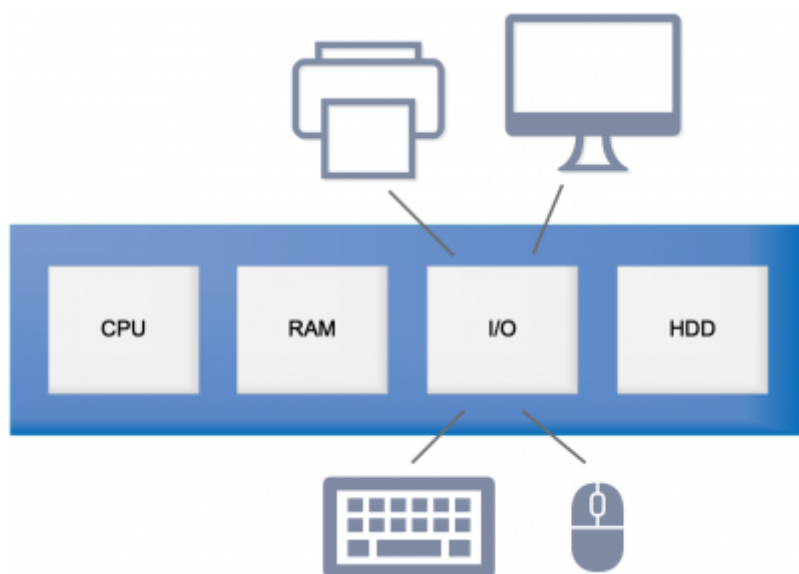


Gestión del Hardware

Todo sistema operativo requiere de un “hierro” que lo sustente y que le permita llevar a cabo las operaciones que el usuario le solicite. Es por ello que debemos conocer los elementos hardware necesarios para hacer funcionar un equipo informático, y de qué manera los podemos administrar y/o consultar.

Elementos Hardware



CPU

Elemento de proceso de las operaciones solicitadas. Es el “motor” del equipo, el encargado de llevar a cabo las instrucciones que el sistema solicita y entrega los resultados obtenidos. Consta de 2 partes:

- ALU: Unidad aritmético-lógica. Encargada de realizar operaciones
- CU: Unidad de control. Encargada de gestionar las tareas y los datos necesarios

Algunas de las herramientas que podemos utilizar para consultar información relacionada con nuestra CPU se muestran a continuación:

- [/proc/cpuinfo](#)
- [uname](#)
- [lscpu](#)

RAM

La memoria RAM de nuestro equipo es el espacio total de almacenamiento de que dispone el sistema para tener tareas abiertas de forma simultánea. Buscando una comparación, es como la mesa de nuestro escritorio, que nos permite tener un número máximo de papeles a nuestra disposición para

interactuar con todos ellos de forma paralela.

Normalmente se tiende a pensar que cuanto más memoria RAM mayor rendimiento podremos sacar al sistema, pero es fundamental un dimensionado ajustado y coherente con el resto de dispositivos (mucha mesa para uno solo no siempre resulta productiva ...)

La memoria RAM se puede diferenciar en 2 tipos a su vez:

- Memoria principal: módulos de memoria RAM
- Memoria virtual (swap): espacio de disco reservado para intercambio de páginas de memoria

Algunas de las herramientas que nos permiten poder consultar y gestionar la memoria del sistema son las que se enumeran a continuación:

- [/proc/meminfo](#)
- [free](#)
- [swapon](#)
- [swapoff](#)
- [/proc/sys/kernel/swapiness](#)

Dispositivo de almacenamiento

Los [dispositivo de almacenamiento](#) son los encargados de almacenar la información que generamos (datos, programas, etc). Son los “cajones” de nuestra mesa, que definen el espacio disponible para almacenar cosas y que cuando queremos utilizar sacamos de ellos para ponerlos en la mesa (acá [RAM](#))

Existen diversos tipos de dispositivos de almacenamiento, en función de la tecnología de interconexión y comunicación que utilicen

- SCSI: *Small Computer System Interface*
- IDE: *Integrated Device Electronics*
- SATA: *Serial Advanced Technology Attachment*
- SSD: *Solid State Drive*
- USB: *Universal Serial Bus*

Algunas herramientas útiles para poder consultar información sobre nuestros dispositivos de almacenamiento son:

- [/sys/block/sda/queue/rotational](#) (“0” equivale a SSD, “1” a HDD)
- [df](#)
- [lspci](#)

Periféricos

Los dispositivos [periféricos](#) son todos aquellos componentes HW no esenciales para el funcionamiento del sistema pero que nos permiten poder hacer un uso de él con mayores comodidades o nos permite ampliar sus funciones. Desde algunos casi indispensables (teclado, monitor) hasta algunos cada vez menos presentes (impresora, escáner), los [periféricos](#) se pueden clasificar en 2 tipos:

- Coldplug: el sistema debe ser apagado para poder conectar el dispositivo.
- Hotplug: el sistema no necesita ser apagado para poder conectar el dispositivo.

Algunas herramientas que nos permiten poder consultar los dispositivos conectados al equipo son:

- [lspci](#)
- [lsusb](#)

Firmware

El [firmware](#) es ese elemento a medio caballo entre HW y SW. Técnicamente hablando es un componente SW, pero se encuentra alojado dentro del HW, y en muchas ocasiones no permite su modificación.

Este “programa” ayuda al [kernel](#) a reconocer los dispositivos conectados al equipo, e inicia el proceso de arranque del sistema. En este sentido, podemos diferenciar 2 grandes tipos de [firmware](#):

- [BIOS](#) (*Basic Input Output System*)
- [UEFI](#) (*Unified Extensible Firmware Interface*)

Recursos Hardware

Los dispositivos necesitan herramientas que les permitan la comunicación entre ellos. Es por esto que el sistema debe ofrecer la posibilidad de interconexión entre ellos a través de diferentes métodos. A continuación vemos los 4 [recursos hardware](#) que el sistema dispone para la operación de los dispositivos entre ellos.

IO ports

Posiciones de memoria reservadas para la comunicación con los dispositivos hardware Ver [/proc/ioports](#)

IO Memory

Espacio de memoria reservado para almacenar datos que han de ser enviados a los dispositivos hardware. Ver [/proc/iomem](#)

IRQ

Conjunto de señales disponibles para los dispositivos hardware que permiten comunicar con la [CPU](#) y solicitar atención. Ver [/proc/irq](#)

DMA

Técnica para acceder a memoria desde los dispositivos hardware sin solicitarlo previamente a la [CPU](#)
Ver </proc/dma>

Subsistemas

A parte de las herramientas que hemos ido mostrando, el sistema también dispone de una serie de [subsistemas](#) que le permiten reconocer e integrar en la operación del sistema a los dispositivos hardware conectados.

udev

Ver [udev](#)

sysfs

Ver [sys](#)

HALD

[HALD](#) son las siglas de *Hardware Abstraction Layer (HAL) daemon*, un servicio del sistema que nos permite recolectar información de los dispositivos hardware disponibles.

Cuando el [kernel](#) detecta un dispositivo, éste guarda dentro de la estructura de [/sys](#) la información relacionada con él. A partir de ese momento, [HALD](#) es el responsable de monitorizar los dispositivos conectados y obtener los atributos que éstos pongan a disposición del sistema para su uso y mantenimiento.

Desde principios de 2011 [HALD](#) ha quedado obsoleto y muchas distribuciones han optado por utilizar el subsistema [udev](#)

d-bus

Cuando un programa necesita información sobre algún dispositivo puede preguntar a **HALD** a través de un canal de comunicación entre procesos, lo que conocemos como **d-bus**.



d-bus es un mecanismo que permite la comunicación entre procesos a través de un medio compartido (IPC - *Inter Process Communication*). Esta técnica permite optimizar la comunicación entre procesos, evitando que se generen canales “p2p” entre procesos que provocan un entramado de canales complejo y que provoca ineficiencia y poca fiabilidad de las comunicaciones.

A parte de este bus al que todos los procesos tienen acceso, cada usuario registrado dispone de su propio bus de sesión, en el que están conectados todos los procesos asociados a ese usuario.

Por otra parte, **HALD** utiliza **d-bus** para enviar notificaciones cuando se producen cambios en el estado de los dispositivos conectados. Estas notificaciones llegan a todos los programas que han solicitado recibirlas (programas registrados).

From:

<https://wiki.deceroauno.net/> - **DE 0 A 1**

Permanent link:

https://wiki.deceroauno.net/doku.php?id=basics:gestion_hardware

Last update: **2020/12/02 19:24**

