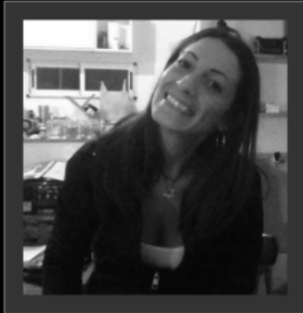


# Para programar, primero entiende a tu ordenador

*Publicado originalmente el 6 de Marzo de 2012*



## **Eugenia Bahit**

GLAMP Hacker y Programadora eXtrema, avocada a la docencia en el ámbito de la Ingeniería de Software y a la investigación de nuevas técnicas, metodologías y paradigmas relacionadas con la seguridad de aplicaciones. Miembro de la Free Software Foundation y The Linux Foundation.

**¿De verdad crees que conoces con exactitud la arquitectura de tu ordenador? Si quieres programar, lo mejor que puedes hacer, es conocer con precisión, como está formado tu ordenador y que función exacta cumple cada componente. En este artículo, intentaremos "describir" virtualmente nuestro ordenador, para poder entenderlo, ayudándonos del comando `lshw`, al cual podremos acceder como super-usuario.**

## **Arquitectura de computadoras: conociendo el hardware**

Un ordenador, con respecto al hardware, se encuentra compuesto por una serie de dispositivos, clasificados según la función que éstos desempeñen. Dicha clasificación, se compone de:

- Dispositivos de entrada
- Dispositivos de salida
- Dispositivos de comunicación
- Dispositivos de almacenamiento
- Dispositivos de cómputo

```
root@cocochito:~# lshw -short
```

Bus info	Device	Class	Description
=====			
		system	To Be Filled By O.E.M.
		bus	775i65G.
		memory	64KiB BIOS
cpu@0		processor	Intel(R) Celeron(R) CPU 2.80GHz
		memory	16KiB L1 cache
		memory	256KiB L2 cache
		memory	2GiB System Memory
		memory	1GiB DIMM SDRAM Synchronous
		memory	1GiB DIMM SDRAM Synchronous
pci@0000:00:00.0		bridge	82865G/PE/P DRAM Controller/Host-Hub Interface
pci@0000:00:01.0		bridge	82865G/PE/P PCI to AGP Controller
pci@0000:01:00.0		display	NV34 [GeForce FX 5500]
pci@0000:00:06.0		generic	82865G/PE/P Processor to I/O Memory Interface
pci@0000:00:1d.0		bus	82801EB/ER (ICH5/ICH5R) USB UHCI Controller #1
pci@0000:00:1d.1		bus	82801EB/ER (ICH5/ICH5R) USB UHCI Controller #2
pci@0000:00:1d.2		bus	82801EB/ER (ICH5/ICH5R) USB UHCI Controller #3
pci@0000:00:1d.3		bus	82801EB/ER (ICH5/ICH5R) USB UHCI Controller #4
pci@0000:00:1d.7		bus	82801EB/ER (ICH5/ICH5R) USB2 EHCI Controller
pci@0000:00:1e.0		bridge	82801 PCI Bridge
pci@0000:02:00.0	wlan0	network	RT2561/RT61 802.11g PCI
pci@0000:02:02.0		communication	SM56 Data Fax Modem
pci@0000:02:05.0	eth0	network	RTL-8139/8139C/8139C+
pci@0000:00:1f.0		bridge	82801EB/ER (ICH5/ICH5R) LPC Interface Bridge
pci@0000:00:1f.1	scsi1	storage	82801EB/ER (ICH5/ICH5R) IDE Controller
scsi@1:0.0.0	/dev/cdrom	disk	DVDRAM GSA-H42N
scsi@1:0.1.0	/dev/sda	disk	122GB Maxtor 6Y120L0
scsi@1:0.1.0,1	/dev/sda1	volume	109GiB EXT4 volume
scsi@1:0.1.0,2	/dev/sda2	volume	4805MiB Extended partition
	/dev/sda5	volume	4805MiB Linux swap / Solaris partition
pci@0000:00:1f.3		bus	82801EB/ER (ICH5/ICH5R) SMBus Controller
pci@0000:00:1f.5		multimedia	82801EB/ER (ICH5/ICH5R) AC'97 Audio Controller
usb@5:1.1	scsi2	storage	
scsi@2:0.0.0	/dev/sdb	disk	SCSI Disk

Los **dispositivos de entrada** son todos aquellos que permiten la entrada de datos a un ordenador. Estos dispositivos, son los que permiten al usuario interactuar con el ordenador. Ejemplos: teclado, mouse (ratón), micrófono, webcam, scanner, etc.

Los **dispositivos de salida**, son todos aquellos que permiten mostrar la información procesada por el ordenador. Ejemplos: monitor, impresora, auriculares, altavoces, etc.

Los **dispositivos de comunicación** son aquellos que permiten la comunicación entre dos o más ordenadores. Ejemplos: modem, router, placa de red, bluetooth, etc.

Los **dispositivos de almacenamiento**, son todos aquellos que permiten almacenar datos en el ordenador. Ejemplos: disco duro, pendrive, disket, CD, DVD, etc.

Los **dispositivos de cómputo**, son aquellos encargados de realizar las operaciones de control necesarias, sobre el resto de los dispositivos del ordenador. Estos dispositivos, se encuentran disponibles, en todos los ordenadores, y los mismos se describen a continuación.

## CPU

La **CPU** (Central Processing Unit - Unidad Central de Procesamiento), también llamada **procesador** o **microprocesador**, es un circuito microscópico que interpreta y ejecuta instrucciones. La CPU se ocupa del control y del proceso de datos en las computadoras. Generalmente, la CPU es un microprocesador fabricado en un chip, un único trozo de silicio que contiene millones de componentes electrónicos. El microprocesador de la CPU está formado por una unidad aritmético-lógica que realiza cálculos y comparaciones y toma decisiones lógicas (determinando si una afirmación es cierta o falsa mediante las reglas del álgebra de Boole. Para aceptar órdenes del usuario, acceder a los datos y presentar los resultados, la CPU se comunica a través de un conjunto de circuitos o conexiones llamado **bus**. El **bus** conecta la CPU a los dispositivos de almacenamiento (por ejemplo un disco duro), los dispositivos de entrada (por ejemplo un teclado o un mouse) y los dispositivos de salida (por ejemplo un monitor o una impresora).

```

root@cocochito:~# lshw -C bus -short
H/W path          Device          Class            Description
=====
/0
/0/100/1d         bus             775i65G.
#1
/0/100/1d.1      bus             82801EB/ER (ICH5/ICH5R) USB UHCI Controller
#2
/0/100/1d.2      bus             82801EB/ER (ICH5/ICH5R) USB UHCI Controller
#3
/0/100/1d.3      bus             82801EB/ER (ICH5/ICH5R) USB UHCI Controller
#4
/0/100/1d.7      bus             82801EB/ER (ICH5/ICH5R) USB2 EHCI Controller
/0/100/1f.3      bus             82801EB/ER (ICH5/ICH5R) SMBus Controller

```

Cuando se ejecuta un programa, el registro de la CPU, llamado contador de programa, lleva la cuenta de la siguiente instrucción del programa, para garantizar que las instrucciones se ejecuten en la secuencia adecuada.

La unidad de control de la CPU coordina y temporiza las funciones de la CPU, tras lo cual recupera la siguiente instrucción desde la memoria. En una secuencia típica, la CPU localiza la instrucción en el dispositivo de almacenamiento correspondiente. La instrucción viaja por el bus desde la memoria hasta la CPU, donde se almacena en el registro de instrucción. Entretanto, el contador de programa se incrementa en uno para prepararse para la siguiente instrucción.

A continuación, la instrucción actual es analizada por un decodificador, que determina lo que hará la instrucción. Cualquier dato requerido por la instrucción es recuperado desde el dispositivo de almacenamiento correspondiente y se almacena en el registro de datos de la CPU.

Luego, la CPU ejecuta la instrucción y, los resultados se almacenan en otro registro o

se copian en una dirección de memoria determinada<sup>2</sup>.

## Memoria

La **memoria** es la encargada de almacenar toda la información que el ordenador se encuentra utilizando. Existen tres tipos de memoria: **memoria RAM**, **memoria ROM** y **memoria caché**.

```

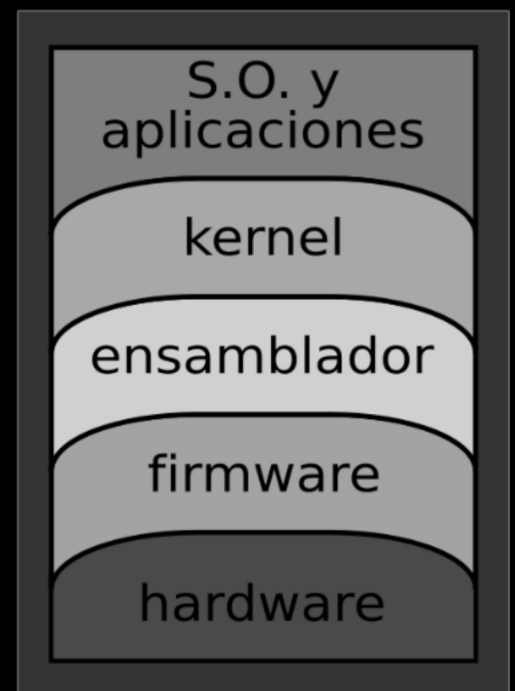
root@cocochito:~# lshw -C memory -short
H/W path          Device          Class          Description
=====
/0/0              memory          64KiB BIOS
/0/4/5            memory          16KiB L1 cache
/0/4/6            memory          256KiB L2 cache
/0/e              memory          2GiB System Memory
/0/e/0            memory          1GiB DIMM SDRAM Synchronous
/0/e/1            memory          1GiB DIMM SDRAM Synchronous

```

*Listando componentes de memoria*

## Memoria RAM

La **memoria RAM** (Random Access Memory), es la memoria desde la cual, el procesador recibe las instrucciones y guarda los resultados. Se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan "de acceso aleatorio" (random access) porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible. Durante el encendido del ordenador, la rutina POST verifica que los módulos de memoria RAM estén conectados de manera correcta. En el caso que no existan o no se detecten los módulos, la mayoría de tarjetas madres (**[motherboard](#)**) emiten una serie de pitidos que indican la ausencia de memoria principal.



<sup>2</sup> Fuente: [http://es.wikiversity.org/wiki/Estructura\\_del\\_computador](http://es.wikiversity.org/wiki/Estructura_del_computador)

Terminado ese proceso, la memoria BIOS<sup>3</sup> (memoria ROM) puede realizar un test básico sobre la memoria RAM indicando fallos mayores en la misma<sup>4</sup>.

```
root@cocochito:# lshw -C memory -short | grep -i ram
/0/e/0                memory          1GiB DIMM SDRAM Synchronous
/0/e/1                memory          1GiB DIMM SDRAM Synchronous
```

Filtrando la memoria RAM

## Memoria ROM

La **memoria ROM** (Read Only Memory), es permanente, ya que lo que permanece en la ROM no se pierde aunque el ordenador se apague. Su función principal es guardar información inicial que el ordenador necesita para colocarse en marcha una vez que se enciende. Permite solo la lectura de la información y no su escritura, independientemente de la presencia o no de una fuente de energía. Los datos almacenados en la ROM no se pueden modificar, o al menos no de manera rápida o fácil. Se utiliza principalmente para contener el **firmware** (programa que está estrechamente ligado a hardware específico, y es poco probable que requiera actualizaciones frecuentes) u otro contenido vital para el funcionamiento del dispositivo, como los programas que ponen en marcha el ordenador y realizan los diagnósticos<sup>5</sup>.

```
root@cocochito:# lshw -C memory -short | grep -i bios
/0/0                memory          64KiB BIOS
```

Filtrando la memoria ROM (también llamada BIOS)

## Memoria Caché

La **memoria caché** es aquella que se usa como puente entre el CPU y la memoria RAM para evitar demoras en el procesamiento de los datos. Existen varios núcleos de esta memoria (denominados con la letra L y un número, por ejemplo L1). Cuanto menor el número más rápida es la memoria. Por proximidad a la CPU, es mucho más rápida que la memoria RAM y también, mucho más pequeña.

---

3 En este caso, el término BIOS hace referencia a la memoria ROM (descrita anteriormente), y NO, al [Sistema Básico de Entrada y Salida](#).

4 Fuente: [http://es.wikipedia.org/wiki/Memoria\\_RAM](http://es.wikipedia.org/wiki/Memoria_RAM)

5 Fuente: [http://es.wikipedia.org/wiki/Memoria\\_ROM](http://es.wikipedia.org/wiki/Memoria_ROM)



```
root@cocochito:~# lshw -C memory -short | grep cache
/0/4/5                memory          16KiB L1 cache
/0/4/6                memory          256KiB L2 cache
```

*Filtrando la memoria caché*

## Bus de datos

El **bus de datos** (o canal de datos) es un sistema digital que transfiere datos entre los componentes de una computadora o entre computadoras. Está formado por cables o pistas en un circuito impreso, dispositivos como resistores y condensadores además de circuitos integrados.

### Notas adicionales sobre el comando `lshw`

Como hemos podido notar, el comando `lshw` nos permite **listar el hardware de nuestro equipo**. En este artículo, hemos utilizado el comando `lshw` con las siguientes opciones:

- **-short**  
Utilizada para listar el hardware en formato compacto.
- **-C tipo\_de\_componente**  
Utilizada para filtrar la salida, por el tipo de componente indicado.

Otras opciones pueden conocerse mediante `man lshw`.

También hemos concatenado (con `|`) el comando `lshw` con el comando `grep patrón` a fin de que la salida de `lshw` sea a la vez, filtrada por el patrón indicado. Cuando utilizamos la opción `-i` del comando `grep`, lo hicimos para que la búsqueda no distinga entre mayúsculas y minúsculas.