

# Sistemas de Computación

2do Semestre 2010

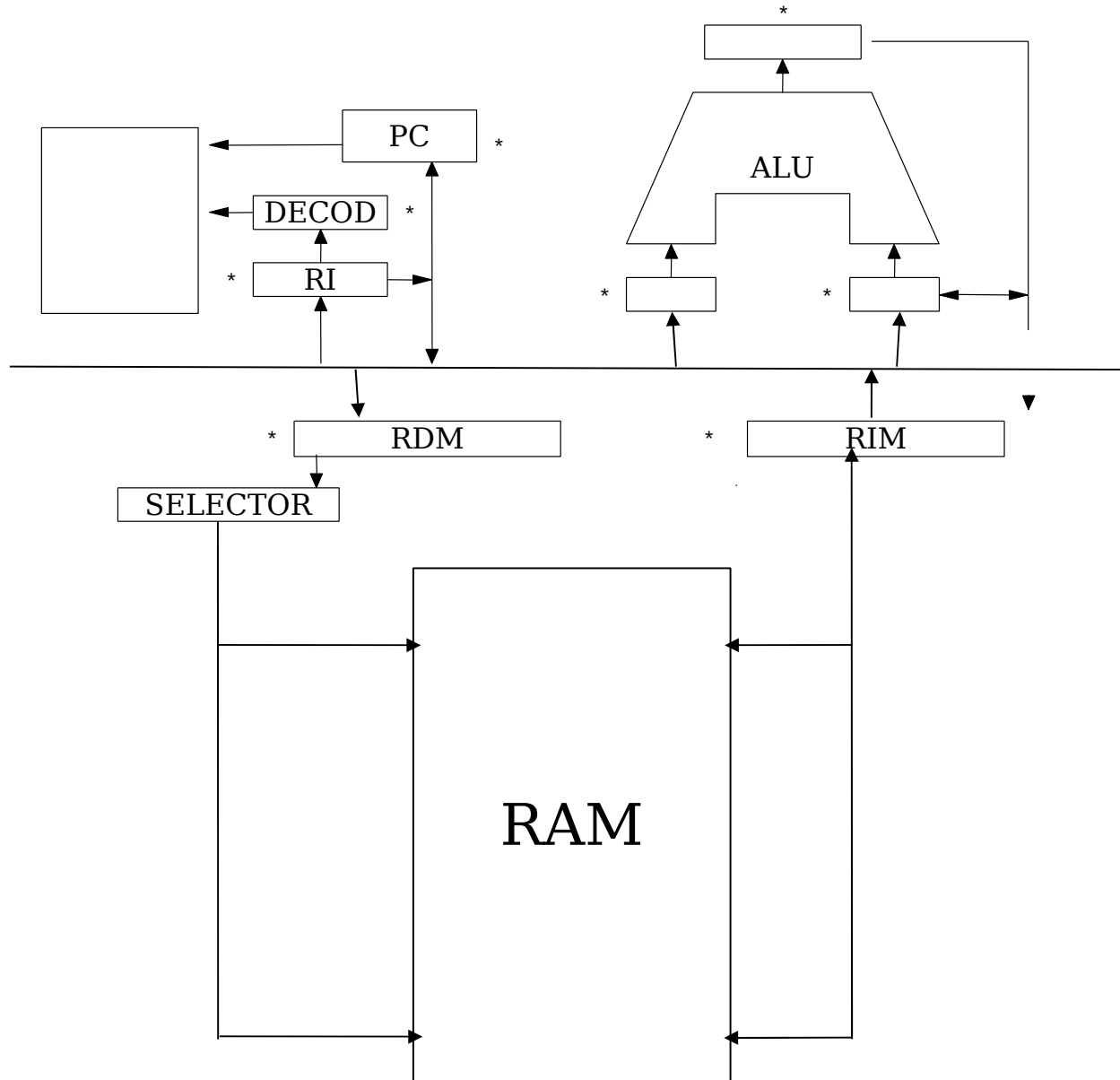
Wenceslao Palma M. <[wenceslao.palma@ucv.cl](mailto:wenceslao.palma@ucv.cl)>

[www.inf.ucv.cl/~wpalma/sc](http://www.inf.ucv.cl/~wpalma/sc)

# Introducción a los Sistemas Computacionales

Un vistazo de alto nivel caracteriza a un sistema computacional compuesto de procesador, memoria y componentes de e/s, los cuales se conectan de alguna forma para llevar a cabo la función principal de un computador, ejecutar programas.

# Introducción a los Sistemas Computacionales



# Introducción a los Sistemas Computacionales

## Ejecución de instrucciones. Funciones de E/S

Un módulo de E/S puede intercambiar datos directamente con el procesador.

El procesador puede leer/escribir datos desde/hacia un módulo de E/S.

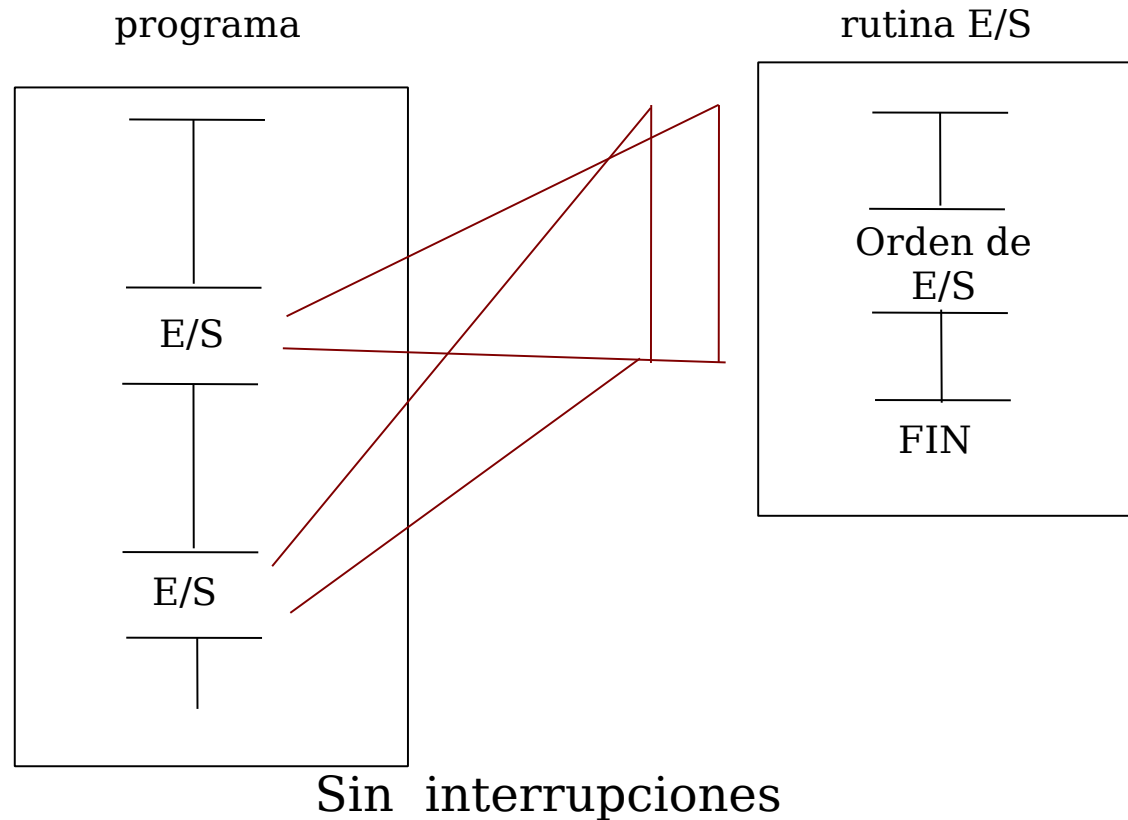
En algunos casos es conveniente que el intercambio de E/S se realice directamente con la memoria (DMA) liberando al procesador.

# Introducción a los Sistemas Computacionales

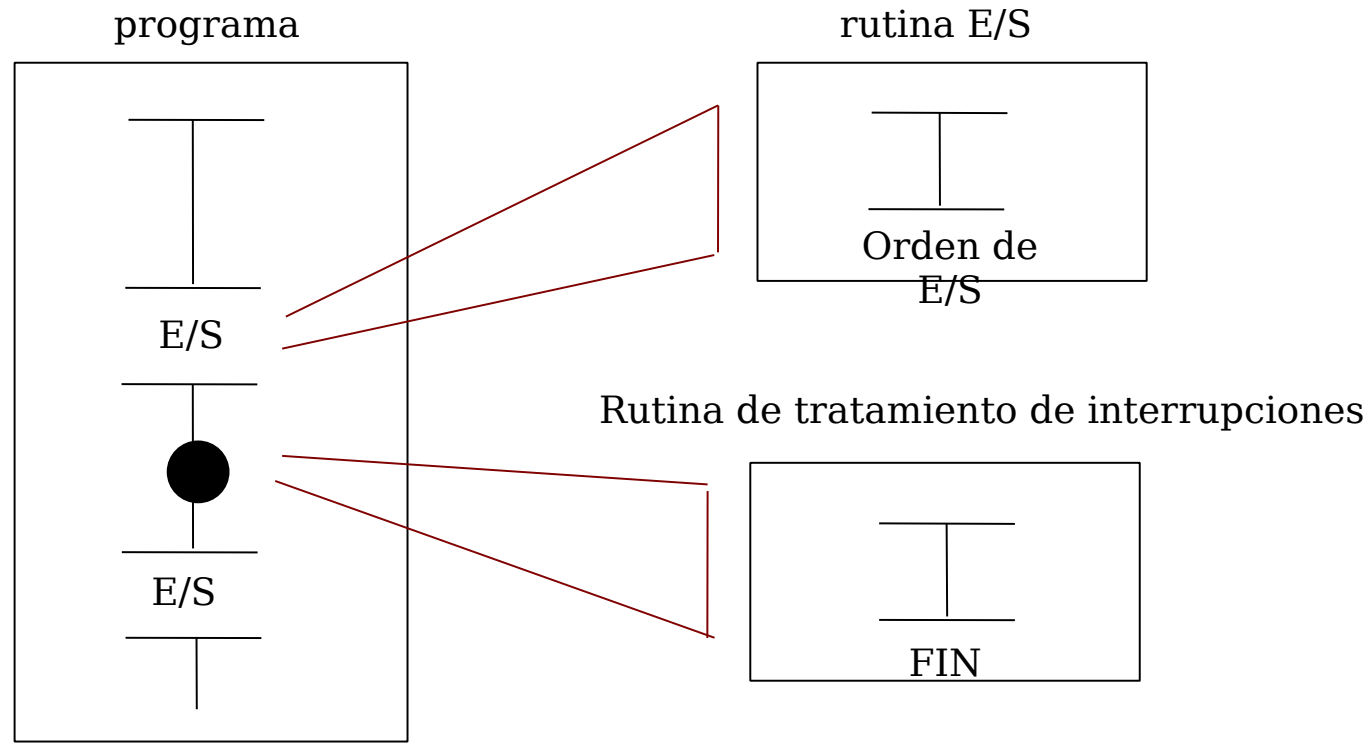
## Interrupciones

La mayoría de los dispositivos externos son mucho más lentos que el procesador.

Las interrupciones aparecen como una vía para mejorar la eficiencia del procesamiento.



# Introducción a los Sistemas Computacionales



## Con interrupciones

El programa percibe la interrupción como una interrupción de la secuencia normal de ejecución.

El procesador y el SO son los responsables de suspender el programa y reanudarlo en el mismo punto.

# Introducción a los Sistemas Computacionales

Generalmente la rutina de tratamiento de la interrupción es parte del SO. Determina el módulo de E/S que generó la interrupción y transfiere el control a dicha rutina.

Para poder procesar una interrupción pendiente se debe agregar un proceso de comprobación de interrupciones al ciclo de instrucción.

Si el tiempo necesario para completar una operación de E/S es más grande que lo que tarda el código del programa entre llamadas a E/S, el procesador tendrá un tiempo ocioso. Para aprovechar mejor el tiempo se utiliza la multiprogramación.

Existe la posibilidad de interrupciones múltiples, un programa puede estar recibiendo datos desde la tarjeta de red e imprimiendo resultados.

Para tratar interrupciones múltiples:  
    inhabilitar interrupciones  
    prioridades

# Introducción a los Sistemas Operativos

Un Sistema Operativo (SO) es un programa que controla la ejecución de las aplicaciones y actúa como interfaz entre el usuario/aplicación y el hardware del computador.

Es posible considerar que un SO tiene como objetivos la comodidad, eficiencia y capacidad de evolución.

El SO como interfaz ofrece los siguientes servicios:

- Creación de programas.

- Ejecución de programas.

- Acceso a dispositivos de E/S.

- Acceso controlado a archivos.

- Acceso al sistema.

- Detección y respuesta a errores.

- Contabilidad.

El SO como administrador de recursos decide cuanto tiempo de procesador se dedica a la ejecución de un programa, así como también determina cuando puede utilizarse un dispositivo de E/S.



# Introducción a los Sistemas Operativos

Un SO debe estar preparado para soportar actualizaciones y nuevos tipos de hardware, otorgar nuevo servicios y correcciones (parches).

Debido a lo anterior un SO requiere ser construido en forma modular, con interfaces bien definidas entre módulos y poseer una buena documentación.

# Introducción a los Sistemas Operativos

Resulta útil dar un vistazo a la evolución de los sistemas operativos y así visualizar el significado de las características actuales de los sistemas operativos contemporáneos.

## **Procesos en Serie**

No había sistema operativo.

El programador interactuaba directamente con el hardware del computador.

El programa debe incluir las rutinas para manejar cada dispositivo.

La depuración es visual, se observaban las luces de la consola.

Desventaja: mucho tiempo ocioso. Es necesario mejorar el rendimiento.

## Sistemas Batch (Procesamiento por Lotes)

Aparece el sistema operativo de procesamiento por lotes (batch).

La idea central es el uso de un elemento de software conocido como **monitor**

Los programadores entregan sus trabajos en cintas o tarjetas perforadas al operador del sistema.

El **monitor** se encarga de leer los trabajos y enviarlos a ejecución.

Cada programa se construía de tal modo que retorne al monitor al terminar su ejecución.

Cuando el programa presentaba algún error, el **monitor** imprime un volcado hexadecimal de la memoria, el cual es entregado al programador.

Desventaja: aún con la secuenciación automática la CPU todavía posee períodos de inactividad

# Introducción a los Sistemas Operativos

## **Operación fuera de línea (Off-line)**

Se utilizan procesadores satélite y cintas para acelerar el proceso de carga y salida.

De aquel modo se aumenta el porcentaje de utilización de CPU sin la necesidad de comprar una nueva CPU.

Desventaja: mientras se lee o escribe la CPU está en *busy waiting*, es decir el procesador aún tiene tiempos de ocio.

## **Buffers**

La idea es realizar en forma simultánea la E/S de un trabajo con su propio cómputo.

Cuando la CPU está lista para trabajar con datos ya leídos, se ordena al dispositivo de entrada que inicie de inmediato la siguiente lectura.

Desventaja: el rendimiento depende de la cuan intensivo en E/S sea el trabajo que se procesa.

## **Spooling (Simultaneous Peripheral Operation On-Line)**

Lectura e impresión ya no se realiza con procesadores satélite.

Lectoras e impresoras se conectan directamente al computador.

Básicamente el cambio sucedió con la aparición de discos de bajo costo.

El disco se utiliza como un buffer de gran tamaño para los datos de entrada y salida de un trabajo.

Es trabajo del monitor leer las tarjetas y dejarlas en disco.

# Introducción a los Sistemas Operativos

El grado de utilización de la CPU se mejora, cuando el monitor superpone las siguientes tareas:

- Procesamiento del trabajo que corresponde.

- Lectura de las tarjetas de uno o más trabajos que esperan ser procesados.

- Impresión de los resultados de trabajos ya procesados.

Desventaja: el disco aún sigue siendo más lento que la CPU.

## **Multiprogramación**

Se aprovechan los intervalos de espera por E/S para “avanzar” con otros trabajos.

El monitor decide que trabajos avanzan y cuales no.

Debido a que hay scheduling ya es posible hablar de un Sistema Operativo.

Desventaja: es necesario evitar que la ejecución concurrente genere “interferencias” entre los procesos.

Asociados a dicha desventaja aparecen:

- Espacio de direcciones protegido.

- Espacios de almacenamiento compartidos.

- Interrupciones.

- Scheduling de trabajos.

# Introducción a los Sistemas Operativos

## **Sistemas de Tiempo Compartido**

Extensión lógica de la multiprogramación.

Se recupera la capacidad de trabajo interactivo.

Múltiples usuarios acceden simultáneamente al sistema por medio de terminales.

El sistema operativo intercala la ejecución de cada programa en quantums de computación.

# Introducción a los Sistemas Operativos

## **Avances teóricos significativos en el desarrollo de los sistemas operativos**

Procesos: programa ejecutable+datos(variables, buffers, etc.)+contexto de ejecución

Administración de memoria: asignación y gestión automática, protección y control de acceso.

Seguridad y protección de los datos: acceso al sistema (autenticación), procesos sobre recursos, etc.

Planificación y gestión de recursos: equidad, sensibilidad diferencial, eficiencia.

Estructura del sistema: shell, procesos de usuario, directorios, dispositivos, sistemas de archivos, comunicaciones, memoria virtual, almacenamiento secundario, procesos primitivos, interrupciones, procedimientos, conjunto de instrucciones.



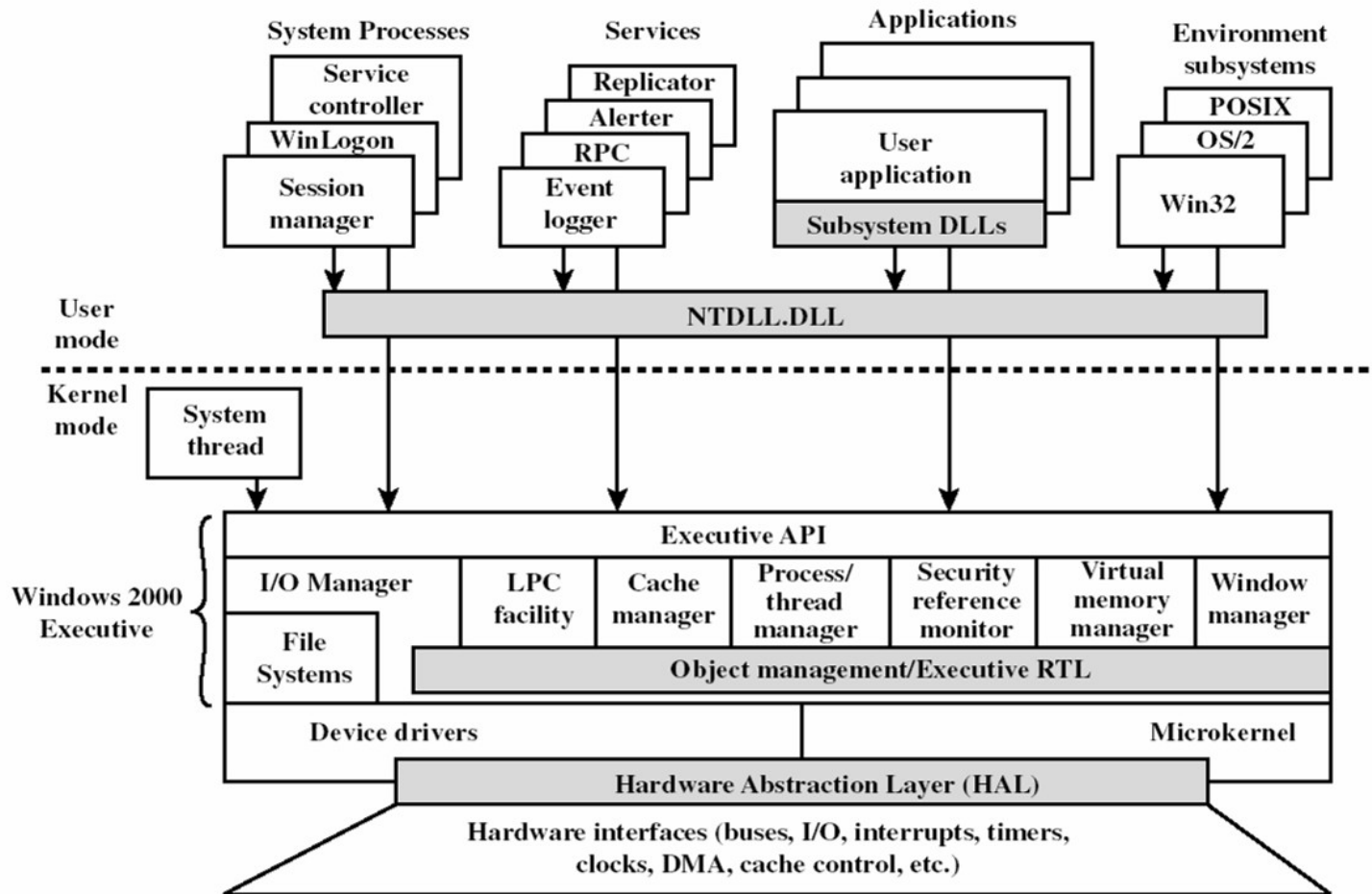


Figure 2.13 Windows 2000 Architecture

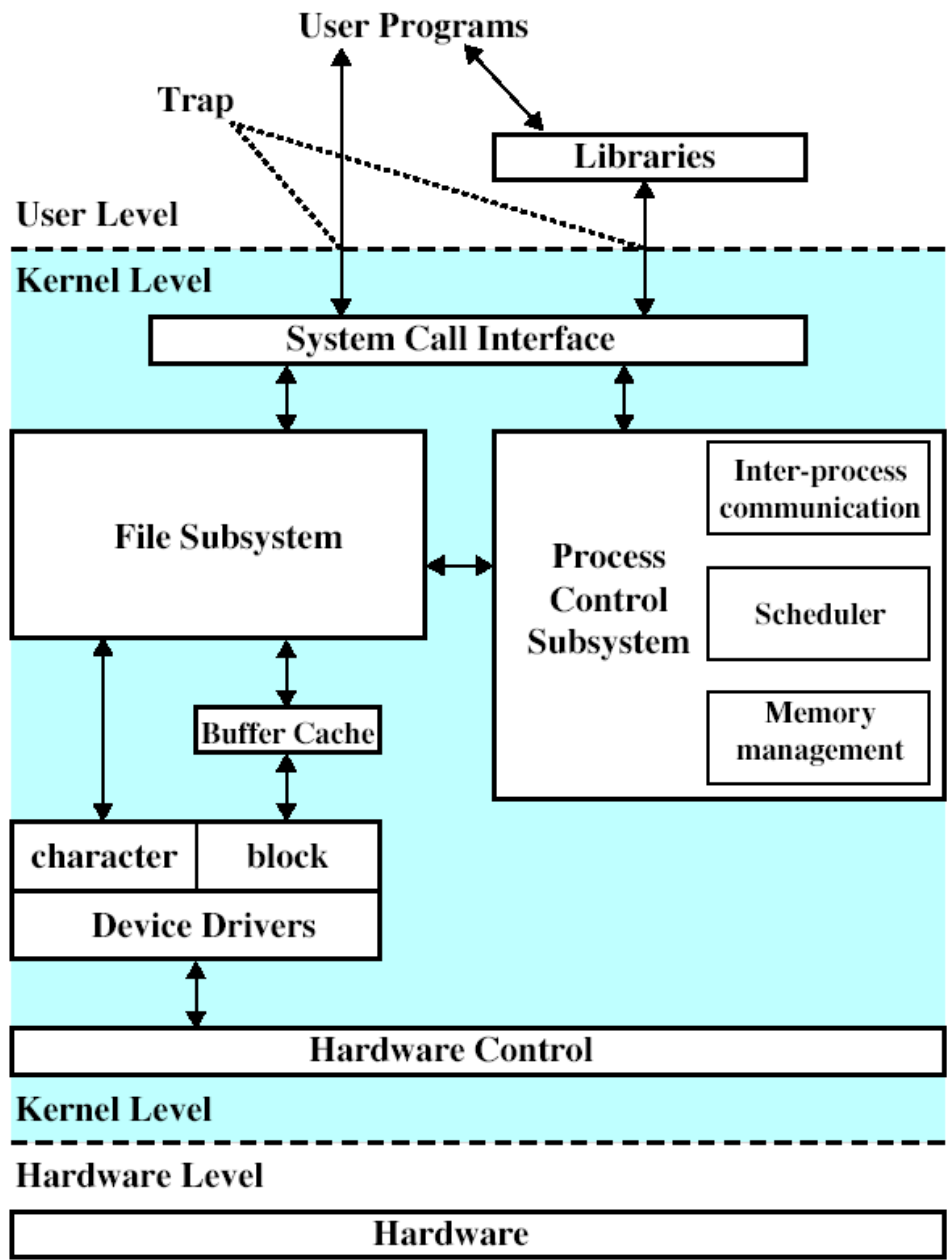


Figure 2.15 Traditional UNIX Kernel [BACH86]