Linuxized ACPICA - Introduction to ACPICA Release Automation

Copyright: © 2013-2016, Intel Corporation
Author: Lv Zheng <lv.zheng@intel.com>

Abstract

This document describes the ACPICA project and the relationship between ACPICA and Linux. It also describes how ACPICA code in drivers/acpi/acpica, include/acpi and tools/power/acpi is automatically updated to follow the upstream.

ACPICA Project

The ACPI Component Architecture (ACPICA) project provides an operating system (OS)-independent reference implementation of the Advanced Configuration and Power Interface Specification (ACPI). It has been adapted by various host OSes. By directly integrating ACPICA, Linux can also benefit from the application experiences of ACPICA from other host OSes.

The homepage of ACPICA project is: www.acpica.org, it is maintained and supported by Intel Corporation.

The following figure depicts the Linux ACPI subsystem where the ACPICA adaptation is included:

```
+---------------------------------------------------------+
|                                                         |
|   +---------------------------------------------------+ |
|   | +------------------+                              | |
|   | | Table Management |                              | |
|   | +------------------+                              | |
|   | +----------------------+                          | |
|   | | Namespace Management |                          | |
|   | +----------------------+                          | |
|   | +------------------+       ACPICA Components      | |
|   | | Event Management |                              | |
|   | +------------------+                              | |
|   | +---------------------+                           | |
|   | | Resource Management |                           | |
|   | +---------------------+                           | |
|   | +---------------------+                           | |
|   | | Hardware Management |                           | |
|   | +---------------------+                           | |
| +---------------------------------------------------+ | |
| | |                            +------------------+ | | |
| | |                            | OS Service Layer | | | |
| | |                            +------------------+ | | |
| | +-------------------------------------------------|-+ |
| |   +--------------------+                          |   |
| |   | Device Enumeration |                          |   |
| |   +--------------------+                          |   |
| |   +------------------+                            |   |
| |   | Power Management |                            |   |
| |   +------------------+     Linux/ACPI Components  |   |
| |   +--------------------+                          |   |
| |   | Thermal Management |                          |   |
| |   +--------------------+                          |   |
| |   +--------------------------+                    |   |
| |   | Drivers for ACPI Devices |                    |   |
| |   +--------------------------+                    |   |
| |   +--------+                                      |   |
| |   | ...... |                                      |   |
| |   +--------+                                      |   |
| +---------------------------------------------------+   |
|                                                         |
+---------------------------------------------------------+

          Figure 1. Linux ACPI Software Components
```

Note

A. OS Service Layer - Provided by Linux to offer OS dependent implementation of the predefined ACPICA interfaces (acpi_os_*).

```
include/acpi/acpiosxf.h
drivers/acpi/osl.c
include/acpi/platform
```

```
include/asm/acenv.h
```

B. ACPICA Functionality - Released from ACPICA code base to offer OS independent implementation of the ACPICA interfaces (acpi_*).

```
drivers/acpi/acpica
include/acpi/ac*.h
tools/power/acpi
```

C. Linux/ACPI Functionality - Providing Linux specific ACPI functionality to the other Linux kernel subsystems and user space programs.

```
drivers/acpi
include/linux/acpi.h
include/linux/acpi*.h
include/acpi
tools/power/acpi
```

D. Architecture Specific ACPICA/ACPI Functionalities - Provided by the ACPI subsystem to offer architecture specific implementation of the ACPI interfaces. They are Linux specific components and are out of the scope of this document.

```
include/asm/acpi.h
include/asm/acpi*.h
arch/*/acpi
```

ACPICA Release

The ACPICA project maintains its code base at the following repository URL: https://github.com/acpica/acpica.git. As a rule, a release is made every month.

As the coding style adopted by the ACPICA project is not acceptable by Linux, there is a release process to convert the ACPICA git commits into Linux patches. The patches generated by this process are referred to as "linuxized ACPICA patches". The release process is carried out on a local copy the ACPICA git repository. Each commit in the monthly release is converted into a linuxized ACPICA patch. Together, they form the monthly ACPICA release patchset for the Linux ACPI community. This process is illustrated in the following figure:

```
+-----------------------------+
| acpica / master (-) commits |
+-----------------------------+
   /|\         |
    |         \|/
    |  /---------------------\    +----------------------+
    | < Linuxize repo Utility >-->| old linuxized acpica |--+
    |  \---------------------/    +----------------------+  |
    |                                                       |
 /---------\                                                |
< git reset >                                                \
 \---------/                                                  \
   /|\                                                        /+-+
    |                                                        /   |
+-----------------------------+                             |    |
| acpica / master (+) commits |                             |    |
+-----------------------------+                             |    |
               |                                            |    |
              \|/                                           |    |
     /-----------------------\    +----------------------+  |    |
    < Linuxize repo Utilities >-->| new linuxized acpica |--+    |
     \-----------------------/    +----------------------+       |
                                                                \|/
+--------------------------+                  /----------------------\
| Linuxized ACPICA Patches |<----------------< Linuxize patch Utility >
+--------------------------+                  \----------------------/
               |
              \|/
 /---------------------------\
< Linux ACPI Community Review >
 \---------------------------/
               |
              \|/
+------------------------+    /------------------\    +----------------+
| linux-pm / linux-next |-->< Linux Merge Window >-->| linux / master |
+------------------------+    \------------------/    +----------------+
```

```
            Figure 2. ACPICA -> Linux Upstream Process
```

Note

A. Linuxize Utilities - Provided by the ACPICA repository, including a utility located in source/tools/acpisrc folder and a number of scripts located in generate/linux folder.
B. acpica / master - "master" branch of the git repository at <https://github.com/acpica/acpica.git>.
C. linux-pm / linux-next - "linux-next" branch of the git repository at <https://git.kernel.org/pub/scm/linux/kernel/git/rafael/linux-pm.git>.
D. linux / master - "master" branch of the git repository at <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git>.

Before the linuxized ACPICA patches are sent to the Linux ACPI community for review, there is a quality assurance build test process to reduce porting issues. Currently this build process only takes care of the following kernel configuration options: CONFIG_ACPI/CONFIG_ACPI_DEBUG/CONFIG_ACPI_DEBUGGER

ACPICA Divergences

Ideally, all of the ACPICA commits should be converted into Linux patches automatically without manual modifications, the "linux / master" tree should contain the ACPICA code that exactly corresponds to the ACPICA code contained in "new linuxized acpica" tree and it should be possible to run the release process fully automatically.

As a matter of fact, however, there are source code differences between the ACPICA code in Linux and the upstream ACPICA code, referred to as "ACPICA Divergences".

The various sources of ACPICA divergences include:

1. Legacy divergences - Before the current ACPICA release process was established, there already had been divergences between Linux and ACPICA. Over the past several years those divergences have been greatly reduced, but there still are several ones and it takes time to figure out the underlying reasons for their existence.
2. Manual modifications - Any manual modification (eg. coding style fixes) made directly in the Linux sources obviously hurts the ACPICA release automation. Thus it is recommended to fix such issues in the ACPICA upstream source code and generate the linuxized fix using the ACPICA release utilities (please refer to Section 4 below for the details).
3. Linux specific features - Sometimes it's impossible to use the current ACPICA APIs to implement features required by the Linux kernel, so Linux developers occasionally have to change ACPICA code directly. Those changes may not be acceptable by ACPICA upstream and in such cases they are left as committed ACPICA divergences unless the ACPICA side can implement new mechanisms as replacements for them.
4. ACPICA release fixups - ACPICA only tests commits using a set of the user space simulation utilities, thus the linuxized ACPICA patches may break the Linux kernel, leaving us build/boot failures. In order to avoid breaking Linux bisection, fixes are applied directly to the linuxized ACPICA patches during the release process. When the release fixups are backported to the upstream ACPICA sources, they must follow the upstream ACPICA rules and so further modifications may appear. That may result in the appearance of new divergences.
5. Fast tracking of ACPICA commits - Some ACPICA commits are regression fixes or stable-candidate material, so they are applied in advance with respect to the ACPICA release process. If such commits are reverted or rebased on the ACPICA side in order to offer better solutions, new ACPICA divergences are generated.

ACPICA Development

This paragraph guides Linux developers to use the ACPICA upstream release utilities to obtain Linux patches corresponding to upstream ACPICA commits before they become available from the ACPICA release process.

1. Cherry-pick an ACPICA commit

First you need to git clone the ACPICA repository and the ACPICA change you want to cherry pick must be committed into the local repository.

Then the gen-patch.sh command can help to cherry-pick an ACPICA commit from the ACPICA local repository:

```
$ git clone https://github.com/acpica/acpica
$ cd acpica
$ generate/linux/gen-patch.sh -u [commit ID]
```

Here the commit ID is the ACPICA local repository commit ID you want to cherry pick. It can be omitted if the commit is "HEAD".

2. Cherry-pick recent ACPICA commits

Sometimes you need to rebase your code on top of the most recent ACPICA changes that haven't been applied to Linux yet.

You can generate the ACPICA release series yourself and rebase your code on top of the generated ACPICA release patches:

```
$ git clone https://github.com/acpica/acpica
$ cd acpica
$ generate/linux/make-patches.sh -u [commit ID]
```

The commit ID should be the last ACPICA commit accepted by Linux. Usually, it is the commit modifying ACPI_CA_VERSION. It can be found by executing "git blame source/include/acpixf.h" and referencing the line that contains "ACPI_CA_VERSION".

3. Inspect the current divergences

If you have local copies of both Linux and upstream ACPICA, you can generate a diff file indicating the state of the current divergences:

```
# git clone https://github.com/acpica/acpica
# git clone https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
# cd acpica
# generate/linux/divergence.sh -s ../linux
```