**Федеральное государственное автономное образовательное учреждение**

**Высшего профессионального образования**

Санкт-Петербургский политехнический университет

Институт информационных технологий и управления

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №4**

«Анализ функционирования и кода стандартной утилиты lspci»

Работу выполнил студент гр. № 53501/3 Цыганов А.А.

Работу принял преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Душутина Е.В.

Санкт-Петербург

2016

# Цель работы:

В процессе работы необходимо провести анализ системной утилиты linux **lspci**:

* Описание утилиты
* Инструкция использования
* Привести примеры использования
* Рассмотреть исходный код утилиты

# Ход работы:

Для работы была использована виртуальная машина VMware Workstation 10.0.4, с образом ОС Linux Ubuntu 64-bit.

alex@ubuntu:~$ lsb\_release -a

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 15.10

Release: 15.10

Codename: wily

alex@ubuntu:~$ lscpu

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit

Byte Order: Little Endian

CPU(s): 8

On-line CPU(s) list: 0-7

Thread(s) per core: 1

Core(s) per socket: 2

Socket(s): 4

NUMA node(s): 1

Vendor ID: GenuineIntel

CPU family: 6

Model: 58

Model name: Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz

Stepping: 9

CPU MHz: 2394.575

BogoMIPS: 4789.15

Hypervisor vendor: VMware

Virtualization type: full

L1d cache: 32K

L1i cache: 32K

L2 cache: 256K

L3 cache: 6144K

NUMA node0 CPU(s): 0-7

# Описание утилиты lspci

Описание утилиты взяты из официального источника, перевода man-страницы для утилиты **lspci**.

**lspci** - выводит список всех устройств на шине PCI.

**СИНТАКСИС**

lspci [опции]

**ОПИСАНИЕ**

Утилита lspci служит для вывода информации о всех имеющихся в системе шинах PCI и всех подключенных к ним устройствах.

Если вы собираетесь отправить отчёт об ошибках в драйвере PCI-устройства или в самом lspci, пожалуйста включите в отчёт вывод команды "lspci -vvx".

**ОПЦИИ**

|  |  |
| --- | --- |
| -v | Предписывает lspci быть многословной и выводить подробную информацию о всех устройствах. |
| -vv | Указывает lspci быть очень многословной и показывать ещё больше информации (фактически всё, что PCI-устройство способно выдать). В этой странице руководства, точный смысл этих данных не разъясняется, если вы хотите узнать больше, посмотрите /usr/include/linux/pci.h или спецификации PCI. |
| -n | Показывать поставщика и коды PCI-устройства в числовом виде, взамен поиска в базе данных идентификаторов PCI. |
| -x | Показать шестнадцатеричный дамп первых 64 байт конфигурационного пространства PCI (стандартный заголовок). Полезно для отладки драйверов и самой lspci. |
| -xxx | Показать шестнадцатеричный дамп всего конфигурационного пространства PCI. Опция доступна только суперпользователю, поскольку некоторые устройства PCI могут вызвать сбой, когда вы попытаетесь прочитать неопределённые части конфигурационного пространства (такое поведение возможно и не нарушает стандарта PCI, хотя это по меньше мере несуразно). |
| -b | Магистральный вид. Показать все номера IRQ и адреса, так как их увидели устройства на шине, вместо увиденного ядром. |
| -t | Отобразить древовидную схему, содержащую все шины, мосты, устройства и соединения между ними. |
| -s [[<шина>]:][<слот>][.[<функция>]] | Показывать устройства только конкретной шины, слота и функции. Можно опустить каждый из компонентов адреса устройства или заменить на "\*", что означает "любое допустимое значение". Все числа шестнадцатеричные. Таким образом, "0:" отберёт все устройства на шине 0; "0" выделит все функции устройства 0 на любой шине; "0.3" выберет третью функцию устройства 0 на всех шинах; и ".4" покажет только четвёртую функцию каждого из устройств. |
| -d [<поставщик>]:[<устройство>] | Показывать абонентов шины с идентификаторами только определённого поставщика и устройства. Оба идентификатора задаются в шестнадцатеричном виде, и могут быть опущены или заменены на "\*", что подразумевает "любое допустимое значение". |
| -i <файл> | Использовать <файл>, как базу данных идентификаторов PCI, взамен /usr/share/pci.ids. |
| -p <каталог> | Использовать <каталог>, как каталог содержащий информацию шины PCI, вместо /proc/bus/pci. |
| -m | Дамп данных PCI-устройства в машинно-читаемой форме (поддерживается нормальный и многословный формат), что удобно для анализа с помощью скриптов. |
| -M | Активировать режим отображения шины, который тщательно сканирует шину для нахождения всех устройств, включая те, что скрыты за неправильно сконфигурированными мостами и т.п. Пожалуйста, обратите внимание, что это предназначено только для отладки и поскольку может привести к краху системы (только в случае наличия сбойных устройств, но к сожалению такое встречается), данная опция доступна только суперпользователю. К тому же не имеет смысла использование -M при методах доступа к PCI, которые непосредственно не связываются с аппаратными средствами, т.к. результаты будут (за исключением ошибок в lspci) идентичны выводу команды в нормальных режимах. |
| --version | Отобразить версию lspci. Эта опция должна применяться отдельно от других. |

**ФАЙЛЫ**

/usr/share/pci.ids

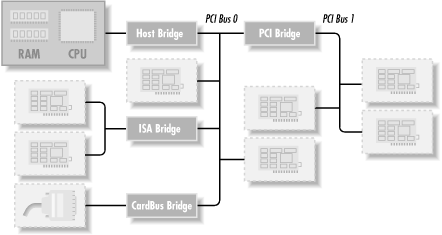
Список всех известных идентификаторов PCI (поставщики, устройства, классы и подклассы).

/proc/bus/pci

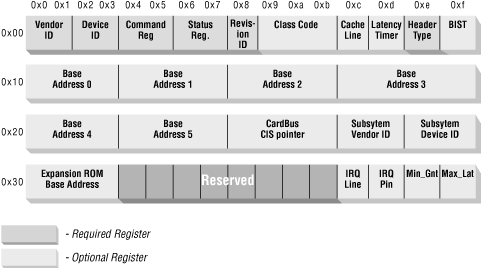
Интерфейс к конфигурационному пространству PCI, предоставляемый пост-2.1.82 ядрами Linux. Содержит подкаталоги для каждой шины с файлами пространств конфигурации и файлом устройств, содержащим список всех PCI-устройств.

Утилита lspci предназначена для вывода информации о всех PCI-шинах в системе, а также о всех устройствах, присоединенных к этим шинам.

Ниже приведена структура PCI систему компьютера:



PCI устройству при подключении к шине выделяется 256 байт:



# Примеры использования утилиты

# Запуск утилите без параметров. Вывод в консоль всех периферийных устройств подключенных к pci. Информация выводится в виде:

# ID устройства, Класс, Вендор, Название, ревизия(версия)

alex@ubuntu:~$ lspci

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)

00:11.0 PCI bridge: VMware PCI bridge (rev 02)

00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.7 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:17.7 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:18.7 PCI bridge: VMware PCI Express Root Port (rev 01)

02:00.0 USB controller: VMware USB1.1 UHCI Controller

02:01.0 Ethernet controller: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) (rev 01)

02:02.0 Multimedia audio controller: Ensoniq ES1371 / Creative Labs CT2518/ES1373 (rev 02)

02:03.0 USB controller: VMware USB2 EHCI Controller

02:05.0 SATA controller: VMware SATA AHCI controller

Чтобы получить больше информации, используем опцию –v и –vv (вывод подробной информации).

alex@ubuntu:~$ lspci -v

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

Subsystem: VMware Virtual Machine Chipset

Flags: bus master, medium devsel, latency 0

Kernel driver in use: agpgart-intel

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, 66MHz, medium devsel, latency 0

Bus: primary=00, secondary=01, subordinate=01, sec-latency=64

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

Subsystem: VMware Virtual Machine Chipset

Flags: bus master, medium devsel, latency 0

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01) (prog-if 8a [Master SecP PriP])

Subsystem: VMware Virtual Machine Chipset

Flags: bus master, medium devsel, latency 64

[virtual] Memory at 000001f0 (32-bit, non-prefetchable) [size=8]

[virtual] Memory at 000003f0 (type 3, non-prefetchable)

[virtual] Memory at 00000170 (32-bit, non-prefetchable) [size=8]

[virtual] Memory at 00000370 (type 3, non-prefetchable)

I/O ports at 1060 [size=16]

Kernel driver in use: ata\_piix

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

Subsystem: VMware Virtual Machine Chipset

Flags: medium devsel, IRQ 9

00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)

Subsystem: VMware Virtual Machine Communication Interface

Flags: bus master, medium devsel, latency 64, IRQ 16

I/O ports at 1080 [size=64]

Memory at febfe000 (64-bit, non-prefetchable) [size=8K]

Capabilities: <access denied>

Kernel driver in use: vmw\_vmci

00:0f.0 VGA compatible controller: VMware SVGA II Adapter (prog-if 00 [VGA controller])

Subsystem: VMware SVGA II Adapter

Flags: bus master, medium devsel, latency 64, IRQ 16

I/O ports at 1070 [size=16]

Memory at e8000000 (32-bit, prefetchable) [size=128M]

Memory at fe000000 (32-bit, non-prefetchable) [size=8M]

[virtual] Expansion ROM at c0000000 [disabled] [size=32K]

Capabilities: <access denied>

Kernel driver in use: vmwgfx

00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)

Subsystem: VMware LSI Logic Parallel SCSI Controller

Flags: bus master, medium devsel, latency 64, IRQ 17

I/O ports at 1400 [size=256]

Memory at feba0000 (64-bit, non-prefetchable) [size=128K]

Memory at febc0000 (64-bit, non-prefetchable) [size=128K]

[virtual] Expansion ROM at c0008000 [disabled] [size=16K]

Kernel driver in use: mptspi

00:11.0 PCI bridge: VMware PCI bridge (rev 02) (prog-if 01 [Subtractive decode])

Flags: bus master, medium devsel, latency 64

Bus: primary=00, secondary=02, subordinate=02, sec-latency=68

I/O behind bridge: 00002000-00003fff

Memory behind bridge: fd500000-fdffffff

Prefetchable memory behind bridge: 00000000e7b00000-00000000e7ffffff

Capabilities: <access denied>

00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 24

Bus: primary=00, secondary=03, subordinate=03, sec-latency=0

I/O behind bridge: 00004000-00004fff

Memory behind bridge: fd400000-fd4fffff

Prefetchable memory behind bridge: 00000000e7a00000-00000000e7afffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 25

Bus: primary=00, secondary=04, subordinate=04, sec-latency=0

I/O behind bridge: 00008000-00008fff

Memory behind bridge: fd000000-fd0fffff

Prefetchable memory behind bridge: 00000000e7600000-00000000e76fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 26

Bus: primary=00, secondary=05, subordinate=05, sec-latency=0

I/O behind bridge: 0000c000-0000cfff

Memory behind bridge: fcc00000-fccfffff

Prefetchable memory behind bridge: 00000000e7200000-00000000e72fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 27

Bus: primary=00, secondary=06, subordinate=06, sec-latency=0

Memory behind bridge: fc800000-fc8fffff

Prefetchable memory behind bridge: 00000000e6e00000-00000000e6efffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 28

Bus: primary=00, secondary=07, subordinate=07, sec-latency=0

Memory behind bridge: fc400000-fc4fffff

Prefetchable memory behind bridge: 00000000e6a00000-00000000e6afffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 29

Bus: primary=00, secondary=08, subordinate=08, sec-latency=0

Memory behind bridge: fc000000-fc0fffff

Prefetchable memory behind bridge: 00000000e6600000-00000000e66fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 30

Bus: primary=00, secondary=09, subordinate=09, sec-latency=0

Memory behind bridge: fbc00000-fbcfffff

Prefetchable memory behind bridge: 00000000e6200000-00000000e62fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 31

Bus: primary=00, secondary=0a, subordinate=0a, sec-latency=0

Memory behind bridge: fb800000-fb8fffff

Prefetchable memory behind bridge: 00000000e5e00000-00000000e5efffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 32

Bus: primary=00, secondary=0b, subordinate=0b, sec-latency=0

I/O behind bridge: 00005000-00005fff

Memory behind bridge: fd300000-fd3fffff

Prefetchable memory behind bridge: 00000000e7900000-00000000e79fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 33

Bus: primary=00, secondary=0c, subordinate=0c, sec-latency=0

I/O behind bridge: 00009000-00009fff

Memory behind bridge: fcf00000-fcffffff

Prefetchable memory behind bridge: 00000000e7500000-00000000e75fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.2 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 34

Bus: primary=00, secondary=0d, subordinate=0d, sec-latency=0

I/O behind bridge: 0000d000-0000dfff

Memory behind bridge: fcb00000-fcbfffff

Prefetchable memory behind bridge: 00000000e7100000-00000000e71fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.3 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 35

Bus: primary=00, secondary=0e, subordinate=0e, sec-latency=0

Memory behind bridge: fc700000-fc7fffff

Prefetchable memory behind bridge: 00000000e6d00000-00000000e6dfffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.4 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 36

Bus: primary=00, secondary=0f, subordinate=0f, sec-latency=0

Memory behind bridge: fc300000-fc3fffff

Prefetchable memory behind bridge: 00000000e6900000-00000000e69fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.5 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 37

Bus: primary=00, secondary=10, subordinate=10, sec-latency=0

Memory behind bridge: fbf00000-fbffffff

Prefetchable memory behind bridge: 00000000e6500000-00000000e65fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.6 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 38

Bus: primary=00, secondary=11, subordinate=11, sec-latency=0

Memory behind bridge: fbb00000-fbbfffff

Prefetchable memory behind bridge: 00000000e6100000-00000000e61fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:16.7 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 39

Bus: primary=00, secondary=12, subordinate=12, sec-latency=0

Memory behind bridge: fb700000-fb7fffff

Prefetchable memory behind bridge: 00000000e5d00000-00000000e5dfffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.0 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 40

Bus: primary=00, secondary=13, subordinate=13, sec-latency=0

I/O behind bridge: 00006000-00006fff

Memory behind bridge: fd200000-fd2fffff

Prefetchable memory behind bridge: 00000000e7800000-00000000e78fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.1 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 41

Bus: primary=00, secondary=14, subordinate=14, sec-latency=0

I/O behind bridge: 0000a000-0000afff

Memory behind bridge: fce00000-fcefffff

Prefetchable memory behind bridge: 00000000e7400000-00000000e74fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.2 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 42

Bus: primary=00, secondary=15, subordinate=15, sec-latency=0

I/O behind bridge: 0000e000-0000efff

Memory behind bridge: fca00000-fcafffff

Prefetchable memory behind bridge: 00000000e7000000-00000000e70fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.3 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 43

Bus: primary=00, secondary=16, subordinate=16, sec-latency=0

Memory behind bridge: fc600000-fc6fffff

Prefetchable memory behind bridge: 00000000e6c00000-00000000e6cfffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.4 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 44

Bus: primary=00, secondary=17, subordinate=17, sec-latency=0

Memory behind bridge: fc200000-fc2fffff

Prefetchable memory behind bridge: 00000000e6800000-00000000e68fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.5 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 45

Bus: primary=00, secondary=18, subordinate=18, sec-latency=0

Memory behind bridge: fbe00000-fbefffff

Prefetchable memory behind bridge: 00000000e6400000-00000000e64fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.6 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 46

Bus: primary=00, secondary=19, subordinate=19, sec-latency=0

Memory behind bridge: fba00000-fbafffff

Prefetchable memory behind bridge: 00000000e6000000-00000000e60fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:17.7 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 47

Bus: primary=00, secondary=1a, subordinate=1a, sec-latency=0

Memory behind bridge: fb600000-fb6fffff

Prefetchable memory behind bridge: 00000000e5c00000-00000000e5cfffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.0 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 48

Bus: primary=00, secondary=1b, subordinate=1b, sec-latency=0

I/O behind bridge: 00007000-00007fff

Memory behind bridge: fd100000-fd1fffff

Prefetchable memory behind bridge: 00000000e7700000-00000000e77fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.1 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 49

Bus: primary=00, secondary=1c, subordinate=1c, sec-latency=0

I/O behind bridge: 0000b000-0000bfff

Memory behind bridge: fcd00000-fcdfffff

Prefetchable memory behind bridge: 00000000e7300000-00000000e73fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.2 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 50

Bus: primary=00, secondary=1d, subordinate=1d, sec-latency=0

Memory behind bridge: fc900000-fc9fffff

Prefetchable memory behind bridge: 00000000e6f00000-00000000e6ffffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.3 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 51

Bus: primary=00, secondary=1e, subordinate=1e, sec-latency=0

Memory behind bridge: fc500000-fc5fffff

Prefetchable memory behind bridge: 00000000e6b00000-00000000e6bfffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.4 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 52

Bus: primary=00, secondary=1f, subordinate=1f, sec-latency=0

Memory behind bridge: fc100000-fc1fffff

Prefetchable memory behind bridge: 00000000e6700000-00000000e67fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.5 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 53

Bus: primary=00, secondary=20, subordinate=20, sec-latency=0

Memory behind bridge: fbd00000-fbdfffff

Prefetchable memory behind bridge: 00000000e6300000-00000000e63fffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.6 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 54

Bus: primary=00, secondary=21, subordinate=21, sec-latency=0

Memory behind bridge: fb900000-fb9fffff

Prefetchable memory behind bridge: 00000000e5f00000-00000000e5ffffff

Capabilities: <access denied>

Kernel driver in use: pcieport

00:18.7 PCI bridge: VMware PCI Express Root Port (rev 01) (prog-if 00 [Normal decode])

Flags: bus master, fast devsel, latency 0, IRQ 55

Bus: primary=00, secondary=22, subordinate=22, sec-latency=0

Memory behind bridge: fb500000-fb5fffff

Prefetchable memory behind bridge: 00000000e5b00000-00000000e5bfffff

Capabilities: <access denied>

Kernel driver in use: pcieport

02:00.0 USB controller: VMware USB1.1 UHCI Controller (prog-if 00 [UHCI])

DeviceName: usb

Subsystem: VMware Device 1976

Physical Slot: 32

Flags: bus master, medium devsel, latency 64, IRQ 18

I/O ports at 2080 [size=32]

Kernel driver in use: uhci\_hcd

02:01.0 Ethernet controller: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) (rev 01)

DeviceName: Ethernet0

Subsystem: VMware PRO/1000 MT Single Port Adapter

Physical Slot: 33

Flags: bus master, 66MHz, medium devsel, latency 0, IRQ 19

Memory at fd5c0000 (64-bit, non-prefetchable) [size=128K]

Memory at fdff0000 (64-bit, non-prefetchable) [size=64K]

I/O ports at 2000 [size=64]

[virtual] Expansion ROM at fd500000 [disabled] [size=64K]

Capabilities: <access denied>

Kernel driver in use: e1000

02:02.0 Multimedia audio controller: Ensoniq ES1371 / Creative Labs CT2518/ES1373 (rev 02)

DeviceName: sound

Subsystem: Ensoniq AudioPCI 64V/128 / Creative CT4810/CT5803/CT5806 [Sound Blaster PCI]

Physical Slot: 34

Flags: bus master, medium devsel, latency 64, IRQ 16

I/O ports at 2040 [size=64]

Kernel driver in use: snd\_ens1371

02:03.0 USB controller: VMware USB2 EHCI Controller (prog-if 20 [EHCI])

DeviceName: ehci

Subsystem: VMware USB2 EHCI Controller

Physical Slot: 35

Flags: bus master, fast devsel, latency 64, IRQ 17

Memory at fd5ef000 (32-bit, non-prefetchable) [size=4K]

Kernel driver in use: ehci-pci

02:05.0 SATA controller: VMware SATA AHCI controller (prog-if 01 [AHCI 1.0])

DeviceName: sata0

Subsystem: VMware SATA AHCI controller

Physical Slot: 37

Flags: bus master, 66MHz, fast devsel, latency 64, IRQ 56

Memory at fd5ee000 (32-bit, non-prefetchable) [size=4K]

[virtual] Expansion ROM at fd510000 [disabled] [size=64K]

Capabilities: <access denied>

Kernel driver in use: ahci

Далее выполним утилиту lspci c параметром –vv и сравним данные. В итоге получены те же устройства, но с более развёрнутыми данными.

alex@ubuntu:~$ lspci -vv

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

Subsystem: VMware Virtual Machine Chipset

Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

Status: Cap- 66MHz- UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0

Kernel driver in use: agpgart-intel

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01) (prog-if 00 [Normal decode])

Control: I/O+ Mem+ BusMaster+ SpecCycle+ MemWINV+ VGASnoop- ParErr- Stepping- SERR+ FastB2B- DisINTx-

Status: Cap- 66MHz+ UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0

Bus: primary=00, secondary=01, subordinate=01, sec-latency=64

Secondary status: 66MHz+ FastB2B+ ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- <SERR- <PERR-

BridgeCtl: Parity- SERR- NoISA+ VGA- MAbort- >Reset- FastB2B+

PriDiscTmr- SecDiscTmr- DiscTmrStat- DiscTmrSERREn-

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

Subsystem: VMware Virtual Machine Chipset

Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

Status: Cap- 66MHz- UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01) (prog-if 8a [Master SecP PriP])

Subsystem: VMware Virtual Machine Chipset

Control: I/O+ Mem- BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

Status: Cap- 66MHz- UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 64

Region 0: [virtual] Memory at 000001f0 (32-bit, non-prefetchable) [size=8]

Region 1: [virtual] Memory at 000003f0 (type 3, non-prefetchable)

Region 2: [virtual] Memory at 00000170 (32-bit, non-prefetchable) [size=8]

Region 3: [virtual] Memory at 00000370 (type 3, non-prefetchable)

Region 4: I/O ports at 1060 [size=16]

Kernel driver in use: ata\_piix

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

Subsystem: VMware Virtual Machine Chipset

Control: I/O+ Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

Status: Cap- 66MHz- UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Interrupt: pin ? routed to IRQ 9

...

02:01.0 Ethernet controller: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) (rev 01)

DeviceName: Ethernet0

Subsystem: VMware PRO/1000 MT Single Port Adapter

Physical Slot: 33

Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV+ VGASnoop- ParErr- Stepping- SERR+ FastB2B- DisINTx-

Status: Cap+ 66MHz+ UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0 (63750ns min), Cache Line Size: 64 bytes

Interrupt: pin A routed to IRQ 19

Region 0: Memory at fd5c0000 (64-bit, non-prefetchable) [size=128K]

Region 2: Memory at fdff0000 (64-bit, non-prefetchable) [size=64K]

Region 4: I/O ports at 2000 [size=64]

[virtual] Expansion ROM at fd500000 [disabled] [size=64K]

Capabilities: <access denied>

Kernel driver in use: e1000

02:02.0 Multimedia audio controller: Ensoniq ES1371 / Creative Labs CT2518/ES1373 (rev 02)

DeviceName: sound

Subsystem: Ensoniq AudioPCI 64V/128 / Creative CT4810/CT5803/CT5806 [Sound Blaster PCI]

Physical Slot: 34

Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

Status: Cap- 66MHz- UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 64 (1500ns min, 63750ns max)

Interrupt: pin A routed to IRQ 16

Region 0: I/O ports at 2040 [size=64]

Kernel driver in use: snd\_ens1371

02:03.0 USB controller: VMware USB2 EHCI Controller (prog-if 20 [EHCI])

DeviceName: ehci

Subsystem: VMware USB2 EHCI Controller

Physical Slot: 35

Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

Status: Cap- 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 64 (1500ns min, 63750ns max)

Interrupt: pin A routed to IRQ 17

Region 0: Memory at fd5ef000 (32-bit, non-prefetchable) [size=4K]

Kernel driver in use: ehci-pci

02:05.0 SATA controller: VMware SATA AHCI controller (prog-if 01 [AHCI 1.0])

DeviceName: sata0

Subsystem: VMware SATA AHCI controller

Physical Slot: 37

Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+

Status: Cap+ 66MHz+ UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 64

Interrupt: pin A routed to IRQ 56

Region 5: Memory at fd5ee000 (32-bit, non-prefetchable) [size=4K]

[virtual] Expansion ROM at fd510000 [disabled] [size=64K]

Capabilities: <access denied>

Kernel driver in use: ahci

Утилита **lspci** вначале считывает информацию с PCI-шины, а потом дополнительную информцию ищет в собственной базе данных, которая находится в файле /usr/share/hwdata/pci.ids и содержит такие данные как идентификатор обрудования, производитель, устройства, классы и подклассы.

Проверим запуск утилиты lspci с параметром –n (показывать поставщика и коды PCI-устройства в числовом виде, взамен поиска в базе данных идентификаторов PCI).

alex@ubuntu:~$ lspci -n

00:00.0 0600: 8086:7190 (rev 01)

00:01.0 0604: 8086:7191 (rev 01)

00:07.0 0601: 8086:7110 (rev 08)

00:07.1 0101: 8086:7111 (rev 01)

00:07.3 0680: 8086:7113 (rev 08)

00:07.7 0880: 15ad:0740 (rev 10)

00:0f.0 0300: 15ad:0405

00:10.0 0100: 1000:0030 (rev 01)

00:11.0 0604: 15ad:0790 (rev 02)

00:15.0 0604: 15ad:07a0 (rev 01)

00:15.1 0604: 15ad:07a0 (rev 01)

00:15.2 0604: 15ad:07a0 (rev 01)

00:15.3 0604: 15ad:07a0 (rev 01)

00:15.4 0604: 15ad:07a0 (rev 01)

00:15.5 0604: 15ad:07a0 (rev 01)

00:15.6 0604: 15ad:07a0 (rev 01)

00:15.7 0604: 15ad:07a0 (rev 01)

00:16.0 0604: 15ad:07a0 (rev 01)

00:16.1 0604: 15ad:07a0 (rev 01)

00:16.2 0604: 15ad:07a0 (rev 01)

00:16.3 0604: 15ad:07a0 (rev 01)

00:16.4 0604: 15ad:07a0 (rev 01)

00:16.5 0604: 15ad:07a0 (rev 01)

00:16.6 0604: 15ad:07a0 (rev 01)

00:16.7 0604: 15ad:07a0 (rev 01)

00:17.0 0604: 15ad:07a0 (rev 01)

00:17.1 0604: 15ad:07a0 (rev 01)

00:17.2 0604: 15ad:07a0 (rev 01)

00:17.3 0604: 15ad:07a0 (rev 01)

00:17.4 0604: 15ad:07a0 (rev 01)

00:17.5 0604: 15ad:07a0 (rev 01)

00:17.6 0604: 15ad:07a0 (rev 01)

00:17.7 0604: 15ad:07a0 (rev 01)

00:18.0 0604: 15ad:07a0 (rev 01)

00:18.1 0604: 15ad:07a0 (rev 01)

00:18.2 0604: 15ad:07a0 (rev 01)

00:18.3 0604: 15ad:07a0 (rev 01)

00:18.4 0604: 15ad:07a0 (rev 01)

00:18.5 0604: 15ad:07a0 (rev 01)

00:18.6 0604: 15ad:07a0 (rev 01)

00:18.7 0604: 15ad:07a0 (rev 01)

02:00.0 0c03: 15ad:0774

02:01.0 0200: 8086:100f (rev 01)

**02:02.0 0401: 1274:1371 (rev 02)**

02:03.0 0c03: 15ad:0770

02:05.0 0106: 15ad:07e0

Например, для устройства: 02:02.0 0401: 1274:1371 (rev 02) это означает,

* 02:02.0- bus number (02), device number (02) and function (0)
* 0401- device class
* 1274 - vendor ID
* 1371 - device ID

02:02.0 Multimedia audio controller: Ensoniq ES1371 / Creative Labs CT2518/ES1373 (rev 02)

Альтернативный способ получения аппаратного состава компьютера, напрямую вытаскивать данные об устройствах из файловой системы /sys/bus/pci/devices. Ниже приведен пример:

alex@ubuntu:~$ ls -la /sys/bus/pci/devices

total 0

drwxr-xr-x 2 root root 0 марта 10 13:23 .

drwxr-xr-x 5 root root 0 марта 10 12:17 ..

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:00.0 -> ../../../devices/pci0000:00/0000:00:00.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:01.0 -> ../../../devices/pci0000:00/0000:00:01.0

**lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:07.0 -> ../../../devices/pci0000:00/0000:00:07.0**

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:07.1 -> ../../../devices/pci0000:00/0000:00:07.1

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:07.3 -> ../../../devices/pci0000:00/0000:00:07.3

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:07.7 -> ../../../devices/pci0000:00/0000:00:07.7

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:0f.0 -> ../../../devices/pci0000:00/0000:00:0f.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:10.0 -> ../../../devices/pci0000:00/0000:00:10.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:11.0 -> ../../../devices/pci0000:00/0000:00:11.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.0 -> ../../../devices/pci0000:00/0000:00:15.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.1 -> ../../../devices/pci0000:00/0000:00:15.1

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.2 -> ../../../devices/pci0000:00/0000:00:15.2

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.3 -> ../../../devices/pci0000:00/0000:00:15.3

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.4 -> ../../../devices/pci0000:00/0000:00:15.4

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.5 -> ../../../devices/pci0000:00/0000:00:15.5

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.6 -> ../../../devices/pci0000:00/0000:00:15.6

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:15.7 -> ../../../devices/pci0000:00/0000:00:15.7

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.0 -> ../../../devices/pci0000:00/0000:00:16.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.1 -> ../../../devices/pci0000:00/0000:00:16.1

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.2 -> ../../../devices/pci0000:00/0000:00:16.2

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.3 -> ../../../devices/pci0000:00/0000:00:16.3

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.4 -> ../../../devices/pci0000:00/0000:00:16.4

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.5 -> ../../../devices/pci0000:00/0000:00:16.5

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.6 -> ../../../devices/pci0000:00/0000:00:16.6

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:16.7 -> ../../../devices/pci0000:00/0000:00:16.7

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.0 -> ../../../devices/pci0000:00/0000:00:17.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.1 -> ../../../devices/pci0000:00/0000:00:17.1

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.2 -> ../../../devices/pci0000:00/0000:00:17.2

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.3 -> ../../../devices/pci0000:00/0000:00:17.3

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.4 -> ../../../devices/pci0000:00/0000:00:17.4

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.5 -> ../../../devices/pci0000:00/0000:00:17.5

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.6 -> ../../../devices/pci0000:00/0000:00:17.6

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:17.7 -> ../../../devices/pci0000:00/0000:00:17.7

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.0 -> ../../../devices/pci0000:00/0000:00:18.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.1 -> ../../../devices/pci0000:00/0000:00:18.1

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.2 -> ../../../devices/pci0000:00/0000:00:18.2

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.3 -> ../../../devices/pci0000:00/0000:00:18.3

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.4 -> ../../../devices/pci0000:00/0000:00:18.4

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.5 -> ../../../devices/pci0000:00/0000:00:18.5

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.6 -> ../../../devices/pci0000:00/0000:00:18.6

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:00:18.7 -> ../../../devices/pci0000:00/0000:00:18.7

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:02:00.0 -> ../../../devices/pci0000:00/0000:00:11.0/0000:02:00.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:02:01.0 -> ../../../devices/pci0000:00/0000:00:11.0/0000:02:01.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:02:02.0 -> ../../../devices/pci0000:00/0000:00:11.0/0000:02:02.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:02:03.0 -> ../../../devices/pci0000:00/0000:00:11.0/0000:02:03.0

lrwxrwxrwx 1 root root 0 марта 10 12:17 0000:02:05.0 -> ../../../devices/pci0000:00/0000:00:11.0/0000:02:05.0

Например, для выбранного устройства "0000:00:07.0" можно получить информацию:

0000 : PCI domain (each domain can contain up to 256 PCI buses)

00 : the bus number the device is attached to

07 : the device number

.0 : PCI device function

Получения подробных данных об устройстве в директории cd 0000:00:07.0.

# Анализ исходного кода утилиты lspci

В первую очередь выясним, откуда утилита получает информацию. Для этого с помощью утилиты strace отобразим все системные вызовы, которые совершает lspci. Ниже приведен отрывок из вывода strace lspci.

alex@ubuntu:~$ strace -o ~/1.txt lspci

execve("/usr/bin/lspci", ["lspci"], [/\* 73 vars \*/]) = 0

brk(0) = 0x1bc6000

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cdf000

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

open("/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=86295, ...}) = 0

mmap(NULL, 86295, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f6c87cc9000

close(3) = 0

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

open("/lib/x86\_64-linux-gnu/libpci.so.3", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0p$\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=52416, ...}) = 0

mmap(NULL, 2147472, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f6c878b1000

mprotect(0x7f6c878bd000, 2093056, PROT\_NONE) = 0

mmap(0x7f6c87abc000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f6c87abc000

close(3) = 0

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

open("/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`\v\2\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=1869392, ...}) = 0

mmap(NULL, 3972864, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f6c874e7000

mprotect(0x7f6c876a7000, 2097152, PROT\_NONE) = 0

mmap(0x7f6c878a7000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c0000) = 0x7f6c878a7000

mmap(0x7f6c878ad000, 16128, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f6c878ad000

close(3) = 0

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

open("/lib/x86\_64-linux-gnu/libz.so.1", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360\35\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=104824, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cc8000

mmap(NULL, 2199880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f6c872cd000

mprotect(0x7f6c872e6000, 2093056, PROT\_NONE) = 0

mmap(0x7f6c874e5000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f6c874e5000

close(3) = 0

access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory)

open("/lib/x86\_64-linux-gnu/libresolv.so.2", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3209\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=105328, ...}) = 0

mmap(NULL, 2210632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f6c870b1000

mprotect(0x7f6c870c8000, 2097152, PROT\_NONE) = 0

mmap(0x7f6c872c8000, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17000) = 0x7f6c872c8000

mmap(0x7f6c872cb000, 6984, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f6c872cb000

close(3) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cc7000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cc6000

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cc5000

arch\_prctl(ARCH\_SET\_FS, 0x7f6c87cc6700) = 0

mprotect(0x7f6c878a7000, 16384, PROT\_READ) = 0

mprotect(0x7f6c872c8000, 8192, PROT\_READ) = 0

mprotect(0x7f6c874e5000, 4096, PROT\_READ) = 0

mprotect(0x7f6c87abc000, 4096, PROT\_READ) = 0

mprotect(0x610000, 4096, PROT\_READ) = 0

mprotect(0x7f6c87ce1000, 4096, PROT\_READ) = 0

munmap(0x7f6c87cc9000, 86295) = 0

brk(0) = 0x1bc6000

brk(0x1be7000) = 0x1be7000

access("/sys/bus/pci", R\_OK) = 0

openat(AT\_FDCWD, "/sys/bus/pci/devices", O\_RDONLY|O\_NONBLOCK|O\_DIRECTORY|O\_CLOEXEC) = 3

getdents(3, /\* 48 entries \*/, 32768) = 1520

open("/sys/bus/pci/devices/0000:00:00.0/resource", O\_RDONLY) = 4

fstat(4, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(4, "0x0000000000000000 0x00000000000"..., 4096) = 741

close(4) = 0

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/devices/0000:00:00.0/irq", O\_RDONLY) = 4

read(4, "0\n", 1024) = 2

close(4) = 0

open("/sys/bus/pci/devices/0000:00:00.0/vendor", O\_RDONLY) = 4

read(4, "0x8086\n", 1024) = 7

close(4) = 0

open("/sys/bus/pci/devices/0000:00:00.0/device", O\_RDONLY) = 4

read(4, "0x7190\n", 1024) = 7

close(4) = 0

open("/sys/bus/pci/devices/0000:00:00.0/class", O\_RDONLY) = 4

read(4, "0x060000\n", 1024) = 9

close(4) = 0

...

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/slots/33/address", O\_RDONLY) = 5

fstat(5, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(5, "0000:02:01\n", 4096) = 11

close(5) = 0

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/slots/34/address", O\_RDONLY) = 5

fstat(5, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(5, "0000:02:02\n", 4096) = 11

close(5) = 0

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/slots/35/address", O\_RDONLY) = 5

fstat(5, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(5, "0000:02:03\n", 4096) = 11

close(5) = 0

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/slots/36/address", O\_RDONLY) = 5

fstat(5, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(5, "0000:02:04\n", 4096) = 11

close(5) = 0

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/slots/37/address", O\_RDONLY) = 5

fstat(5, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(5, "0000:02:05\n", 4096) = 11

close(5) = 0

munmap(0x7f6c87cde000, 4096) = 0

open("/sys/bus/pci/slots/38/address", O\_RDONLY) = 5

fstat(5, {st\_mode=S\_IFREG|0444, st\_size=4096, ...}) = 0

mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f6c87cde000

read(5, "0000:02:06\n", 4096) = 11

close(5) = 0

...

open("/sys/bus/pci/devices/0000:00:17.0/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\23\23\0``\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:17.0/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.7/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\22\22\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.7/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.6/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\21\21\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.6/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.5/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\20\20\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.5/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.4/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\17\17\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.4/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.3/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\16\16\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.3/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.2/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\r\r\0\320\320\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.2/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.1/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\f\f\0\220\220\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.1/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:16.0/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\v\v\0PP\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:16.0/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:15.7/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\n\n\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:15.7/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

close(3) = 0

open("/sys/bus/pci/devices/0000:00:15.6/config", O\_RDONLY) = 3

pread(3, "\255\25\240\7\7\0\20\0\1\0\4\6\10\0\201\0\0\0\0\0\0\0\0\0\0\t\t\0\360\0\0\0"..., 64, 0) = 64

open("/sys/bus/pci/devices/0000:00:15.6/label", O\_RDONLY) = -1 ENOENT (No such file or directory)

...

read(4, "lyzer-RT CompactPCI Serial PWR-A"..., 16384) = 16384

read(4, "iness multimedia\n\t8904 86c365, "..., 16384) = 16384

read(4, " Security Controller\n\t080d Moor"..., 16384) = 16384

read(4, "Crystal Beach DMA Channel 3\n\t0e2"..., 16384) = 16384

read(4, "Latitude D500\n\t\t8086 103d 82562"..., 16384) = 16384

read(4, "port)\n\t\t0e11 b0e1 NC3133 Fast E"..., 16384) = 16384

brk(0x1d1a000) = 0x1d1a000

read(4, "004 Ethernet Converged Network "..., 16384) = 16384

read(4, "torage Control Unit\n\t1d5b C602 "..., 16384) = 16384

read(4, "rk Connection\n\t\t1186 7801 Ether"..., 16384) = 16384

read(4, "ard (865PE)\n\t\t1462 7280 865PE N"..., 16384) = 16384

read(4, "op\n\t\t1028 0188 Inspiron 6000 la"..., 16384) = 16384

read(4, "0cc Pavilion dv6700\n\t\t103c 30d9"..., 16384) = 16384

brk(0x1d3c000) = 0x1d3c000

read(4, "same ID possibly also on other A"..., 16384) = 16384

read(4, " Decoder\n\t2e5d CE Media Process"..., 16384) = 16384

read(4, "C\n\t\te4bf 0cc9 CC9-SAMBA\n\t\te4bf "..., 16384) = 16384

read(4, "0b Latitude E6510\n\t\t15d9 060d "..., 16384) = 16384

read(4, "f38 Xeon E7 v4/Xeon E5 v4/Xeon "..., 16384) = 16384

read(4, "eries Chipset Family PCI Express"..., 16384) = 16384

brk(0x1d5e000) = 0x1d5e000

read(4, "rise Point-H Gigabit Ethernet Co"..., 16384) = 16384

read(4, "C RAID)\n\t0450 ASC-1405 Unified "..., 16384) = 16384

read(4, " Wireless controller\n\t00 IRDA c"..., 16384) = 887

read(4, "", 15497) = 0

read(4, "", 16384) = 0

close(4) = 0

write(1, "00:00.0 Host bridge: Intel Corpo"..., 88) = 88

write(1, "00:01.0 PCI bridge: Intel Corpor"..., 86) = 86

write(1, "00:07.0 ISA bridge: Intel Corpor"..., 71) = 71

write(1, "00:07.1 IDE interface: Intel Cor"..., 74) = 74

write(1, "00:07.3 Bridge: Intel Corporatio"..., 68) = 68

write(1, "00:07.7 System peripheral: VMwar"..., 83) = 83

write(1, "00:0f.0 VGA compatible controlle"..., 58) = 58

write(1, "00:10.0 SCSI storage controller:"..., 112) = 112

write(1, "00:11.0 PCI bridge: VMware PCI b"..., 47) = 47

write(1, "00:15.0 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.1 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.2 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.3 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.4 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.5 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.6 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:15.7 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.0 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.1 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.2 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.3 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.4 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.5 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.6 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:16.7 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.0 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.1 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.2 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.3 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.4 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.5 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.6 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:17.7 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.0 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.1 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.2 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.3 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.4 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.5 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.6 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "00:18.7 PCI bridge: VMware PCI E"..., 58) = 58

write(1, "02:00.0 USB controller: VMware U"..., 54) = 54

write(1, "02:01.0 Ethernet controller: Int"..., 101) = 101

write(1, "02:02.0 Multimedia audio control"..., 91) = 91

write(1, "02:03.0 USB controller: VMware U"..., 52) = 52

write(1, "02:05.0 SATA controller: VMware "..., 53) = 53

close(3) = 0

brk(0x1d5c000) = 0x1d5c000

brk(0x1d5a000) = 0x1d5a000

brk(0x1d58000) = 0x1d58000

brk(0x1d56000) = 0x1d56000

brk(0x1d54000) = 0x1d54000

brk(0x1d52000) = 0x1d52000

brk(0x1d50000) = 0x1d50000

brk(0x1d4e000) = 0x1d4e000

brk(0x1d4c000) = 0x1d4c000

brk(0x1d4a000) = 0x1d4a000

brk(0x1d48000) = 0x1d48000

brk(0x1d46000) = 0x1d46000

...

brk(0x1c0d000) = 0x1c0d000

brk(0x1c0b000) = 0x1c0b000

brk(0x1c09000) = 0x1c09000

brk(0x1c07000) = 0x1c07000

brk(0x1c05000) = 0x1c05000

brk(0x1c03000) = 0x1c03000

brk(0x1c01000) = 0x1c01000

brk(0x1bff000) = 0x1bff000

brk(0x1be9000) = 0x1be9000

exit\_group(0) = ?

+++ exited with 0 +++

Как видно, основной источник данных является ресурс **/sys/bus/pci.** . Основные системные функции по открытию и закрытию файлов в катологах /sys/bus/pci. Это означает что утилита производит выборку данных из директории “/sys/bus/pci”, т.к. в этой директории все данные об устройствах, подключённых к PCI.

Далее происходит выборка данных и затем преобразовывает в структуры:

struct device {

struct device \*next;

struct pci\_dev \*dev;

unsigned int config\_cached, config\_bufsize;

byte \*config;

byte \*present;

};

В итоге данные найденных устройствах выводятся в консоль в виде последовательного списка или в виде структуры дерева.

Далее рассмотрим исходный код утилиты lspci. Ниже приведена структура файлов lspci проекта 5 директорий, 81 файл:

alex@ubuntu:~/Dropbox/политех/Системное программирование/Л. р. № 4/lspci$ tree

.

├── ChangeLog

├── common.c

├── compat

│   ├── getopt.c

│   ├── getopt.h

│   └── README

├── COPYING

├── example.c

├── example.sublime-workspace

├── install.tup

├── lib

│   ├── access.c

│   ├── aix-device.c

│   ├── caps.c

│   ├── config.h

│   ├── configure

│   ├── dump.c

│   ├── fbsd-device.c

│   ├── filter.c

│   ├── generic.c

│   ├── header.h

│   ├── i386-io-cygwin.h

│   ├── i386-io-hurd.h

│   ├── i386-io-linux.h

│   ├── i386-io-sunos.h

│   ├── i386-io-windows.h

│   ├── i386-ports.c

│   ├── init.c

│   ├── internal.h

│   ├── libpci.pc.in

│   ├── libpci.ver

│   ├── Makefile

│   ├── names.c

│   ├── names-cache.c

│   ├── names.h

│   ├── names-hash.c

│   ├── names-net.c

│   ├── names-parse.c

│   ├── nbsd-libpci.c

│   ├── obsd-device.c

│   ├── params.c

│   ├── pci.h

│   ├── pread.h

│   ├── proc.c

│   ├── sysdep.h

│   ├── sysfs.c

│   ├── Tupfile

│   └── types.h

├── ls-caps.c

├── ls-ecaps.c

├── ls-kernel.c

├── ls-map.c

├── lspci.c

├── lspci.h

├── lspci.man

├── ls-tree.c

├── ls-vpd.c

├── maint

│   ├── gen-zone

│   ├── release

│   └── release.pm

├── Makefile

├── pci.ids

├── pcilib.man

├── pciutils.h

├── pciutils.lsm

├── pciutils.spec

├── README

├── README.Windows

├── setpci.c

├── setpci.man

├── tests

│   ├── cap-address-xlation

│   ├── cap-debug-port

│   ├── cap-MSI-mapping

│   ├── cap-pci-af

│   ├── cap-pcie-1

│   ├── cap-pcie-2

│   └── PCI-X-bridges-and-domains

├── TODO

├── Tupfile

├── update-pciids.man

├── update-pciids.sh

└── win32

├── config.h

└── config.mk

5 directories, 81 files

Главная функция int main(), вход в программу в файле lspci.m:911.

/\* Main \*/

int

main(int argc, char \*\*argv)

{

int i;

char \*msg;

if (argc == 2 && !strcmp(argv[1], "--version"))

{

puts("lspci version " PCIUTILS\_VERSION);

return 0;

}

pacc = pci\_alloc();

pacc->error = die;

pci\_filter\_init(pacc, &filter);

while ((i = getopt(argc, argv, options)) != -1)

switch (i)

{

case 'n':

pacc->numeric\_ids++;

break;

case 'v':

verbose++;

break;

case 'b':

pacc->buscentric = 1;

break;

case 's':

if (msg = pci\_filter\_parse\_slot(&filter, optarg))

die("-s: %s", msg);

break;

case 'd':

if (msg = pci\_filter\_parse\_id(&filter, optarg))

die("-d: %s", msg);

break;

case 'x':

opt\_hex++;

break;

case 't':

opt\_tree++;

break;

case 'i':

pci\_set\_name\_list\_path(pacc, optarg, 0);

break;

case 'm':

opt\_machine++;

break;

case 'p':

opt\_pcimap = optarg;

break;

#ifdef PCI\_OS\_LINUX

case 'k':

opt\_kernel++;

break;

#endif

case 'M':

opt\_map\_mode++;

break;

case 'D':

opt\_domains = 2;

break;

#ifdef PCI\_USE\_DNS

case 'q':

opt\_query\_dns++;

break;

case 'Q':

opt\_query\_all = 1;

break;

#else

case 'q':

case 'Q':

die("DNS queries are not available in this version");

#endif

default:

if (parse\_generic\_option(i, pacc, optarg))

break;

bad:

fprintf(stderr, help\_msg, pacc->id\_file\_name);

return 1;

}

if (optind < argc)

goto bad;

if (opt\_query\_dns)

{

pacc->id\_lookup\_mode |= PCI\_LOOKUP\_NETWORK;

if (opt\_query\_dns > 1)

pacc->id\_lookup\_mode |= PCI\_LOOKUP\_REFRESH\_CACHE;

}

if (opt\_query\_all)

pacc->id\_lookup\_mode |= PCI\_LOOKUP\_NETWORK | PCI\_LOOKUP\_SKIP\_LOCAL;

pci\_init(pacc);

if (opt\_map\_mode)

map\_the\_bus();

else

{

scan\_devices();

sort\_them();

if (opt\_tree)

show\_forest();

else

show();

}

pci\_cleanup(pacc);

return (seen\_errors ? 2 : 0);

}

При входе в функцию в первую очередь определяются опции, переданные при запуске утилиты.

Как только опции распределены по глобальным переменным,

int verbose; /\* Show detailed information \*/

static int opt\_hex; /\* Show contents of config space as hexadecimal numbers \*/

struct pci\_filter filter; /\* Device filter \*/

static int opt\_tree; /\* Show bus tree \*/

static int opt\_machine; /\* Generate machine-readable output \*/

static int opt\_map\_mode; /\* Bus mapping mode enabled \*/

static int opt\_domains; /\* Show domain numbers (0=disabled, 1=auto-detected, 2=requested) \*/

static int opt\_kernel; /\* Show kernel drivers \*/

static int opt\_query\_dns; /\* Query the DNS (0=disabled, 1=enabled, 2=refresh cache) \*/

static int opt\_query\_all; /\* Query the DNS for all entries \*/

/\*\*\* Our view of the PCI bus \*\*\*/

struct pci\_access \*pacc;

struct device \*first\_dev;

static int seen\_errors;

происходит инициализация pacc = pci\_init(pacc), т.е. обращение к внутренней библиотеке (\lib\common.c). Далее вызов функции scan\_devices(); для выборки данных об устройствах.

Затем данные выводятся в консоль функцией show() или show\_forest() в зависимости от начального условия.

Спецификация к основным функциям исходного кода утилиты lspci:

|  |  |  |
| --- | --- | --- |
| Функция | Место в коде | Описание |
| Int main(int argc, char \*\*argv) | *lspci.c:911* | Вход в программу |
| struct pci\_access \*pci\_alloc(void) | *init.c:148* | Выделение памяти для struct pci\_access |
| void pci\_filter\_init(struct pci\_access \*, struct pci\_filter \*) | *filter.c:14* | Название устройства в переменную szBuf |
| void pci\_init(struct pci\_access \*a) | *init.c:167* | Инициализация параметров фильтра struct pci\_filter |
| void scan\_devices(void) | *lspci.c:146* | Сканирование устройств |
| void pci\_scan\_bus(struct pci\_access \*a) | *access.c:16* | Поиск метода сканирования и получение списка устрйоств: struct pci\_dev |
| struct device \*scan\_device(struct pci\_dev \*) | *lspci.c:112* | Считывание данных выбранного устройства и формирование структуры struct device |
| void show(struct device \*d) | *lspci.c:900* | Вывод всех устройств в консоль |
| void show\_device(struct device \*d) | *access.c:101* | Вывод параметров выбранного устройства. |
|  |  |  |
| void pci\_cleanup(struct pci\_access \*a) | *init.c:208* | Освобождение ресурсов |

Схема последовательности вызовов функций внутри главной функции int main(int argc, char \*\*argv):

Int main(int argc, char \*\*argv)

*lspci.c:911*

struct pci\_access \*pci\_alloc(void)

*init.c:148*

void pci\_set\_name\_list\_path(struct pci\_access \*a, char \*name, int to\_be\_freed)

*names-parse.c:244*

void\* malloc(size\_t \_Size)

*malloc.c:97*

void pci\_filter\_init(struct pci\_access \*a UNUSED, struct pci\_filter \*f)

*filter.c:14*

void pci\_init(struct pci\_access \*a)

*init.c:167*

void scan\_devices(void)

*lspci.c:146*

void pci\_scan\_bus(struct pci\_access \*a)

*access.c:16*

struct device \*scan\_device(struct pci\_dev \*p)

*lspci.c:112*

int config\_fetch(struct device \*d, unsigned int pos, unsigned int len)

*lspci.c:83*

int pci\_filter\_match(struct pci\_filter \*f, struct pci\_dev \*d)

*filter.c:107*

int pci\_read\_block(struct pci\_dev \*, int pos, u8 \*buf, int len)

*access.c:101*

void pci\_setup\_cache(struct pci\_dev \*, u8 \*cache, int len)

*access.c:175*

int pci\_fill\_info(struct pci\_dev \*, int flags)

*pci.h:157*

void show(struct device \*d)

*lspci.c:900*

void show\_device(struct device \*d)

*access.c:101*

void show\_machine(struct device \*d)

*lspci.c:820*

void get\_subid(struct device \*d,word \*subvp,word\*subdp)

*lspci.c:261*

void char \*pci\_lookup\_name(struct pci\_access \*a, char \*buf, int size, int flags, ...) (struct device \*d)

*names.c:110*

int printf (char const\* const Format, ...)

*stdio.h:944*

void show\_kernel\_machine(struct device \*d)

*ls-kernel.h:162*

char \*find\_driver(struct device \*d, char \*buf)

*ls-kernel.h:109*

char \*pci\_get\_param(struct pci\_access \*, char \*)

*params.c:15*

char \*snprintf(char\*, size\_t, char \*, ...)

*stdio.h:1932*

void show\_verbose(struct device \*d)

*lspci.c:640*

Word get\_conf\_word(struct device \*d, unsigned int pos)

*lspci.c:180*

Byte get\_conf\_byte(struct device \*d, unsigned int pos)

*lspci.c:173*

int printf (char const\* const Format, ...)

*stdio.h:944*

void show\_terse(struct device \*d)

*lspci.c:281*

void show\_kernel(struct device \*d)

*ls-kernel.c:140*

Void show\_forest(void)

*ls-tree.c:249*

Void pci\_cleanup(struct pci\_access \*a)

*init.c:208*

Void pci\_free\_dev(struct pci\_dev \*d)

*access.c:57*

Void pci\_mfree(void \*x)

*init.c:72*

Ниже приведен упрощенный пример получения списка устройств pci в своей программе:

#include <stdio.h>

#include "lib/pci.h"

int main(void)

{

struct pci\_access \*pacc;

struct pci\_dev \*dev;

unsigned int c;

char namebuf[1024], \*name;

pacc = pci\_alloc(); /\* Get the pci\_access structure \*/

/\* Set all options you want -- here we stick with the defaults \*/

pci\_init(pacc); /\* Initialize the PCI library \*/

pci\_scan\_bus(pacc); /\* We want to get the list of devices \*/

for (dev=pacc->devices; dev; dev=dev->next) /\* Iterate over all devices \*/

{

pci\_fill\_info(dev, PCI\_FILL\_IDENT | PCI\_FILL\_BASES | PCI\_FILL\_CLASS); /\* Fill in header info we need \*/

c = pci\_read\_byte(dev, PCI\_INTERRUPT\_PIN); /\* Read config register directly \*/

printf("%04x:%02x:%02x.%d vendor=%04x device=%04x class=%04x irq=%d (pin %d) base0=%lx",

dev->domain, dev->bus, dev->dev, dev->func, dev->vendor\_id, dev->device\_id,

dev->device\_class, dev->irq, c, (long) dev->base\_addr[0]);

/\* Look up and print the full name of the device \*/

name = pci\_lookup\_name(pacc, namebuf, sizeof(namebuf), PCI\_LOOKUP\_DEVICE, dev->vendor\_id, dev->device\_id);

printf(" (%s)\n", name);

}

pci\_cleanup(pacc); /\* Close everything \*/

return 0;

}

Используемые источники

1. https://www.opennet.ru/man.shtml?topic=lspci&category=8&russian=0
2. <http://rus-linux.net/kos.php?name=/papers/hwmon/hwmon.koi>
3. <http://prefetch.net/articles/linuxpci.html>
4. <https://www.cl.cam.ac.uk/cgi-bin/manpage?8+lspci>
5. http://www.xml.com/ldd/chapter/book/ch15.html#t1