

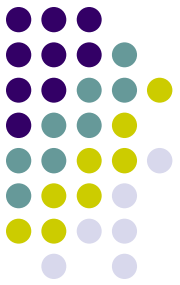
# **BIOS: Conociendo y reflexionando sobre la bestia**

Tomás Balderas Contreras  
INAOE

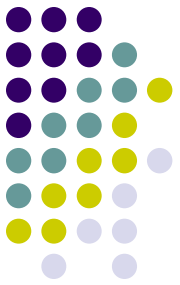
III Simposio de Software Libre de la Mixteca  
Octubre 2008



# I. INTRODUCCION



**¿Un sistema de cómputo con una  
instalación de GNU/Linux se  
encuentra libre de código  
propietario?**



**Respuesta: NO, el BIOS (firmware)  
es propiedad intelectual del  
fabricante del sistema**

# ¿Qué es el BIOS?

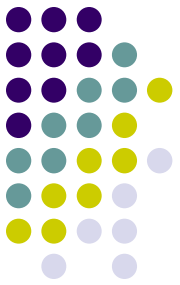


- Es una secuencia de instrucciones y estructuras de datos (programa).
- Se encuentra almacenado en la memoria Flash (EEPROM) de la computadora.
- Permite al usuario modificar los parámetros de configuración del sistema (SETUP) y los almacena en memoria no volátil (CMOS).
- Se encarga de inicializar los componentes del sistema de acuerdo al diseño del mismo (POST – Power On Self Test) y desplegar POST Codes.

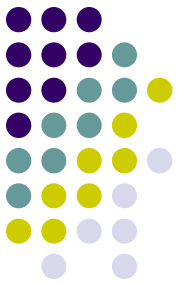
# ¿Qué es el BIOS?



- Inicializa la DRAM y el controlador de memoria a través del Memory Reference Code (MRC).
- Proporciona el control del sistema al sistema operativo.
- Proporciona servicios al sistema operativo en tiempo de ejecución.
- Proporciona una interfaz que aísla al sistema operativo y a los programas de aplicación de las particularidades de los dispositivos de hardware.

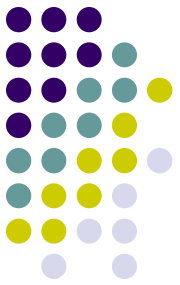


**¿Quién ha tenido la necesidad de  
compilar el núcleo Linux en su  
PC?**



**¿Quién ha tenido la necesidad de actualizar el BIOS de su PC?**

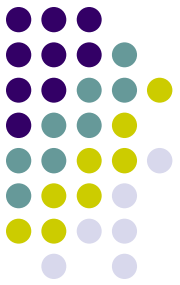




## **II. PLATAFORMAS BASADAS EN COMPONENTES DE INTEL**

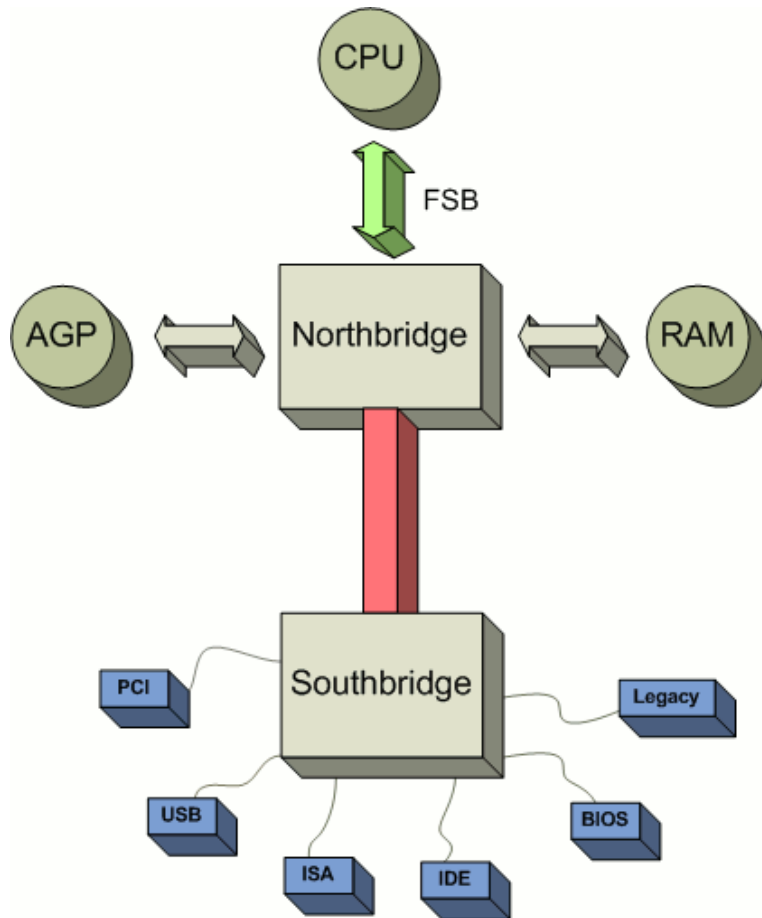


**¿Puede mencionar algunos nombres comerciales de procesadores de propósito general?**



**¿Puede mencionar algunos nombres comerciales de chipsets (MCH e ICH)?**

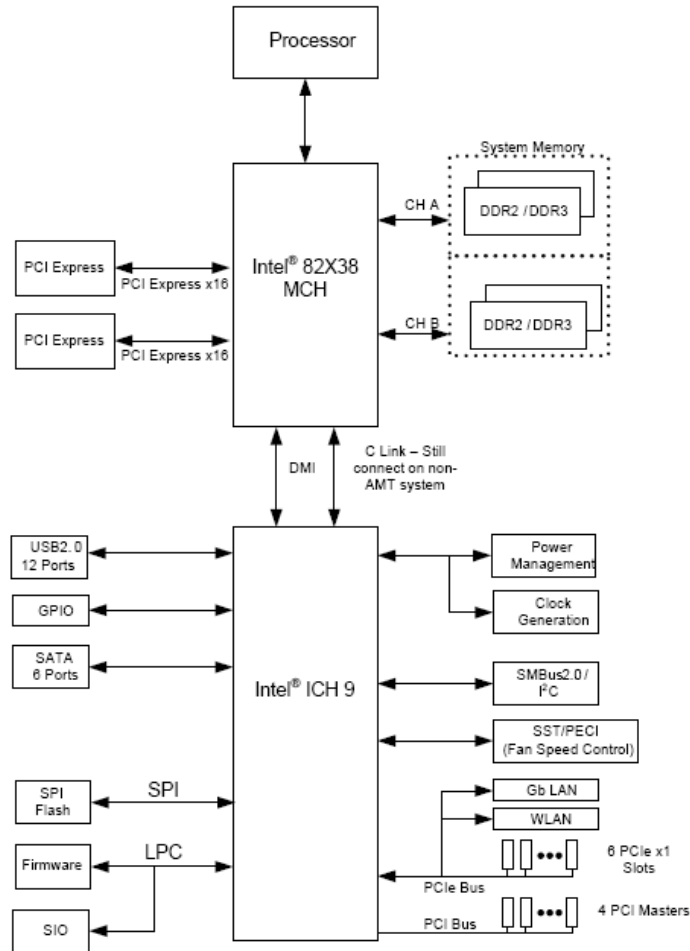
# Estructura genérica de una motherboard



Fuente: Wikipedia.com

- Chipset: Grupo de circuitos integrados diseñados para trabajar en conjunto y comercializados como un solo producto.
- El chipset es parte fundamental del funcionamiento de una computadora; es como el torso de la computadora.

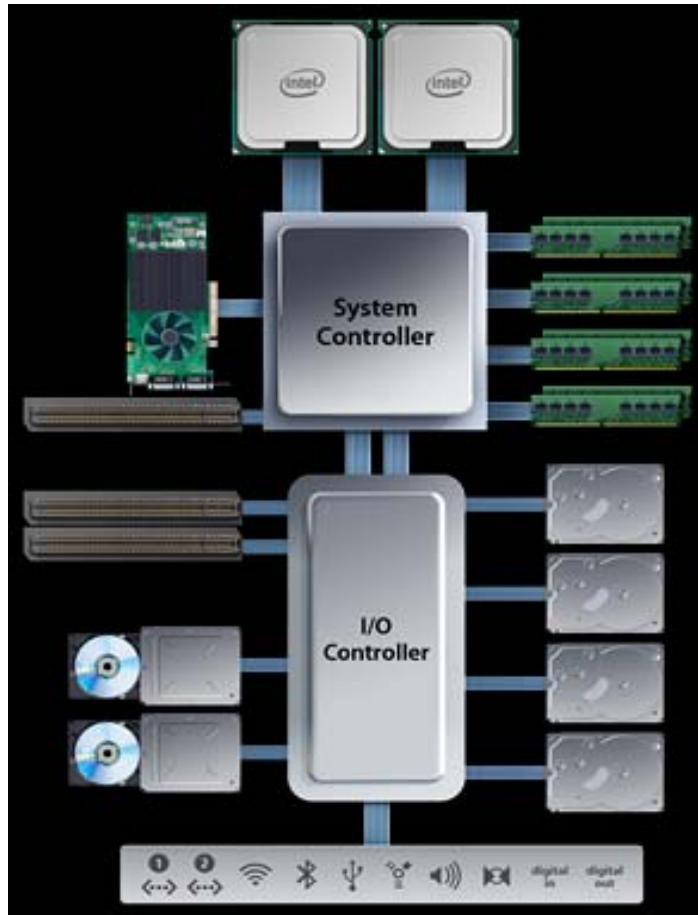
# Estructura actual de una plataforma Intel



- MCH (northbridge): Memory Controller Hub.
- ICH (southbridge): I/O Controller Hub.

Fuente: Intel X38 Express Chipset Datasheet

# Ejemplo: Mac Pro



Fuente: apple.com

- Quad-Core Intel Xeon “Harpertown” processors.
- Two FSBs running at 1333MHz.
- Four 800MHz DDR2 fully buffered DIMM channels.
- Server 5000X MCH (the “System Controller”).
- 321ESB ICH
- 6 dispositivos SATA .
- 2 dispositivos PATA.

# Tecnologías



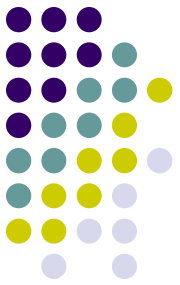
- Advanced Programmable Interrupt Controller (APIC).
- Advanced Configuration and Power Interface (ACPI):
  - Gestión de potencia.
  - Configuración y reconocimiento de hardware.
- Virtualización:
  - VTi.
  - VTd.
- Reliability, Availability and Serviceability (RAS):
  - Tolerancia a fallas.
  - Redundancia (mirroring, memory hot-plug, cpu hot-plug).
- Windows Hardware Error Architecture (WHEA).

# Itanium



- Explicitly Parallel Instruction Computing (EPIC).
- Explota el paralelismo a nivel de instrucción.
- Cómputo de 64 bits.
- RAS.
- Dedicado al mercado de cómputo de alto desempeño.





# III. IMPLEMENTACION



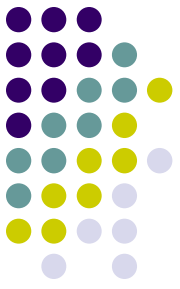
**¿En qué lenguaje cree usted que se desarrolla el BIOS (firmware)?**

# Desarrollo de BIOS

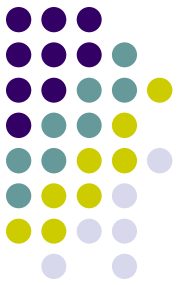


- Lenguaje ensamblador.
- AMIBIOS 8
  - Gestión de potencia (ACPI 3.0).
  - Opciones de booteo.
  - Buses de expansión: USB, PCI, PCIe.
  - Visual eBIOS
  - eModules.
  - Gestión de proyectos, control de versiones, IDE.
- Phoenix Technologies.
- Se ejecuta en modo compatible de 16 bits.

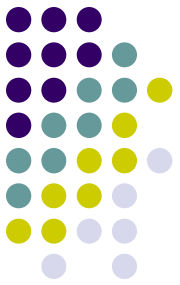
# Desarrollo de BIOS



- Unified Extensible Firmware Interface (UEFI).
  - Interfaz entre el sistema operativo y el firmware del sistema.
  - Incluye tablas con información sobre el sistema y servicios en tiempo de ejecución y de booteo disponibles para el cargador del sistema operativo y para el sistema operativo mismo.
  - Reemplazo moderno al BIOS.
  - Su desarrollo inició a la par de Itanium.
  - Contempla aplicaciones preboot.
- Tiano Core
  - Implementación de UEFI.
  - Permite el desarrollo en lenguaje C.
  - Entorno de desarrollo moderno con conceptos de alto nivel.

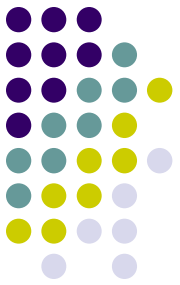


# **IV. EL MOVIMIENTO DE SOFTWARE LIBRE Y BIOS**



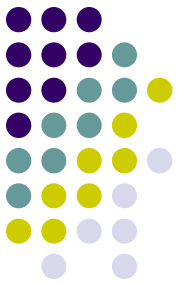
**Today the BIOS is no longer burned in ROM; it is stored in nonvolatile writable memory that users can rewrite. Today the BIOS sits square on the edge of the line. It comes prewritten in our computers, and normally we never install another. So far, that is just barely enough to excuse treating it as hardware. But once in a while the manufacturer suggests installing another BIOS, which is available only as an executable. This, clearly, is installing a non-free program--it is just as bad as installing Microsoft Windows, or Adobe Photoshop, or Sun's Java Platform. As the unethical practice of installing another BIOS executable becomes common, the version delivered inside the computer starts to raise an ethical problem issue as well.**

**Richard Stallman – Febrero 26, 2005  
Fuente: <http://www.fsf.org/news/freebios.html>**



**Not all of our community perceives the non-free BIOS as an acute problem. Much of our community supports the open source philosophy, which says that the issue at stake is choosing a development model that produces powerful, reliable software. The open source philosophy doesn't say that "closed source" software is unethical, only that it is likely not to be as reliable. People who hold those views might not care about the loss of freedom imposed by a non-free BIOS, because in their philosophy, freedom is not the issue. For us in the free software movement, freedom is the main issue; we have to solve this problem, whether they help or not.**

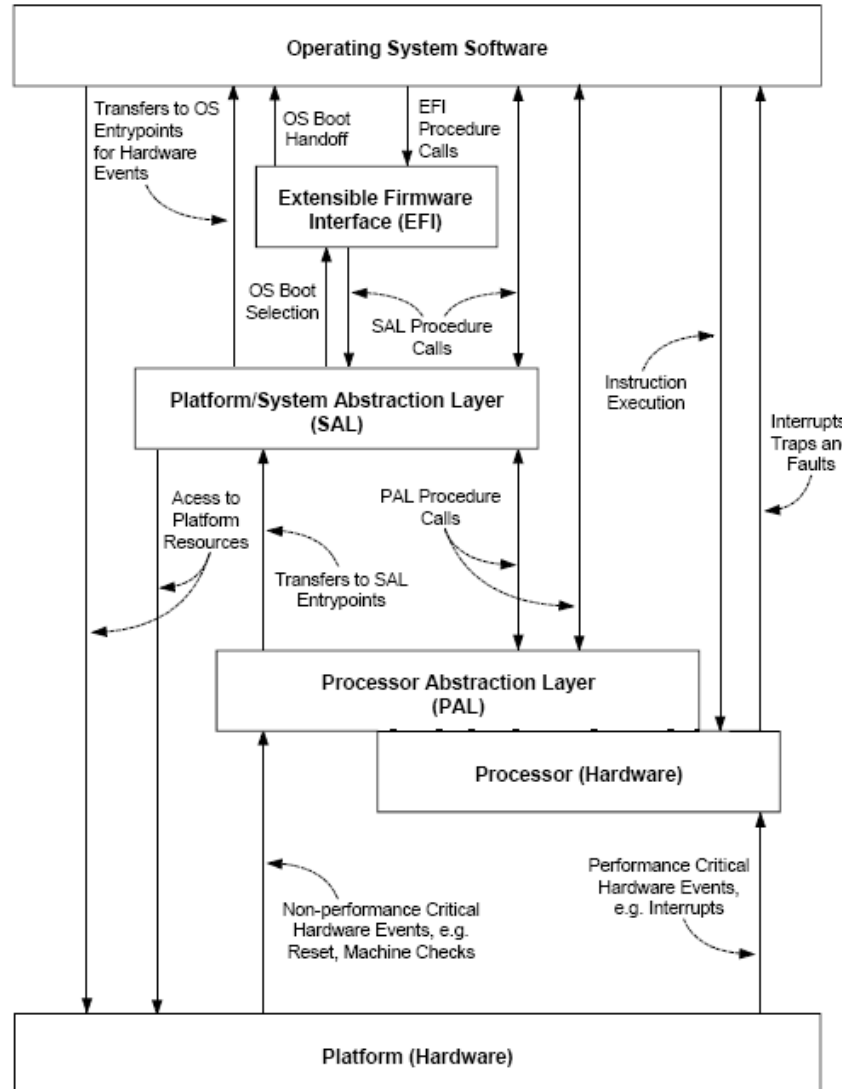
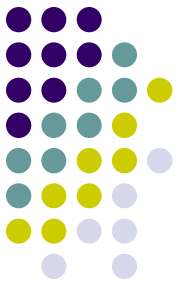
**Richard Stallman – Febrero 26, 2005  
Fuente: <http://www.fsf.org/news/freebios.html>**



**Según el punto de vista de  
Richard Stallman, ¿en qué frente  
se encuentra usted (free software  
u open source)?**



# ¿Qué hacer en el caso de sistemas Itanium?



Fuente: Intel Itanium  
Architecture Software  
Developer's Manuals  
Volume 2: System Architecture

# Alternativas

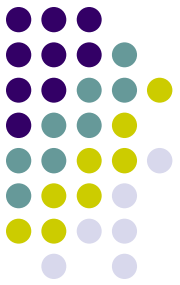


## Open Firmware

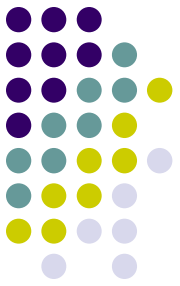
- Estándar que define las interfaces presentes en el firmware de una computadora.
- Es similar a EFI.
- Descrito por el estándar IEEE 1275-1994.
- Ha sido utilizado por Sun, Apple e IBM.
- Existen varias implementaciones libres:
  - Sun OpenBoot (licencia BSD).
  - Firmworks OpenFirmware.
  - Codegen SmartFirmware.
- Código fuente disponible a través del proyecto OpenBIOS (<http://www.openfirmware.info>).

## Coreboot

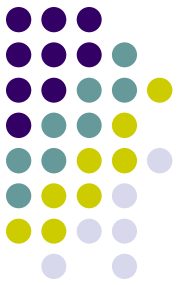
- Sistema de firmware ligero, lleva a cabo lo mínimo necesario antes de cargar un sistema operativo.
- Puede cargar el núcleo del sistema operativo o algún otro programa ejecutable en formato ELF, incluyendo cargadores de Windows (ADLO).
- Debe ser portado a diferentes chipsets y sistemas, lo cual puede ser muy complicado o imposible.
- No tiene soporte completo para ACPI.
- No proporciona servicios en tiempo de ejecución, Linux no los necesita.
- Patrocinado por FSF y algunas compañías; disponible bajo GPL.



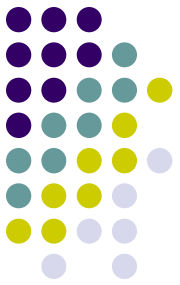
**¿Cuál es su posición frente al dilema del BIOS propietario?**



# V. CONCLUSIONES



- BIOS (firmware) es un componente crucial para el funcionamiento correcto de un sistema de cómputo.
- La complejidad del BIOS se ha incrementado en forma proporcional a la evolución de la tecnología de cómputo.
- El uso de un BIOS propietario plantea un dilema moral para muchos defensores del software libre.
- Dependiendo del fervor de la persona puede optar por ignorar el dilema o involucrarse en las soluciones al mismo.



# **BIOS: Conociendo y reflexionando sobre la bestia**

Tomás Balderas Contreras  
INAOE

**PREGUNTAS Y RESPUESTAS**