Upgrading ACPI tables via initrd

What is this about

If the ACPI_TABLE_UPGRADE compile option is true, it is possible to upgrade the ACPI execution environment that is defined by the ACPI tables via upgrading the ACPI tables provided by the BIOS with an instrumented, modified, more recent version one, or installing brand new ACPI tables.

When building initrd with kernel in a single image, option ACPI_TABLE_OVERRIDE_VIA_BUILTIN_INITRD should also be true for this feature to work.

For a full list of ACPI tables that can be upgraded/installed, take a look at the char **table_sigs[MAX_ACPI_SIGNATURE];* definition in drivers/acpi/tables.c.

All ACPI tables iasl (Intel's ACPI compiler and disassembler) knows should be overridable, except:

- ACPI_SIG_RSDP (has a signature of 6 bytes)
- ACPI_SIG_FACS (does not have an ordinary ACPI table header)

Both could get implemented as well.

What is this for

Complain to your platform/BIOS vendor if you find a bug which is so severe that a workaround is not accepted in the Linux kernel. And this facility allows you to upgrade the buggy tables before your platform/BIOS vendor releases an upgraded BIOS binary.

This facility can be used by platform/BIOS vendors to provide a Linux compatible environment without modifying the underlying platform firmware.

This facility also provides a powerful feature to easily debug and test ACPI BIOS table compatibility with the Linux kernel by modifying old platform provided ACPI tables or inserting new ACPI tables.

It can and should be enabled in any kernel because there is no functional change with not instrumented initrds.

How does it work

```
# Extract the machine's ACPI tables:
cd /tmp
acpidump >acpidump
acpixtract -a acpidump
# Disassemble, modify and recompile them:
iasl -d *.dat
# For example add this statement into a _PRT (PCI Routing Table) function
# of the DSDT:
Store("HELLO WORLD", debug)
# And increase the OEM Revision. For example, before modification:
DefinitionBlock ("DSDT.aml", "DSDT", 2, "INTEL ", "TEMPLATE", 0x00000000)
# After modification:
DefinitionBlock ("DSDT.aml", "DSDT", 2, "INTEL ", "TEMPLATE", 0x00000001)
iasl -sa dsdt.dsl
# Add the raw ACPI tables to an uncompressed cpio archive.
# They must be put into a /kernel/firmware/acpi directory inside the cpio
# archive. Note that if the table put here matches a platform table
# (similar Table Signature, and similar OEMID, and similar OEM Table ID)
# with a more recent OEM Revision, the platform table will be upgraded by
# this table. If the table put here doesn't match a platform table
# (dissimilar Table Signature, or dissimilar OEMID, or dissimilar OEM Table
# ID), this table will be appended.
mkdir -p kernel/firmware/acpi
cp dsdt.aml kernel/firmware/acpi
# A maximum of "NR_ACPI_INITRD_TABLES (64)" tables are currently allowed
# (see osl.c):
iasl -sa facp.dsl
iasl -sa ssdt1.dsl
cp facp.aml kernel/firmware/acpi
cp ssdt1.aml kernel/firmware/acpi
# The uncompressed cpio archive must be the first. Other, typically
# compressed cpio archives, must be concatenated on top of the uncompressed
# one. Following command creates the uncompressed cpio archive and
# concatenates the original initrd on top:
find kernel | cpio -H newc --create > /boot/instrumented_initrd
cat /boot/initrd >>/boot/instrumented_initrd
# reboot with increased acpi debug level, e.g. boot params:
acpi.debug_level=0x2 acpi.debug_layer=0xFFFFFFFF
```

```
# and check your syslog:
[    1.268089] ACPI: PCI Interrupt Routing Table [\_SB_.PCI0._PRT]
[    1.272091] [ACPI Debug]  String [0x0B] "HELLO WORLD"
```

iasl is able to disassemble and recompile quite a lot different, also static ACPI tables.

Where to retrieve userspace tools

iasl and acpixtract are part of Intel's ACPICA project: https://acpica.org/

and should be packaged by distributions (for example in the acpica package on SUSE).

acpidump can be found in Len Browns pmtools: ftp://kernel.org/pub/linux/kernel/people/lenb/acpi/utils/pmtools/acpidump

This tool is also part of the acpica package on SUSE. Alternatively, used ACPI tables can be retrieved via sysfs in latest kernels: /sys/firmware/acpi/tables