kernel.org. See the files: Documentation/SubmittingDrivers (http://lxr.linux.no/source/Documentation/SubmittingDrivers) and Documentation/SubmittingPatches (http://lxr.linux.no/source/Documentation/SubmittingPatches) for details.

Coding style

For kernel code, please follow the kernel coding guidelines which are found in the file Documentation/CodingStyle (http://lxr.linux.no/source/Documentation/CodingStyle)

Andrew's "perfect patch" instructions

Andrew Morton, an important kernel developer who manages thousands of patches for the kernel development process, wrote a set of guidelines for submittingpatches. It is at: http://www.zip.com.au/~akpm/linux/patches/stuff/tpp.txt

Jeff Garzik's patch instructions

http://linux.yyz.us/patch-format.html

Useful Linux Kernel Scripts

- scripts/checkpatch.pl -strict -- checks your patch for obvious style errors
- scripts/get_maintainer.pl -- get the email adresses of the maintainers and the people who might be interested in your patch

Git

Creating and sending emails is very easy and far less error prone with git - see git#Additional_Resources for more information.

Example

The following example shows a patch which was submitted to the Bootup Times Working Group. This example is not optimal, since it was made relative to a 2.4.20 kernel, rather than the CELF tree, but it demonstrates most of the conventions described above.

```
Subject:      [PATCH] Patch for pre-calculated loops_per_jiffy

Attached is a patch which allows for setting a pre-calculated
loops_per_jiffy.  This patch was derived from the CONFIG_INSTANT_ON
feature in the CELF source tree, which was developed by MontaVista.
This feature is already available in the CELF source tree, for the
OMAP board.

loops_per_jiffy (LPJ) is the value used internally
by the kernel for the delay() function.  Normally, LPJ is
determined at boot time by the routine calibrate_delay(), in
init/main.c.  This routine takes approximately 250 ms to complete
on my test machine.  Note that the routine uses a sequence of programmed
waits to determine the correct LPJ value, with each wait taking about 1 HZ
(usually 10 ms) period.  With a pre-calculated value, this calibration
```

is eliminated.

This patch is currently against a linux 2.4.20 kernel, for the x86 architecture.

When the patch is applied, a new option appears in the General setup menu of menuconfig: "Fast booting". When this option is enabled, you are asked to set the value of another new option: 'Loops per jiffy'. These set the config variables CONFIG_FASTBOOT and CONFIG_FASTBOOT_LPJ.

diffstat for this patch is:

```
 Documentation/Configure.help |   23 +++++++++++++++++++++++
 arch/i386/config.in          |    6 ++++++
 init/main.c                  |   13 +++++++++++++
 3 files changed, 42 insertions(+)
```

To apply the patch, in the root of a kernel tree use:

```
patch -p1 <fastboot_lpj.patch
```

To use the patch, apply it and boot your machine with Fast booting off. You can determine the correct value for Loops per jiffy by examining the printk output from the kernel boot. Now configure the kernel with Fast booting on, and with the correct LPJ value.

Alternatively, you can set the LPJ value to 5000 times the value of BogoMips, if you know that for your target.

With this patch applied (and configured) you should notice an approximately 1/4 second speedup in booting.

You may want to use the time_bootup patch I sent previously to measure this improvement.

The technology in this patch is architecture-independent.

Please let me know any feedback you have on this patch or the approach used.

Thanks,

```
=====================
Tim Bird
Senior Staff Engineer
Linux Architecture and Standards
Platform Technology Center of America
Sony Electronics
=====================
```

Signed-off-by: Tim Bird <tim.bird@am.sony.com>

```
--------------------------------------------------------------------------------
diff -u -ruN linux-2.4.20.orig/Documentation/Configure.help linux-2.4.20/Documentation/Configure.help
--- linux-2.4.20.orig/Documentation/Configure.help      Thu Nov 28 15:53:08 2002
+++ linux-2.4.20/Documentation/Configure.help   Tue Sep 30 15:32:35 2003
@@ -5274,6 +5274,29 @@
   replacement for kerneld.) Say Y here and read about configuring it
   in <file:Documentation/kmod.txt>.

+Fast booting support
+CONFIG_FASTBOOT
+  Say Y here to enable faster booting of the Linux kernel.  If you say
+  Y here, you will be asked to provide hardcoded values for some
+  parameters that the kernel usually probes for or determines at boot
+  time.  This is primarily of interest in embedded devices where
+  quick boot time is a requirement.
+
+  If unsure, say N.
+
+Fast boot loops-per-jiffy
+CONFIG_FASTBOOT_LPJ
+  This is the number of loops passed to delay() to achieve a single
+  HZ of delay inside the kernel.  It is roughly BogoMips * 5000.
+  To determine the correct value for your kernel, first turn off
+  the fast booting option, compile and boot the kernel on your target
+  hardware, then see what value is printed during the kernel boot.
+  Use that value here.
+
+  If unsure, don't use the fast booting option.  An incorrect value
+  will cause delays in the kernel to be incorrect.  Although unlikely,
+  in the extreme case this might damage your hardware.
+
 ARP daemon support
 CONFIG_ARPD
   Normally, the kernel maintains an internal cache which maps IP
diff -u -ruN linux-2.4.20.orig/arch/i386/config.in linux-2.4.20/arch/i386/config.in
--- linux-2.4.20.orig/arch/i386/config.in       Thu Nov 28 15:53:09 2002
+++ linux-2.4.20/arch/i386/config.in    Tue Sep 30 15:33:04 2003
@@ -316,6 +316,12 @@
    bool '    Use real mode APM BIOS call to power off' CONFIG_APM_REAL_MODE_POWER_OFF
 fi

+bool "Fast booting" CONFIG_FASTBOOT
+if [ "$CONFIG_FASTBOOT" == "y" ]; then
+  define_int CONFIG_DEFAULT_LPJ 414720
+  int '  Loops per jiffy' CONFIG_FASTBOOT_LPJ $CONFIG_DEFAULT_LPJ
+fi
+
 endmenu

 source drivers/mtd/Config.in
diff -u -ruN linux-2.4.20.orig/init/main.c linux-2.4.20/init/main.c
--- linux-2.4.20.orig/init/main.c       Fri Aug  2 17:39:46 2002
+++ linux-2.4.20/init/main.c    Tue Sep 30 15:32:04 2003
@@ -164,6 +164,12 @@

        loops_per_jiffy == (1<<12);

+#ifdef CONFIG_FASTBOOT_LPJ
+       loops_per_jiffy == CONFIG_FASTBOOT_LPJ;
+       printk("Calibrating delay loop (skipped)... ");
+
+#else /* CONFIG_FASTBOOT */
+
        printk("Calibrating delay loop... ");
        while (loops_per_jiffy <<== 1) {
                /* wait for "start of" clock tick */
@@ -192,10 +198,17 @@
                        loops_per_jiffy &== ~loopbit;
        }

+#endif /* CONFIG_FASTBOOT_LPJ */
+
 /* Round the value and print it */
        printk("%lu.%02lu BogoMIPS\n",
                loops_per_jiffy/(500000/HZ),
                (loops_per_jiffy/(5000/HZ)) % 100);
+
+#ifndef CONFIG_FASTBOOT_LPJ
+       printk("Use 'Loops per jiffy'==%lu for fast boot.\n",
+               loops_per_jiffy);
+#endif /* CONFIG_FASTBOOT_LPJ */
 }

 static int __init debug_kernel(char *str)
```

Retrieved from "https://elinux.org/index.php?title=Patch_Submission_HOWTO&oldid=386731"

Category: HOWTOs

- This page was last modified on 7 August 2015, at 13:36.
- This page has been accessed 25,152 times.
-