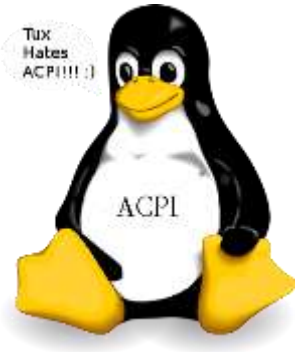
Agenda

1. Platform configuration
2. Device tree
3. **ACPI**

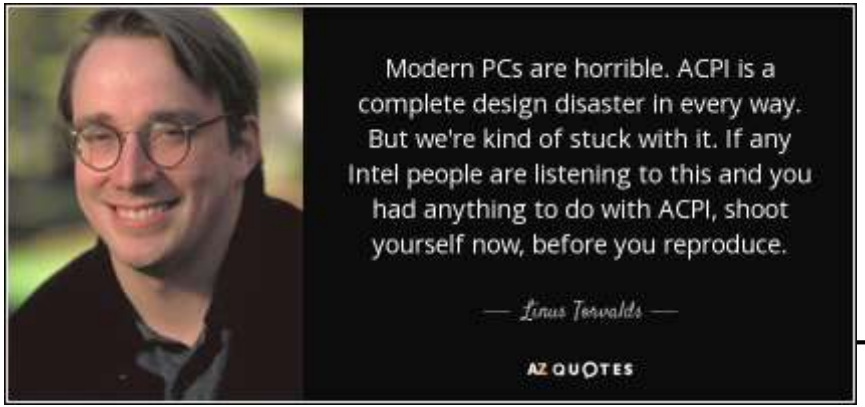
Module configuration

- module_param()
- Platform configuration
- Device tree
- ACPI

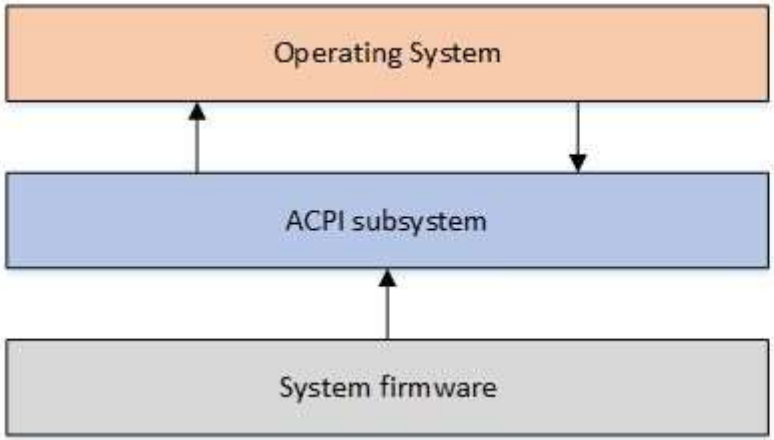


Before ACPI

1. the behavior of OS applications could be negatively affected by the BIOS configured power management settings, causing systems to go to sleep during presentations or other inconvenient times.
2. the power management interface was proprietary on each system. This required developers to learn how to configure power management for each individual system.
3. the default settings for various devices could also conflict with each other, causing devices to crash, behave erratically, or become undiscoverable.



What is ACPI?



The *ACPI subsystem* is an interface layer between the *System firmware* and the *OS*. The arrows indicate data flow.

ACPI Terminology

- ACPI (Advanced Configuration and Power Interface) - an advanced configuration and power management interface, its task is to provide interaction between the operating system, hardware and motherboard BIOS.
- Especially needed in laptops, without ACPI, for example, the OS can not determine the battery charge, the device reacts incorrectly to pressing the "Power" button and is quickly discharged.

ACPI on you PC

- dmesg|grep acpi

```
[    0.000000] ACPI: LAPIC_NMI (acpi_id[0xff] high edge lint[0x1])
[    0.238086] acpiphp: ACPI Hot Plug PCI Controller Driver version: 0.5
[    0.273222] acpi PNP0A08:00: _OSC: OS supports [ExtendedConfig ASPM ClockPM Segments MSI]
[    0.273358] acpi PNP0A08:00: _OSC: platform does not support [PCIeHotplug]
[    0.273477] acpi PNP0A08:00: _OSC: OS now controls [PME AER PCIeCapability]
[    0.273501] acpi PNP0A08:00: [Firmware Info]: MMCONFIG for domain 0000 [bus 00-3f] only partially covers this bridge
[    0.346619] clocksource: acpi_pm: mask: 0xffffff max_cycles: 0xffffff, max_idle_ns: 2085701024 ns
[    1.029471] acpi_cpufreq: overriding BIOS provided _PSD data
[    1.105739] acpi device:56: registered as cooling_device4
```

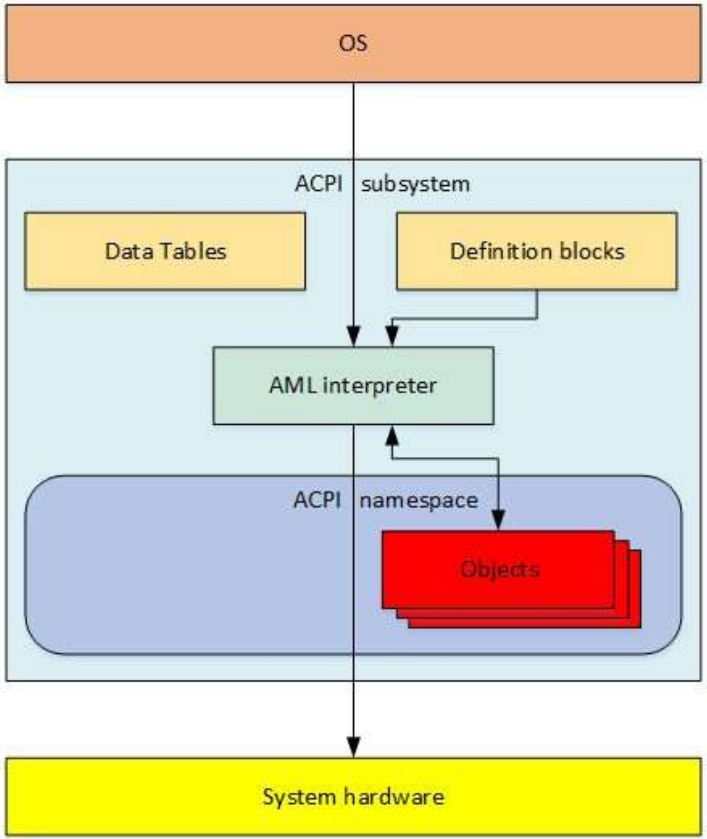
To set the action when closing the lid

- /etc/acpi/actions/lm_lid.sh

To set the action when the power button is pressed

- /etc/acpi/powerbtn.sh

ACPI data types and structure



The ACPI subsystem consists of two types of data structures: *data tables* and *definition blocks*.

Upon initialization, the *AML interpreter* extracts the byte code in the definition blocks as enumerable objects.

This collection of enumerable objects forms the OS construct called the *ACPI namespace*.

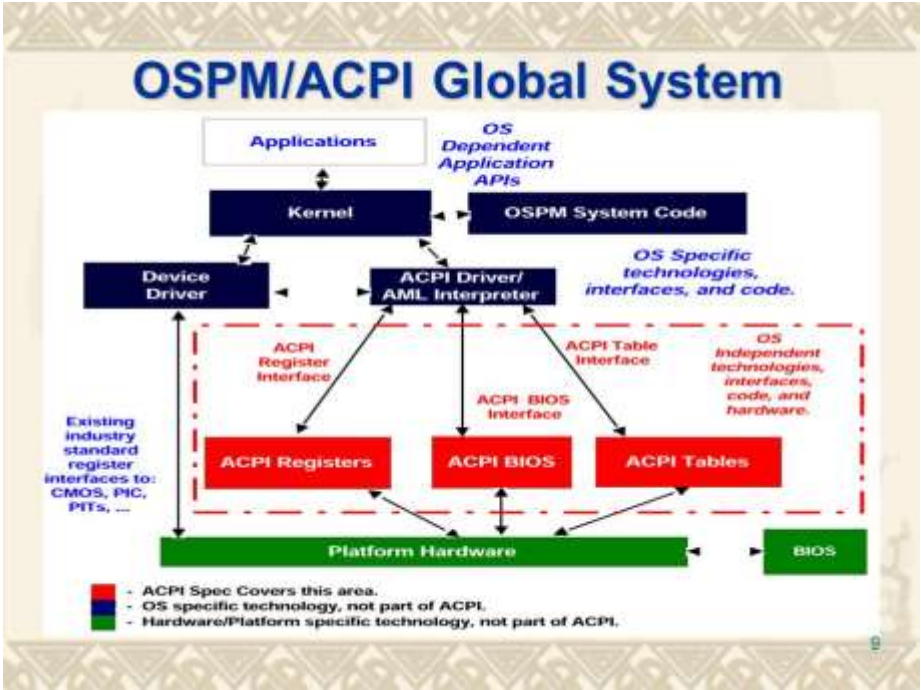
Objects can either have a directly defined value or must be evaluated and interpreted by the AML interpreter.

The AML interpreter, directed by the *OS*, evaluates objects and interfaces with *system hardware* to perform necessary operations.

1. **data tables**
2. **definition blocks**

AML?

- ACPI **Machine Language (AML)** is the platform independent code that ACPI utilizes. A knowledge of it is required to even shutdown the computer. It is found in the DSDT and SSDT tables, which are in turn found by parsing the RSDT or XSDT.
- AML code is **byte code** which is parsed from the beginning of each table when that table is read. It contains definitions of devices and objects within the ACPI namespace.
- See https://wiki.osdev.org/AML



ASL 2.0 Introduction and Overview

- Legacy ASL

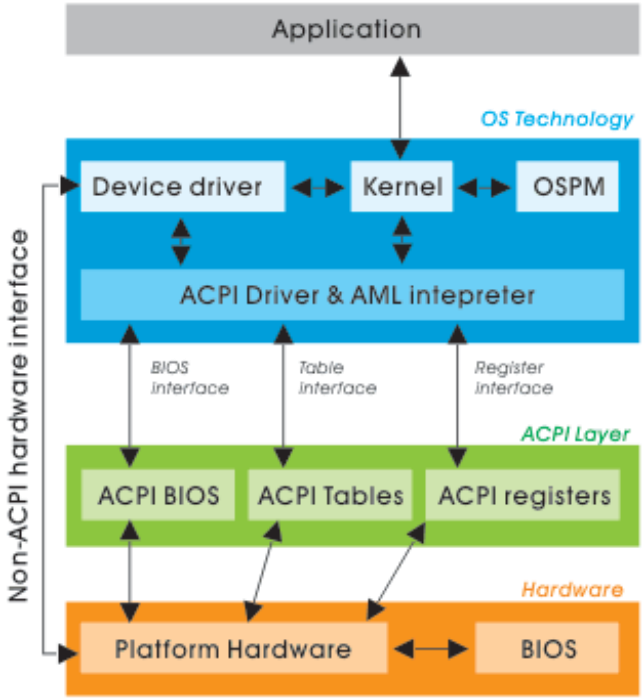
```
Add (X, Y, Z)
LEqual (X, Y)
```

- ASL 2.0

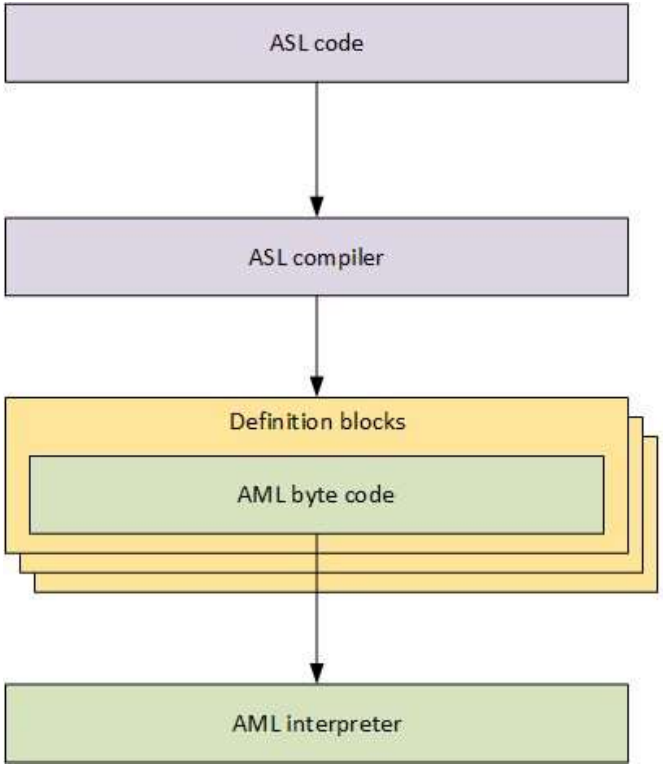
```
Z = X + Y
If (X == Y)
```

See
https://acpica.org/sites/acpica/files/ASL2.0Overview.pdf

ACPI


Application
OS Technology
Device driver
Kernel
OSPM
ACPI Driver & AML intepreter
Non-ACPI hardware interface
BIOS interface
Table interface
Register interface
ACPI Layer
ACPI BIOS
ACPI Tables
ACPI registers
Hardware
Platform Hardware
BIOS

ASL and AML

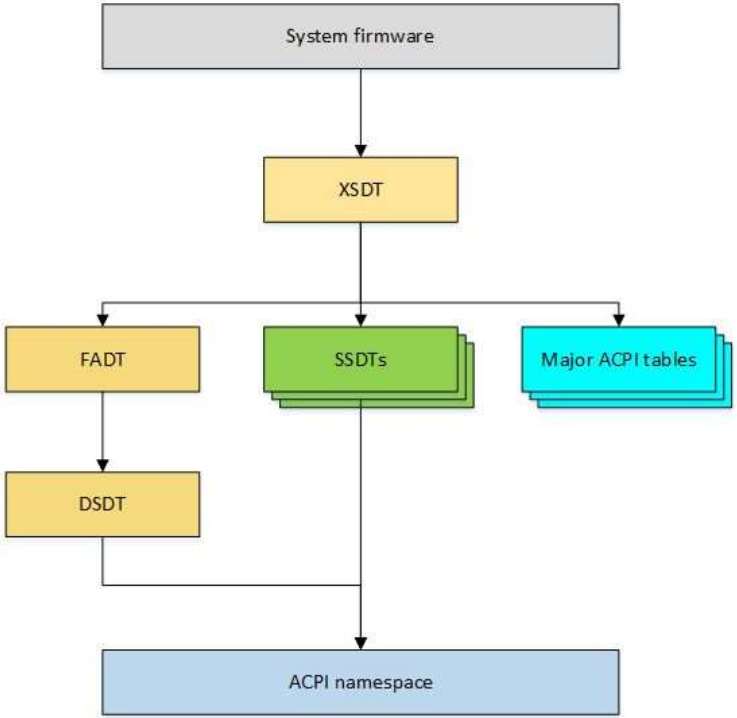


ACPI Source Langauge (ASL) code is used to define objects and control methods.

The ASL compiler translates ASL into ACPI Machine Language (AML) byte code contained within the ACPI definition blocks.

Definition blocks consist of an identifying table header and byte code that is executable by an AML interpreter.

ACPI initialization



System firmware updates the ACPI tables as necessary with information only available at runtime before handing off control to the boostrap loader.

The *XSDT* is the first table used by the OS's ACPI subsystem and contains the addresses of most of the other ACPI tables on the system.

The *XSDT* points to the *FADT*, the *SSDTs*, and other *major ACPI tables*.

The FADT directs the ACPI subsystem to the *DSDT*, which is the beginning of the namespace by virtue of being the first table that contains a definition block.

The ACPI subsystem then consumes the DSDT and begins building the *ACPI namespace* from the definition blocks. The XSDT also points to the SSDTs and adds them to the namespace.

extended root system description table

fixed ACPI description table
secondary system description table

differentiated system description table

ACPI can control

- System Power management
- Device Power Management
- Processor Power Management
- System Events
- Plug and Play
- Battery Management
- Thermal Management
- System Management Bus Controller

- sensors
- watchdogs

Links on the topic of ACPI

https://www.acpica.org/

https://01.org/

https://www.kernel.org/doc/Documentation/acpi/

https://wiki.ubuntu.com/Kernel/Reference/ACPITricksAndTips

https://wiki.linaro.org/LEG/Engineering/test-acpi

https://www.kernel.org/doc/ols/2005/ols2005v1-pages-59-76.pdf

Tools

-- iasl: compiles asl (acpi source language) into aml (acpi machine language), suitable for inclusion as a dsdt in system firmware. it also can disassemble aml, for debugging purposes.
-- acpibin: performs basic operations on binary aml files (e.g., comparison, data extraction)
-- acpidump: write out the current contents of acpi tables
-- acpiexec: simulate aml execution in order to debug method definitions
-- acpihelp: display help messages describing asl keywords and op-codes
-- acpinames: display complete acpi name space from input aml
-- acpisrc: manipulate the acpica source tree and format source files for specific environments
-- acpixtract: extract binary acpi tables from acpidump output (see also the pmtools package)

ACPI tools

- sudo apt install acpi
- https://wiki.ubuntu.com/BIOSandUbuntu
- sudo apt-get install iasl acpidump
  sudo acpidump -o acpidump.txt
  acpixtract acpidump.txt
  iasl -d dsdt.dat ssdt*.dat

How to

- sudo acpidump > acpidata.dat

  acpixtract -sDSDT acpidata.dat
  acpixtract -sSSDT acpidata.dat
- acpixtract -a acpidata.dat
- iasl -d DSDT.dat SSDT*.dat
- acpiexec -da dsdt.aml