



UNIVERSIDAD DE COSTA RICA

INTRODUCCIÓN A LINUX

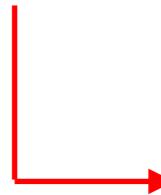
Prof. Marlon Brenes y Prof. Federico Muñoz
Escuela de Física, Universidad de Costa Rica

Porqué Linux en computación científica?

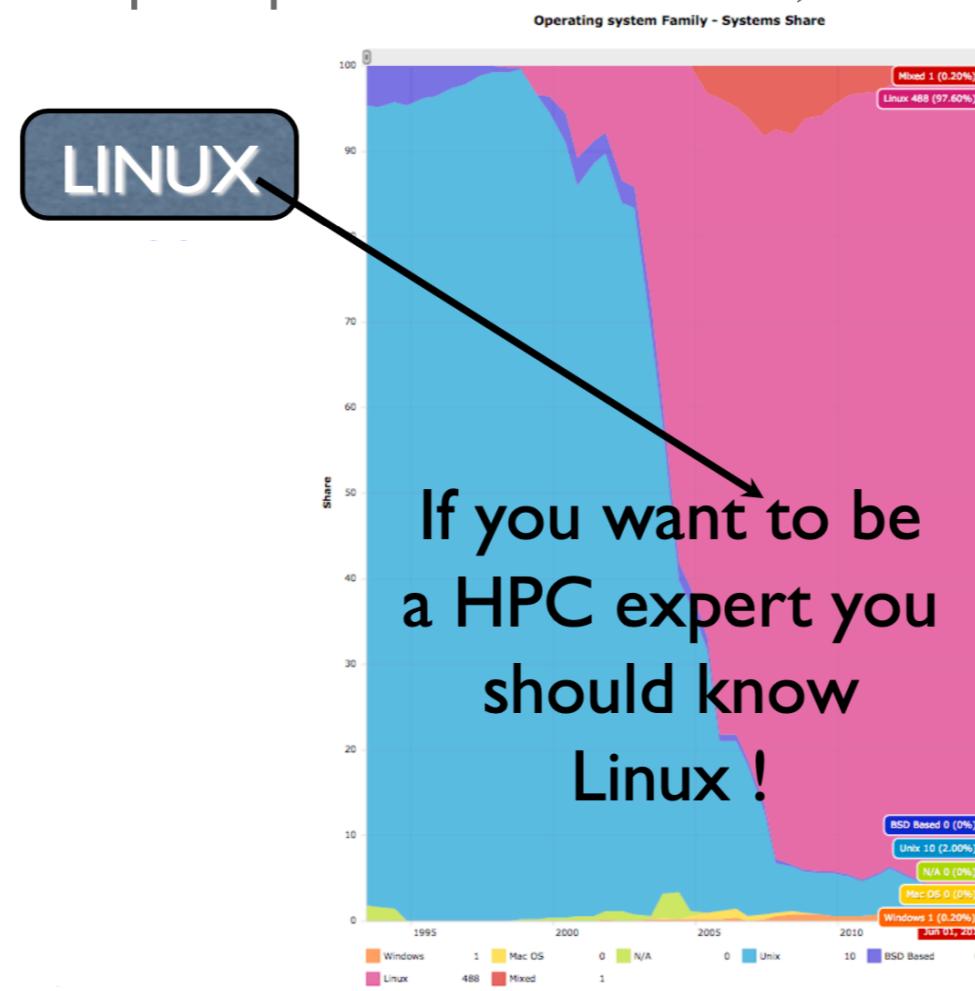
- Predominantemente por razones históricas
- Pero más importante: es un sistema operativo de fuente abierta
 - La esencia de la comunidad científica es accesibilidad, distribución y transparencia
 - Dichas cualidades son difíciles de ejercer con software propietario
- Hoy en día, Linux es mantenido y distribuido de manera comunitaria, con las distribuciones más populares (Ubuntu, centOS, entre otras) mantenidas profesionalmente por instituciones privadas

Computación de alto rendimiento

- La abrumante mayoría de sistemas de super cómputo son máquinas que operan con el sistema Linux
- Razones



- Históricas pero predominantemente, flexibilidad



Linux no es UNIX

- Se suelen confundir ambos sistemas operativos
 - GNU no es UNIX
 - Linux no es UNIX
- Linux es un sistema operativo basado en UNIX
- Existen muchas similitudes y Linux es “UNIX-compatible” pero
 - Linux no tiene una historia de comercialización
 - Flujo constante
- UNIX está “escrito en piedra”
 - Desarrollado comercialmente
 - No es gratuito

Características principales de Linux

- Portabilidad
 - └→ ● Software trabaja de la misma forma en distintas plataformas
 - Software se puede instalar en distintas plataformas
- Escalabilidad (*scalability*)
 - └→ ● De pocas a muchas unidades de procesamiento
- Fuente abierta (*open source*)
 - └→ ● Código disponible y desarrollo comunitario. Evolución continua
- Seguridad
 - └→ ● Encriptación, acceso controlado, protección autenticada
- Multi-acceso y multi-tarea
 - └→ ● Múltiples usuarios, múltiples aplicaciones ejecutando al mismo tiempo
- Jerarquía estructural de archivos
 - └→ ● Estructura estándar de archivos

Multi-acceso en Linux

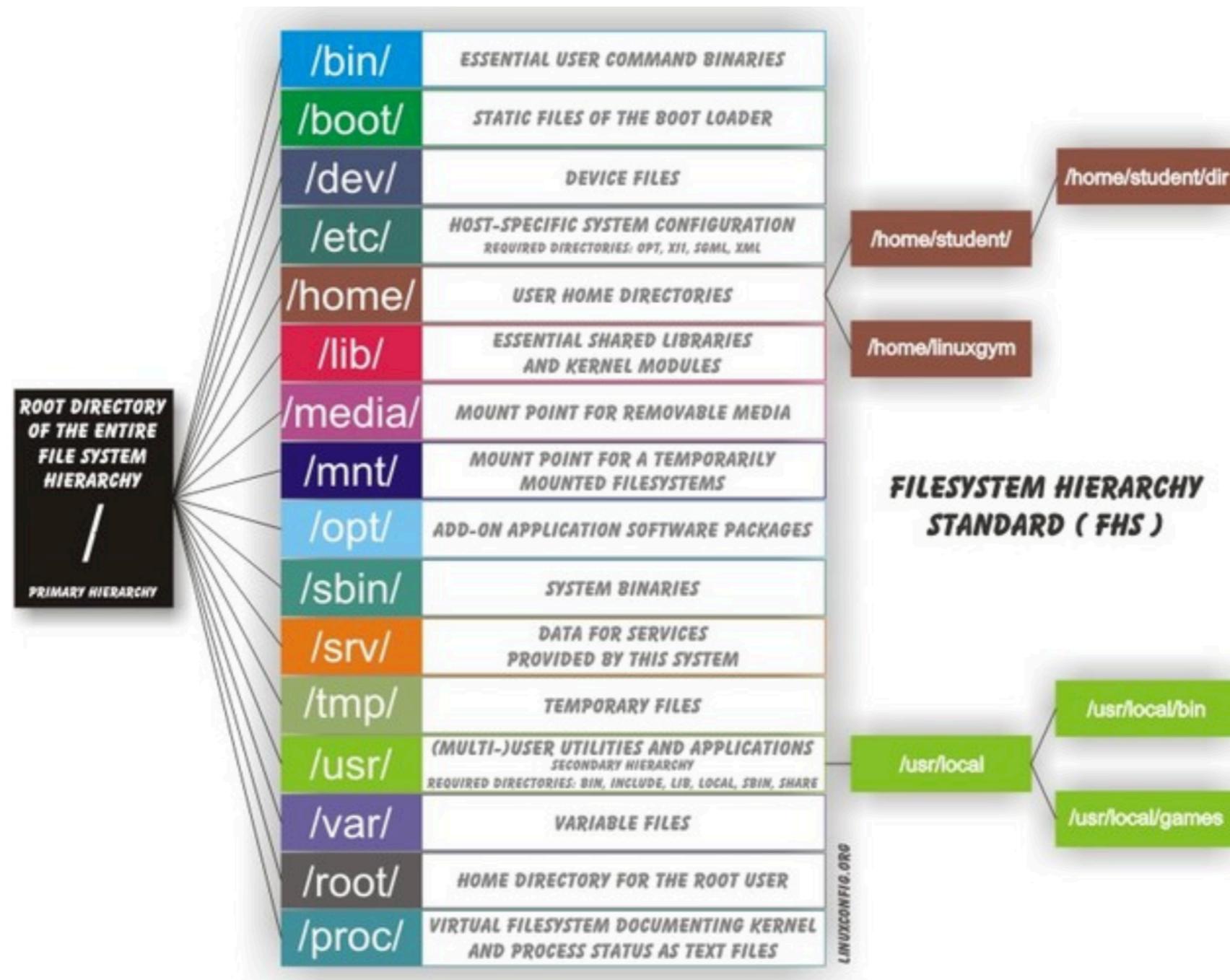
- Cada usuario tiene un nombre y un número (uid)
 - Un usuario puede estar en uno o más grupos (gid)
- Cada tarea o proceso tiene identificación única (pid) y tiene un propietario asignado
- Todos los archivos tienen como propietario un usuario y un grupo
 - Para un usuario o grupo, se requiere permisos para escribir, leer o ejecutar un archivo. Dichos permisos están señalados para cada archivo
- Los permisos se dan a usuario, grupo y “otros”
 - De esta manera los archivos se pueden proteger
- El usuario del sistema “root” tiene todos los permisos

Multi-tareas en Linux

- Un número de programas puede correr simultáneamente
 - • Dichos programas, conocidos como *procesos*, se pueden comunicar entre ellos
 - Por ejemplo, se puede compilar un programa en C mientras se edita un email
- Existen procesos internos que operan a ciertas frecuencias sin ser ejecutados directamente por un usuario
 - • Dichos procesos se llaman *daemons*
 - Se utilizan para distintas tareas; tales como sincronización de discos de memoria, correos, impresión, entre otras.

Sistema de archivos

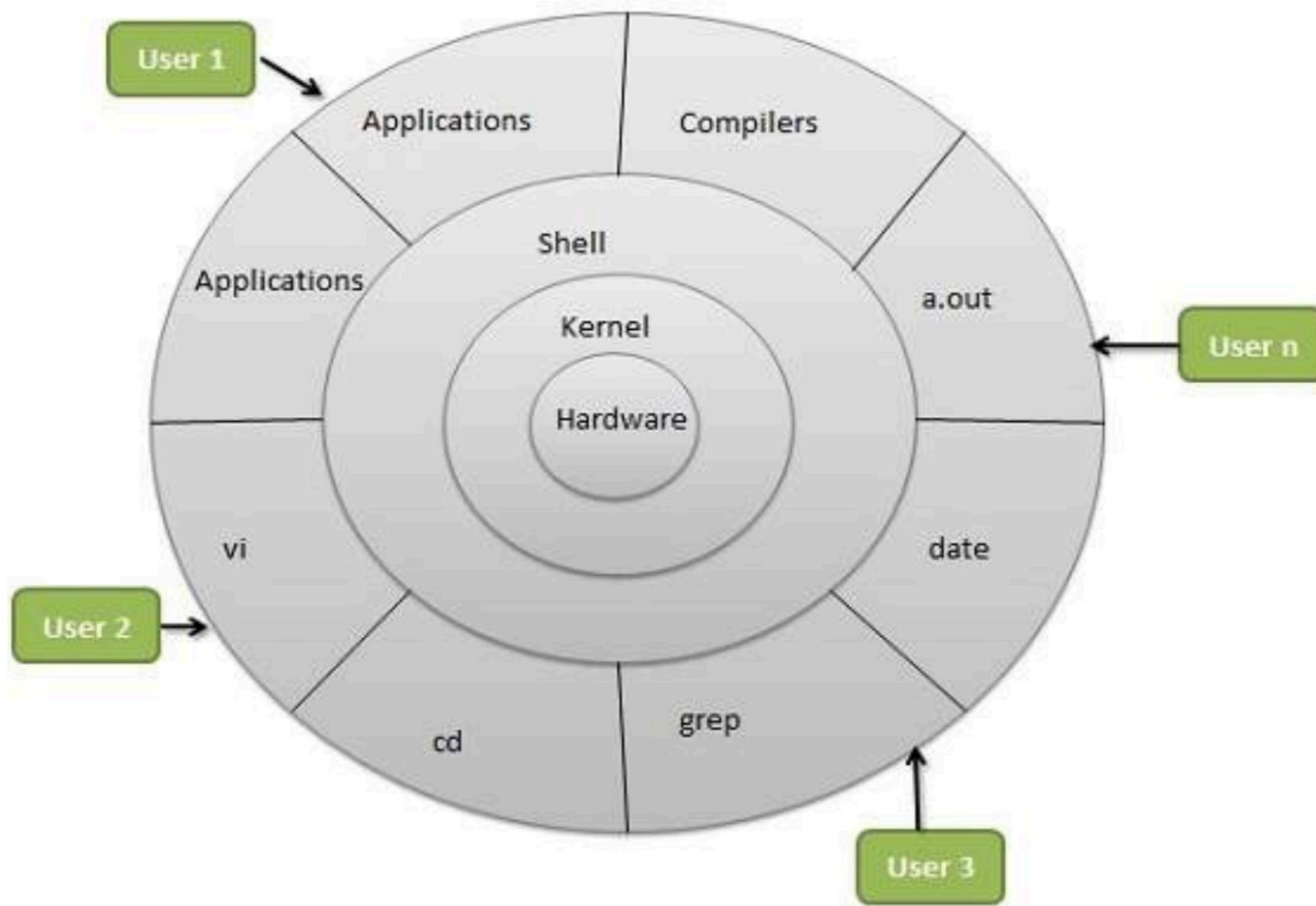
- Todos los archivos están organizados en un sistema jerárquico global. El nodo inicial se denota con el símbolo “/”



Arquitectura del sistema

- Hardware
 - CPU, RAM, HDD, entre otros
 - Kernel
 - El componente central del sistema operativo. Interactúa directamente con el hardware, provee la conexión entre el nivel más bajo (hardware) y las capas de más alto nivel
 - Shell
 - Interfaz al kernel. Esconde las complejidades de las funciones del kernel al usuario mediante la interfaz
 - Aplicaciones utilitarias
 - La mayor de las funcionalidades de un sistema operativo
- Existen diferentes shells. El shell de Linux (CLI) provee una interfaz de usuario para ejecutar comandos

Arquitectura del sistema



Linux es el Kernel



- Linux corresponde al kernel - la interfaz entre el nivel más bajo de la jerarquía (hardware) y el más alto (software)

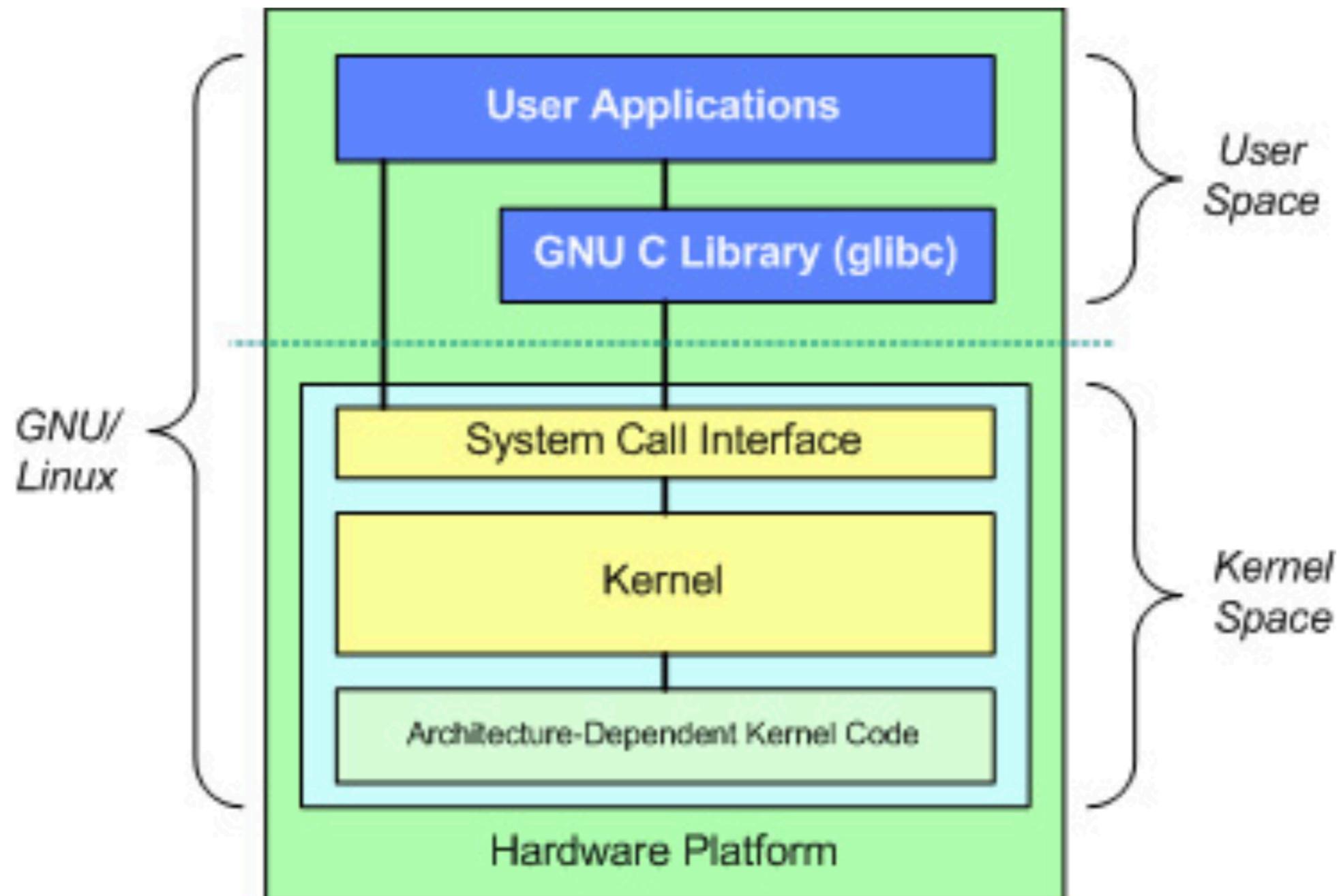
Distribución de Linux

- • Una distribución es una colección de aplicaciones empacadas junto con el kernel de Linux emparejado con bibliotecas relevantes
- • Una distribución está usualmente construida con base en un *sistema de manejo de paquetes*.
- • Hoy en día, la distribución se conoce como el “Linux flavour”. Cada distribución incluye aplicaciones desarrolladas para enfocar ciertas tareas
- • Ejemplos: Fedora, Ubuntu, Debian, Red Hat, SuSE
- • centOS es la distribución más adaptada a sistemas de super cómputo

Filosofía

- • En un sistema **UNIX (UNIX-based)** **todo** es un archivo. Si algo no es un archivo, es un **proceso**
- • Desarrollar programas pequeños que cumplen una tarea sencilla y bien definida
- • Todos los programas aceptan entrada del `stdin`, escriben resultados al `stdout` y reportan errores al `stderr`
- • Los *pipes* conectan el I/O de los programas secuencialmente (*plomería*) y la *data* fluye, en analogía, como el agua
- • Separación estricta entre el espacio de usuario y el espacio de kernel (espacio privilegiado)

Espacio de kernel y espacio de usuario



Arquitectura de Linux

Espacio de kernel

- El espacio de kernel está reservado para ejecutar operaciones privilegiadas del kernel y sus extensiones, además de la mayoría de los controladores de dispositivos

Espacio de usuario

- El espacio de usuario corresponde a la región lógica donde se ejecutan todas las aplicaciones del usuario y algunos controladores de dispositivos

- Los programas en espacio de usuario no pueden accesar los recursos directamente, de manera tal que el acceso se maneja directamente generando solicitudes al kernel
 - Esto usualmente ocurre mediante *llamados de sistema*

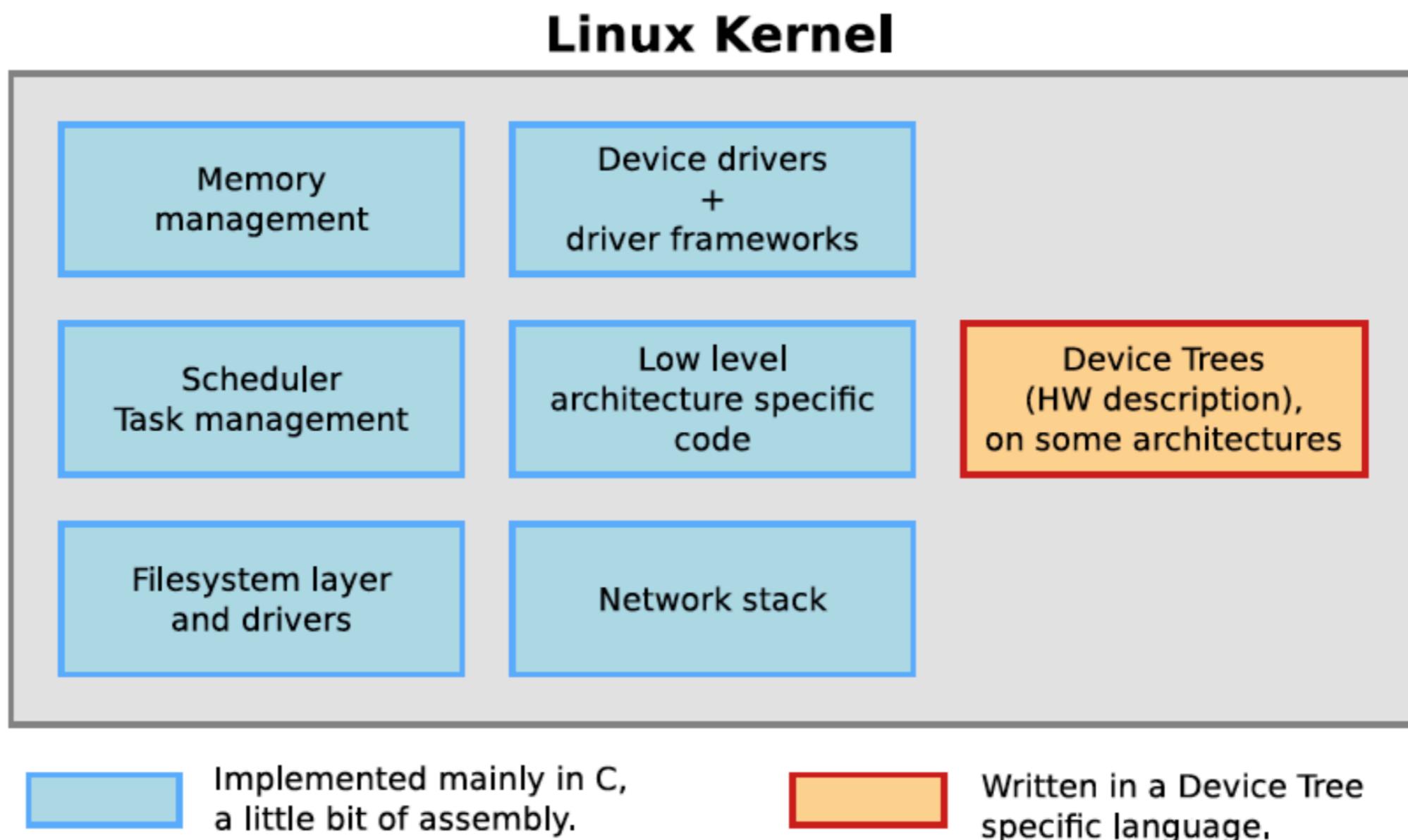
Tareas primordiales del kernel

- └→ • Manejo de todos los recursos del hardware: CPU, memoria, I/O
- └→ • Proveer APIs (*application programming interfaces*) que permiten a las aplicaciones en el espacio de usuario usar recursos de hardware
- └→ • Manejar el acceso concurrido y uso de recursos de hardware de las distintas aplicaciones

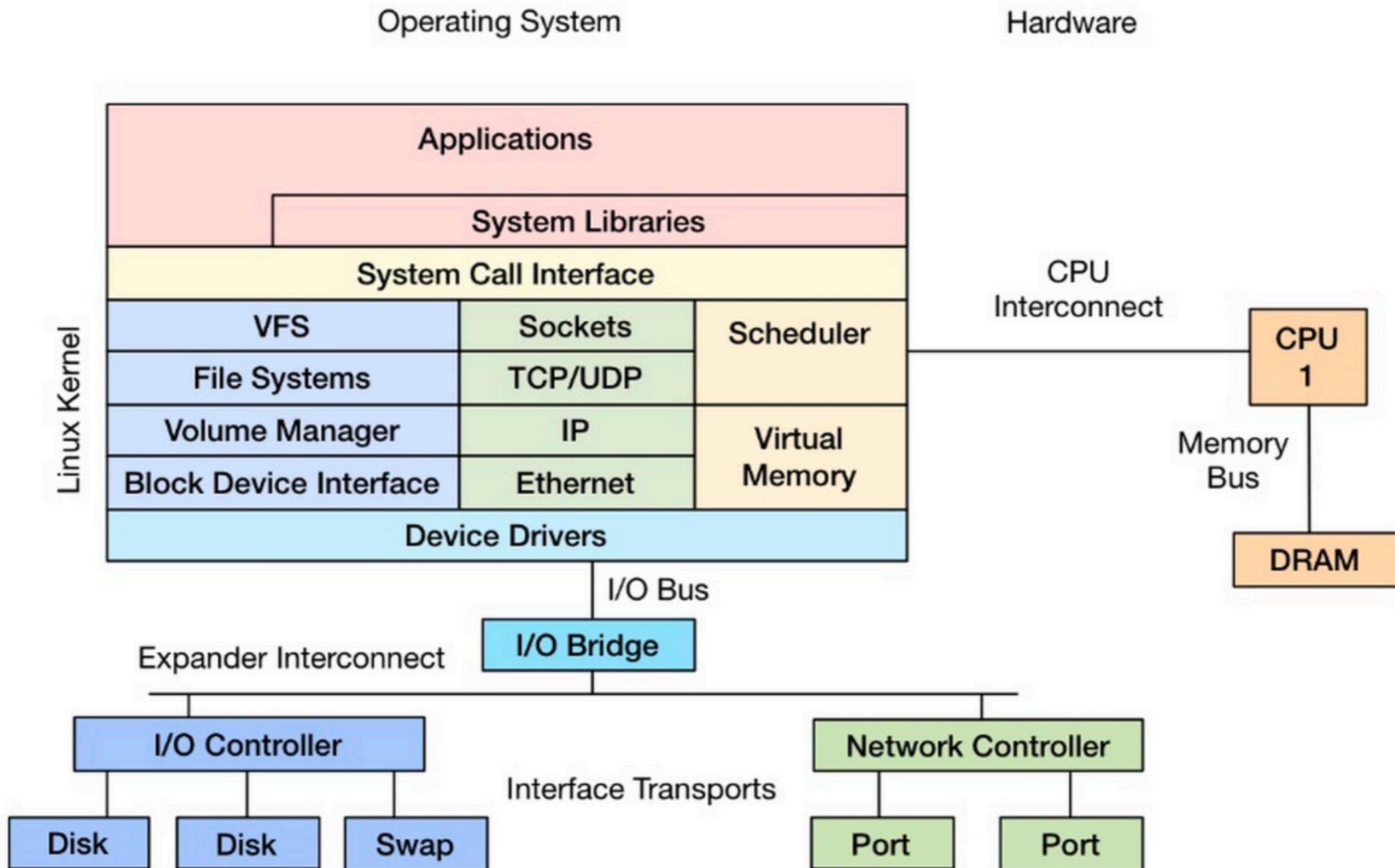
Llamados de sistema

- └→ • Corresponde al método de interfaz principal entre el kernel y el espacio de usuario
- └→ • Bajo estos llamados se proveen la mayoría de operaciones básicas:
 - └→ • Operaciones de archivos, dispositivos y redes, comunicación entre procesos, mapeo de memoria, hilos, temporizadores, entre otras
- └→ • La interface a los llamados de sistema están envueltos en la biblioteca de C. Las aplicaciones de espacio de usuario usualmente nunca hacen llamados de sistema directamente, si no mediante funciones de la biblioteca de C

Core del kernel de Linux



Core del kernel de Linux



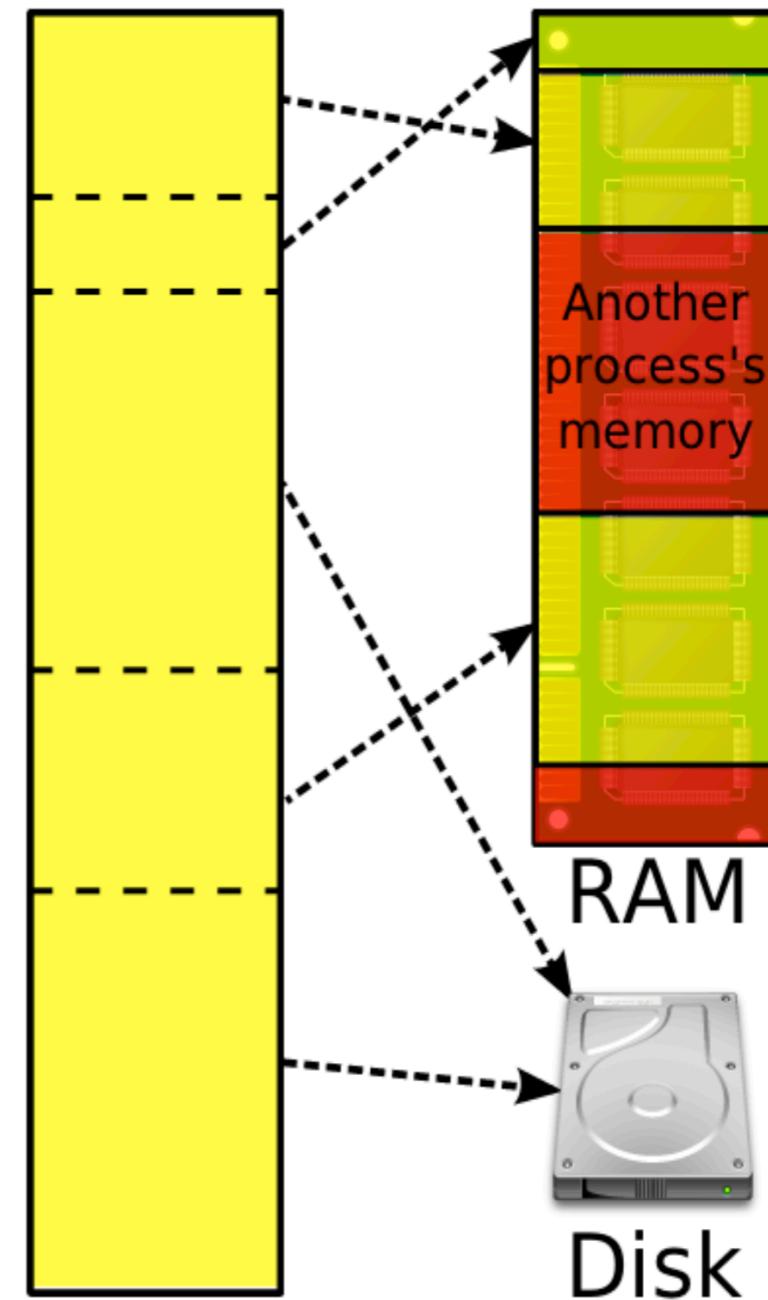
Pseudo-sistema de archivos

- En Linux la información del sistema y el kernel está disponible en el espacio de usuario mediante un pseudo-sistema de archivos
 - Los pseudo-sistemas de archivos permiten a las aplicaciones observar directorios y archivos que no están presentes en memoria directamente, si no que son creados y actualizados a demanda por medio del kernel
- Dos de los pseudo-sistemas de archivos más importantes son:
 - proc (montado en /proc): Información relacionada al sistema operativo (procesos, memoria, parámetros de manejo)
 - sysfs (montado en /sys): Representación del sistema como un conjunto de dispositivos y buses. Entrega información sobre éstos

`cat /proc/cpuinfo`
`cat /proc/meminfo`

Manejo de memoria

- El kernel traduce la dirección lógica de memoria virtual a la dirección física de los datos en memoria física
- Cada proceso tiene su propio espacio de dirección de memoria
- La memoria física debe ser mapeada a espacios lógicos de memoria
- Accesos a memoria fuera de su dirección incurren en un error muy común de memoria: “Segmentation fault”
- El kernel puede decir *intercambiar* espacios de memoria (*swaps*)



Recursos adicionales



- <http://linux-training.be/linuxfun.pdf>



- <https://akkadia.org/drepper/cpumemory.pdf>