

# Conceptos básicos en arquitectura de GNU/Linux

Publicado el 01/02/2012 por Miguel Garcia

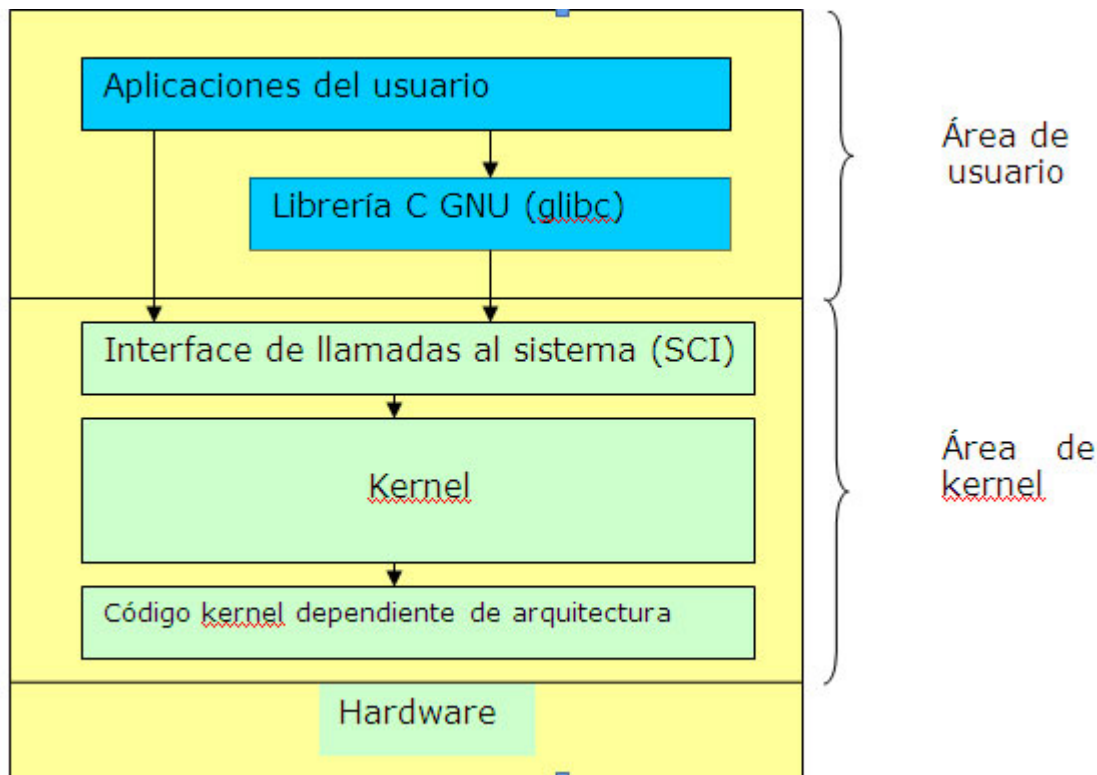
February 11, 2017

## 1 Estructura del sistema

El kernel (núcleo) es un componente del sistema que se sitúa por encima del hardware y provee toda una serie de abstracciones al resto de los programas. Veamos algunas de esas abstracciones:

- pipes
- archivos
- procesos
- threads

La misión del kernel es hacer funcionar la maquina por medio de la Gestión de procesos, gestión de memoria, gestión de acceso al hardware y gestión del sistema de archivos



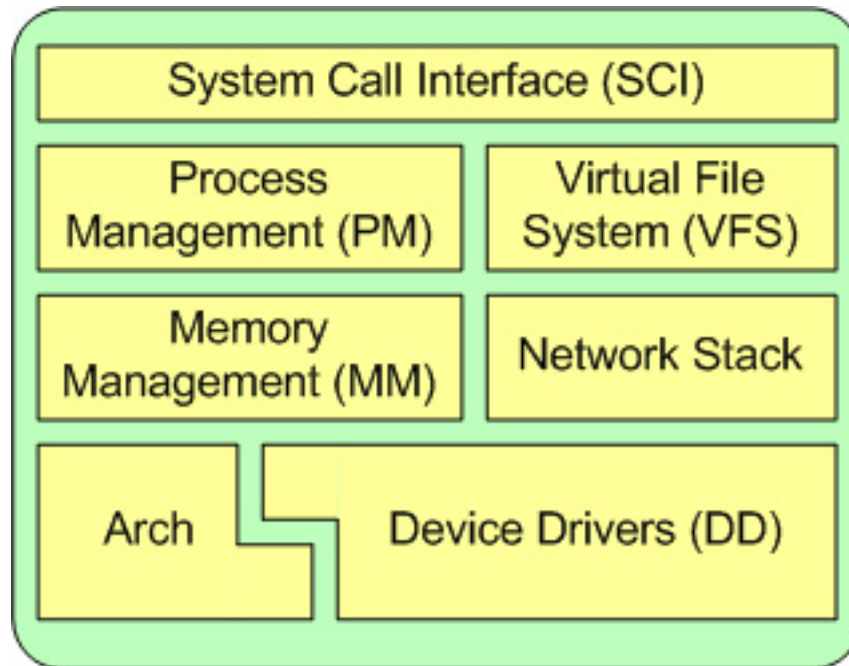
En la imagen, podemos ver en la parte superior el área donde corren los programas de usuario, también está la librería GNU/C que proporciona herramientas para las llamadas al sistema.

En el centro aparece lo que es básicamente el Kernel, que puede dividirse a su vez en tres áreas; en la parte superior la SCI que implementa las funciones básicas como pueden ser lectura o escritura por debajo, se encuentra el código del Kernel que puede ser definido como independiente del hardware, y como tal, común a todas las

implementaciones Linux. Mas abajo, se encuentra ya el código que depende de la arquitectura y como tal es distinto para cada tipo de procesador y arquitectura hardware.

En todos los niveles, el kernel se estratifica en una serie de subsistemas distintos, ofreciendo un gran numero de servicios. Es por ello que Linux se puede considerar monolítico, ya que agrupa en el kernel la mayoría de servicios, en contra de la filosofía de microkernels de otros S.O., en donde solo proporciona los servicios de mas bajo nivel, y los demás se han de ir montando fuera ya del kernel

## 2 Subsistemas del núcleo



### 2.1 Interfaz de la llamada del sistema (SCI)

El SCI es una delgada capa que proporciona los medios para realizar llamadas desde el área de usuario al núcleo, y en algunas ocasiones, puede ser también dependiente de la arquitectura. Por ejemplo, debido a la capacidad de virtualización que poseen los últimos procesadores x86.

### 2.2 Gestión de procesos (PM)

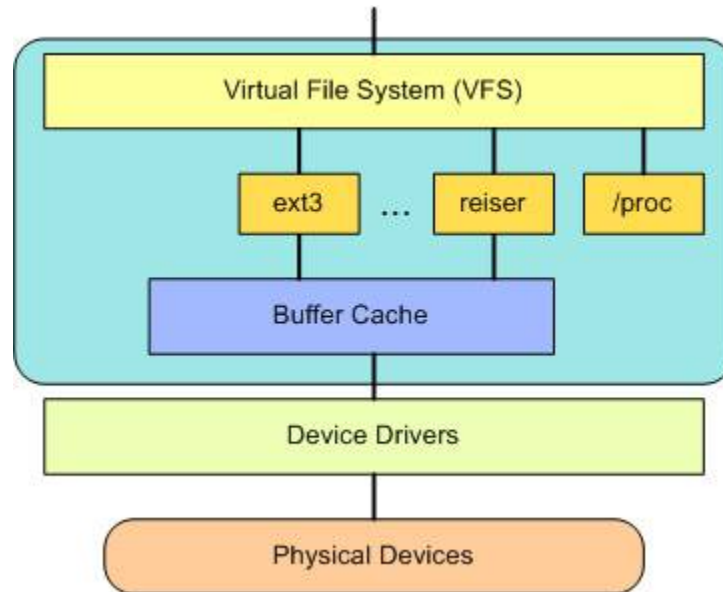
La gestión de procesos se centra en administrar la CPU para distribuir su tiempo entre cada uno de los procesos concurrentes que hay funcionando en un momento dado. Cada proceso o “thread” representa la virtualización de la CPU (código thread, datos, snack, registros CPU). El kernel, proporciona un API para que a través del SCI, se pueda crear un nuevo proceso, pararlo, y comunicarse con el. Todo esto es también necesario para compartir la CPU entre distintos procesos, para ello, se dispone de un algoritmo que distribuye el tiempo de la CPU entre todos los thread, a la vez que soporta la gestión de procesadores múltiples.

### 2.3 Gestión de memoria(MM)

Otro recurso que debe ser manejado por el kernel es la memoria. Por razones de eficiencia, la memoria se maneja en bloques (llamados paginas) de 4 Kb de tamaño en la mayoría de arquitecturas. Pero la gestión de memoria no es únicamente controlar esas paginas, también, y en función de su contenido crea esquemas de gestión que irían creciendo a medida que sea necesario y permitirá controlar la grabación de las paginas en disco (swap) cuando se agote la memoria física.

## 2.4 Sistema de ficheros virtuales (VFS)

El sistema de ficheros virtual en un interesante aspecto del kernel de Linux, ya que provee una abstracción acerca del sistema de ficheros



En la parte superior, tenemos la API con funciones como Open, Close, read, y write. Debajo se encuentra la abstracción del sistema de ficheros, y que establece como se implementan las funciones de la capa superior. Estos son plug-ins para el sistema de fichero dado(hay mas de 50).

Mas abajo, se encuentra el cache, que proporciona un sistema común de acceso independiente del sistema, y a la vez optimiza el acceso a los dispositivos fisicos, guardando la información durante un corto periodo de tiempo. Por ultimo esta los drivers del dispositivo que realizan el enlace con el dispositivo físico correspondiente.

## 2.5 Network stack

La pila de red sigue el mismo diseño por capas de los protocolos. Los sockets pertenecen al API y proporcionan el acceso desde la capa de usuario a una gran variedad de protocolos.

## 2.6 Controladores de dispositivos (DD)

Existe una gran profusión de programas para poder gestionar todo tipo de dispositivos. El sistema Linux proporciona un subdirectorio en donde se ubican todos los controladores (drivers) para cualquier dispositivo, como Bluetooth, puertos series, sonido,...

## 2.7 Código dependiente de la arquitectura.

Aunque la mayoría del núcleo de Linux tiende a ser independiente de la arquitectura en la que este instalado, hay partes del mismo, que por razones de eficiencia no puede ignorarla.Toda esta parte esta en un subdirectorio denominado genéricamente BSP que a su vez esta dividido en distintos subdirectorios para ocuparse de la parte que le corresponda, como Arranque, memoria.