Sistemas Operativos

Estructura y servicios de un SO

Departamento de Ingeniería en Sistemas y Computación
Universidad Católica del Norte, Antofagasta.

Arquitectura de sistemas computacionales

- Procesador simple de propósito general
- Multiprocesadores
  - Aumento de productividad
  - Confiabilidad, tolerancia a fallas

Arquitectura de sistemas computacionales

- Procesador simple de propósito general
- Multiprocesadores
  - Aumento de productividad
  - Confiabilidad, tolerancia a fallas

Arquitectura de sistemas computacionales

- Procesador simple de propósito general
- Multiprocesadores
  - Aumento de productividad
  - Confiabilidad, tolerancia a fallas

Arquitectura de sistemas computacionales

- Procesador simple de propósito general
- Multiprocesadores
  - Aumento de productividad
  - Confiabilidad, tolerancia a fallas

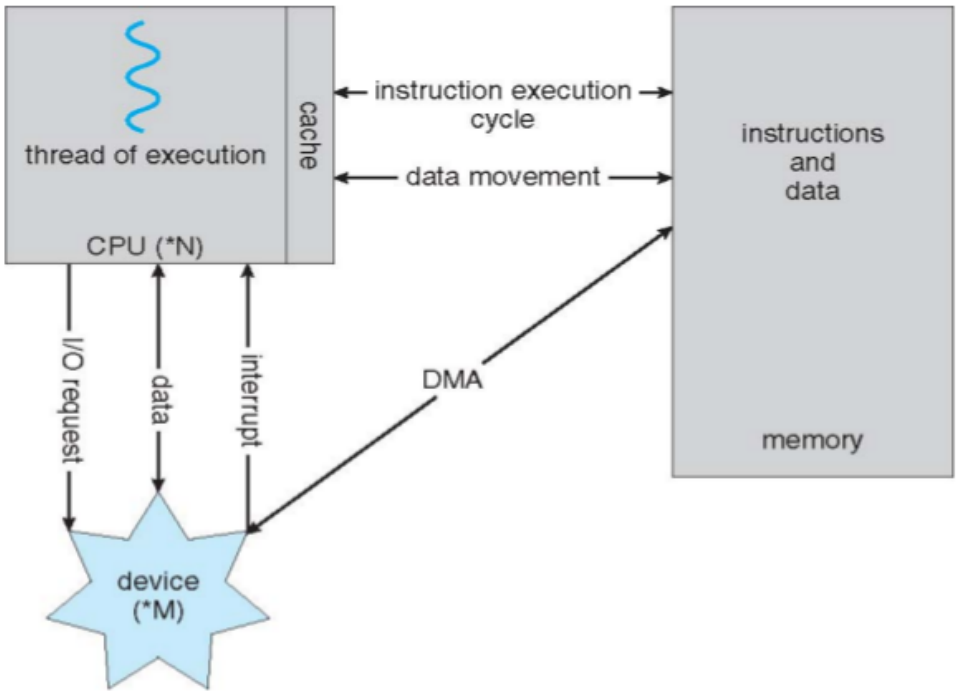
Multiprocesadores

- Multiprocesamiento asimétrico: maestro-esclavo
- Multiprocesamiento simétrico: todos los procesadores son pares

Multiprocesadores

- Multiprocesamiento asimétrico: maestro-esclavo
- Multiprocesamiento simétrico: todos los procesadores son pares

Interacciones en un computador



Sistemas en cluster

- Equivalente a un multiprocesador (no es lo mismo)
- Compuesto por múltiples sistemas trabajando de forma coordinada
- Usualmente comparten almacenamiento a través de un storage-area network

Sistemas en cluster

- Equivalente a un multiprocesador (no es lo mismo)
- Compuesto por múltiples sistemas trabajando de forma coordinada
- Usualmente comparten almacenamiento a través de un storage-area network

Sistemas en cluster

- Equivalente a un multiprocesador (no es lo mismo)
- Compuesto por múltiples sistemas trabajando de forma coordinada
- Usualmente comparten almacenamiento a través de un storage-area network

Estructura de un SO

Dos conceptos relevantes:

- Multiprogramación: para lograr eficiencia

- Tiempo compartido (multitarea):
  - Extensión lógica de multiprogramación
  - CPU conmuta entre distintas tareas

Estructura de un SO

Dos conceptos relevantes:

- Multiprogramación: para lograr eficiencia
- Tiempo compartido (multitarea):
  - Extensión lógica de multiprogramación
  - CPU conmuta entre distintas tareas

Estructura de un SO

Dos conceptos relevantes:

- Multiprogramación: para lograr eficiencia
- Tiempo compartido (multitarea):
  - Extensión lógica de multiprogramación
  - CPU conmuta entre distintas tareas

Multiprogramación

- Se basa en que un usuario no puede ocupar simultáneamente la CPU y dispositivos E/S
- Se organizan los trabajos para que la CPU siempre tenga uno que ejecutar
- Cuando un trabajo espera por E/S, el SO conmuta a otro

Multiprogramación

- Se basa en que un usuario no puede ocupar simultáneamente la CPU y dispositivos E/S
- Se organizan los trabajos para que la CPU siempre tenga uno que ejecutar
- Cuando un trabajo espera por E/S, el SO conmuta a otro

Multiprogramación

- Se basa en que un usuario no puede ocupar simultáneamente la CPU y dispositivos E/S
- Se organizan los trabajos para que la CPU siempre tenga uno que ejecutar
- Cuando un trabajo espera por E/S, el SO conmuta a otro

Multitarea

- Cada usuario tiene al menos un programa ejecutándose en memoria
- Si un proceso no cabe en la memoria, se usa swapping para mover trabajos hacia y desde la memoria
- La memoria virtual permite la ejecución de procesos que no tienen suficiente memoria

Multitarea

- Cada usuario tiene al menos un programa ejecutándose en memoria
- Si un proceso no cabe en la memoria, se usa swapping para mover trabajos hacia y desde la memoria
- La memoria virtual permite la ejecución de procesos que no tienen suficiente memoria

Multitarea

- Cada usuario tiene al menos un programa ejecutándose en memoria
- Si un proceso no cabe en la memoria, se usa swapping para mover trabajos hacia y desde la memoria
- La memoria virtual permite la ejecución de procesos que no tienen suficiente memoria

Operaciones de un SO

- Las interrupciones son activadas por hardware
- Las excepciones (traps) se activan por requerimientos o errores en el software(división por cero, loops infinitos, violaciones de memoria)

Operaciones de un SO

- Las interrupciones son activadas por hardware
- Las excepciones (traps) se activan por requerimientos o errores en el software(división por cero, loops infinitos, violaciones de memoria)

Operaciones de un SO

- Las interrupciones son activadas por hardware
- Las excepciones (traps) se activan por requerimientos o errores en el software(división por cero, loops infinitos, violaciones de memoria)

Modo dual de operación

- El bit de modo protege al SO y sus componentes
- Instrucciones privilegiadas sólo en modo kernel
- Una llamada al sistema permite pasar a modo kernel. Al retorno se vuelve a modo usuario.

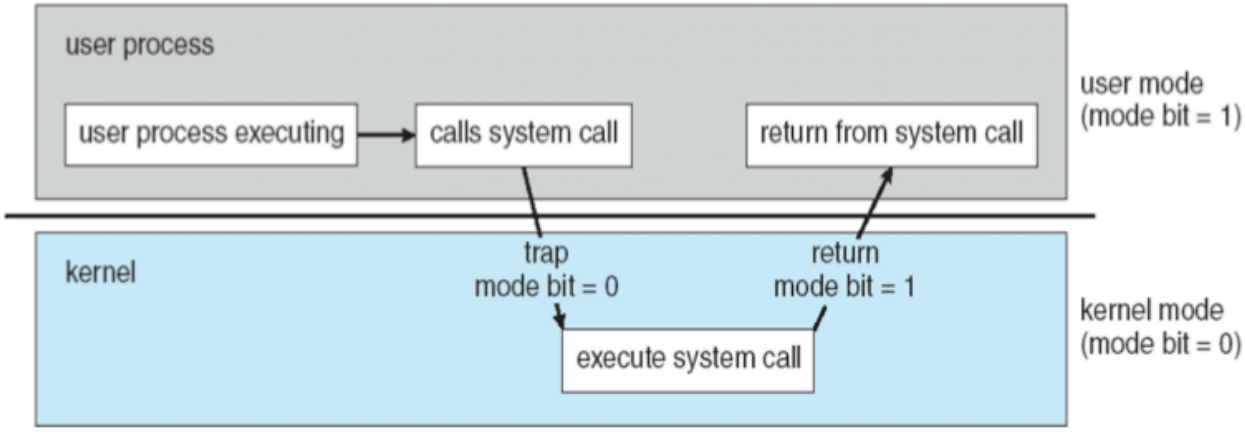
Modo dual de operación

- El bit de modo protege al SO y sus componentes
- Instrucciones privilegiadas sólo en modo kernel
- Una llamada al sistema permite pasar a modo kernel. Al retorno se vuelve a modo usuario.

Modo dual de operación

- El bit de modo protege al SO y sus componentes
- Instrucciones privilegiadas sólo en modo kernel
- Una llamada al sistema permite pasar a modo kernel. Al retorno se vuelve a modo usuario.

Modo dual de operación



Gestión de procesos

- Un proceso (entidad activa) es un programa (entidad pasiva) en ejecución.
- Un proceso necesita recursos para completar su tarea, que son devueltos a su término.
- El registro PC (Program Counter) determina la siguiente instrucción a ejecutar, de forma secuencial hasta el término.

Gestión de procesos

- Un proceso (entidad activa) es un programa (entidad pasiva) en ejecución.
- Un proceso necesita recursos para completar su tarea, que son devueltos a su término.
- El registro PC (Program Counter) determina la siguiente instrucción a ejecutar, de forma secuencial hasta el término.

Gestión de procesos

- Un proceso (entidad activa) es un programa (entidad pasiva) en ejecución.
- Un proceso necesita recursos para completar su tarea, que son devueltos a su término.
- El registro PC (Program Counter) determina la siguiente instrucción a ejecutar, de forma secuencial hasta el término.

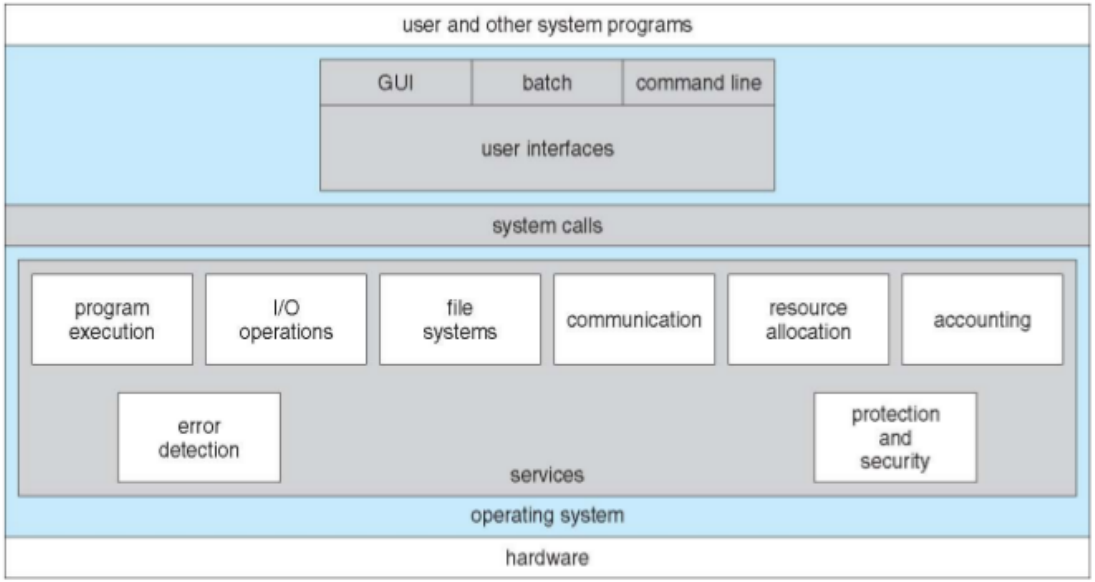
- En procesos multi-hebras, cada hebra tiene su propio PC
- Tipicamente, en un sistema hay múltiples procesos corriendo en una o varias CPU (procesos de usuarios y del sistema operativo)

Gestión de procesos

- En procesos multi-hebras, cada hebra tiene su propio PC
- Tipicamente, en un sistema hay múltiples procesos corriendo en una o varias CPU (procesos de usuarios y del sistema operativo)

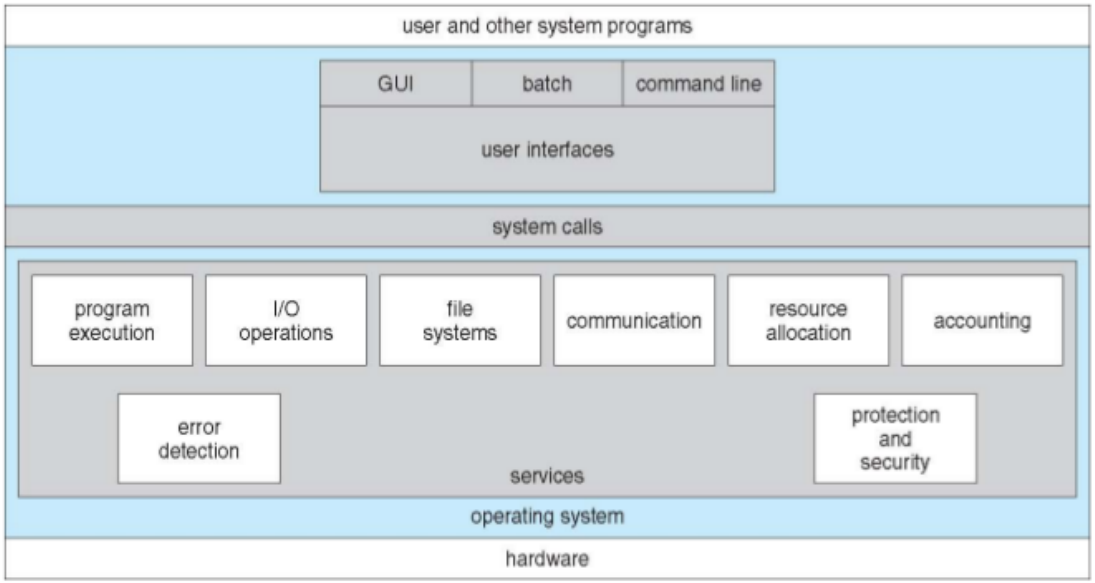
- En procesos multi-hebras, cada hebra tiene su propio PC
- Tipicamente, en un sistema hay múltiples procesos corriendo en una o varias CPU (procesos de usuarios y del sistema operativo)

Servicios



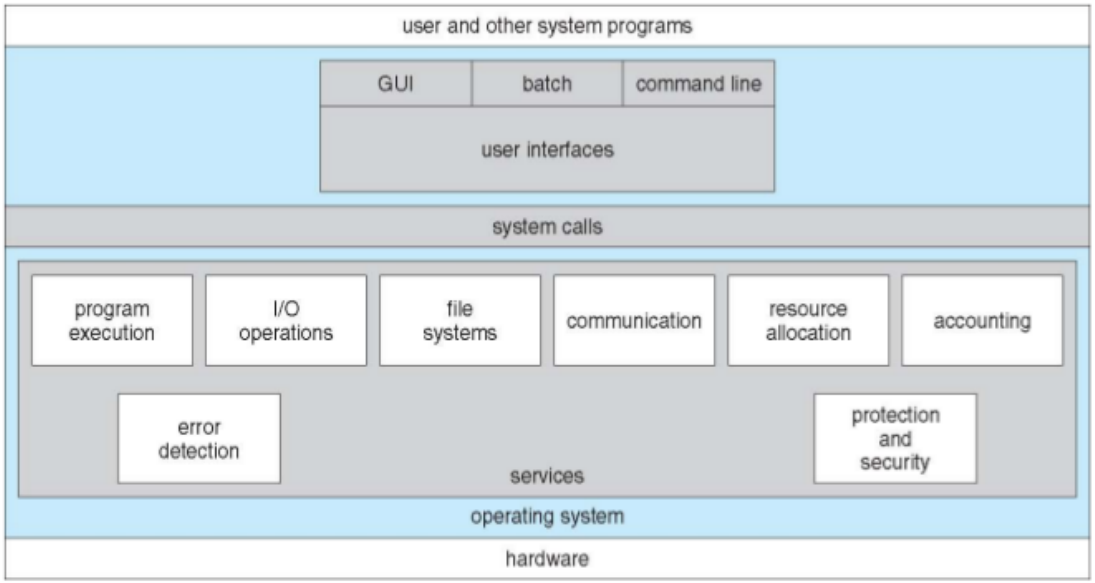
- Un conjunto de servicios corresponden a funciones útiles a los usuarios

Servicios



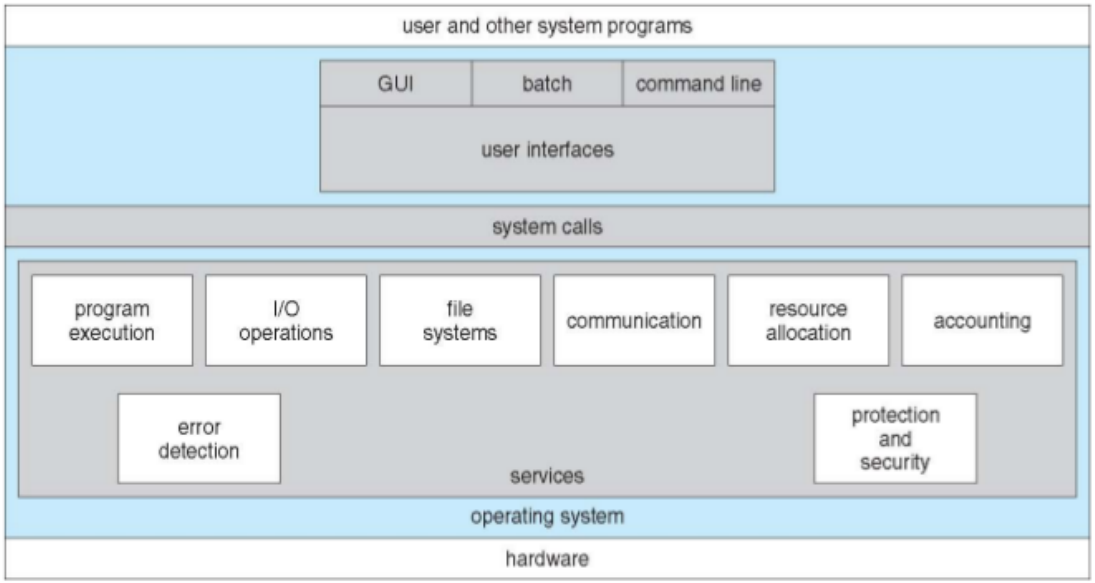
- **Interfaz**: La interfaz con el usuario tipicamente varía entre línea de comandos (CLI) a interfaz gráfica (GUI)

Servicios



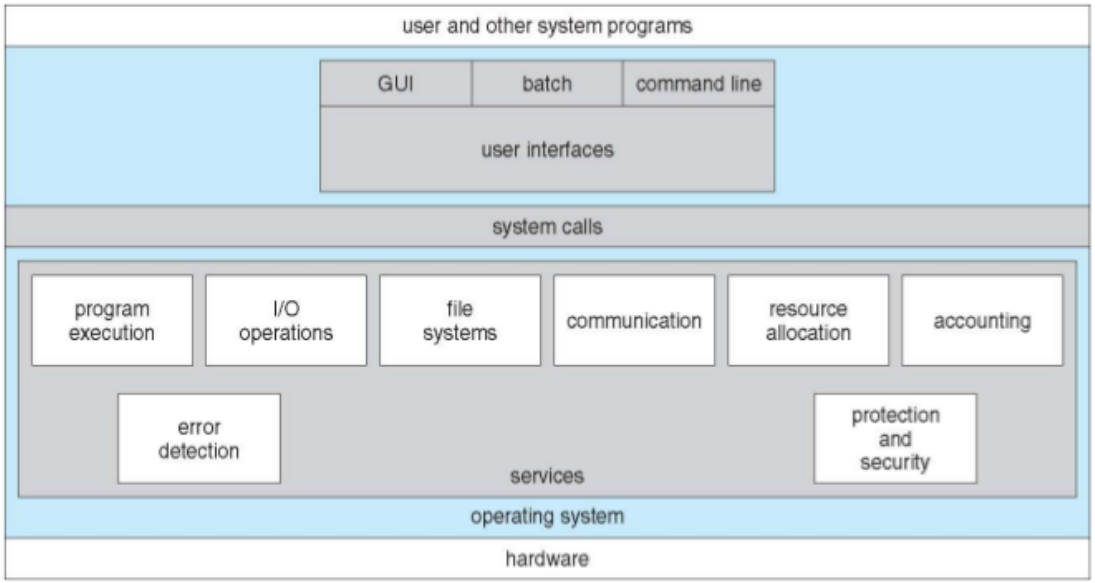
- **Ejecución de programas**: El SO debe poder cargar el programa en memoria y ejecutar.

Servicios



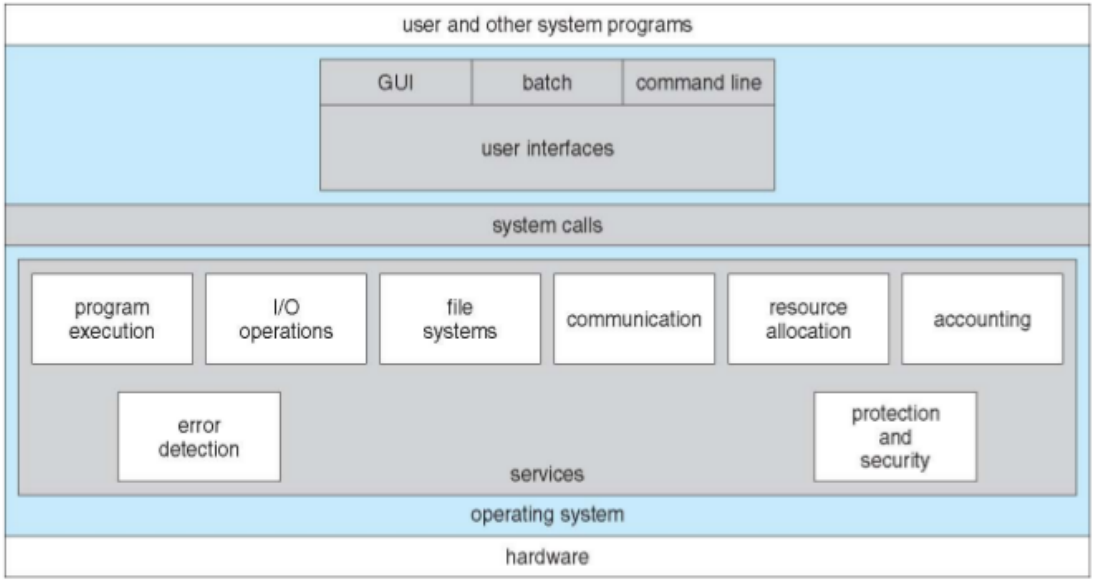
- **Operaciones E/S**: Un programa requiere de E/S, ya sea archivos o algún dispositivo.

Servicios



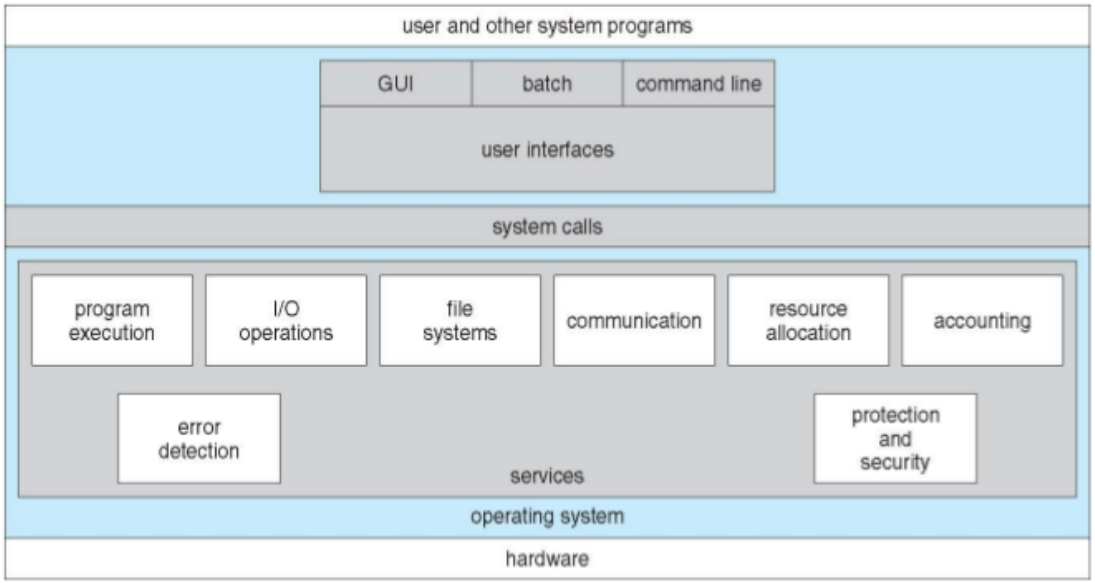
- **Manejo de archivos**: Los programas necesitan leer y escribir archivos, crear, borrar, buscar en ellos, mostrar información y gestionar permisos.

Servicios



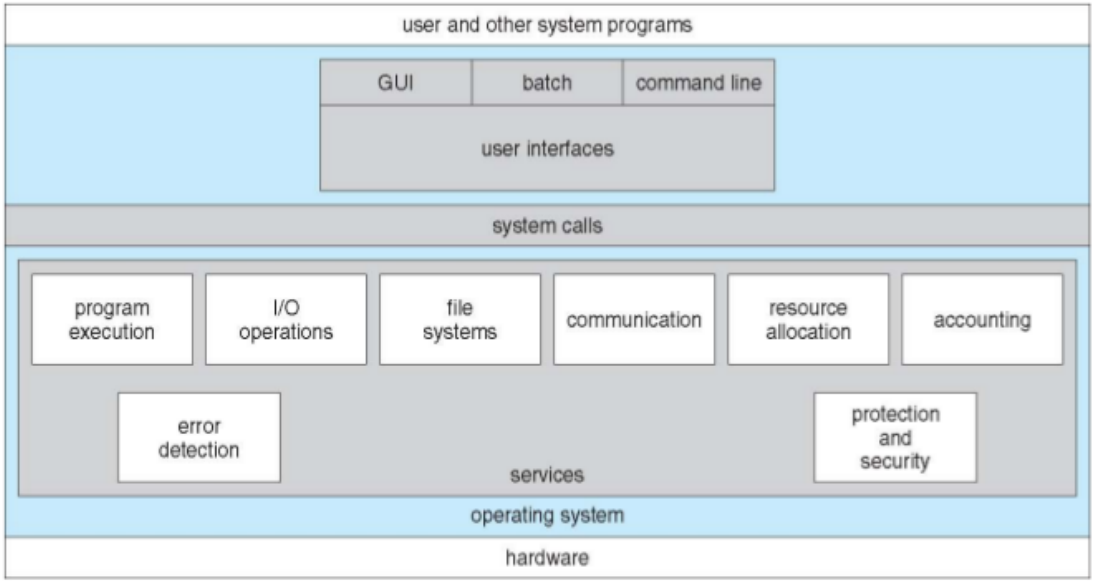
- **Comunicación**: Los procesos pueden intercambiar información, dentro del mismo computador o entre computadores sobre una red.

Servicios



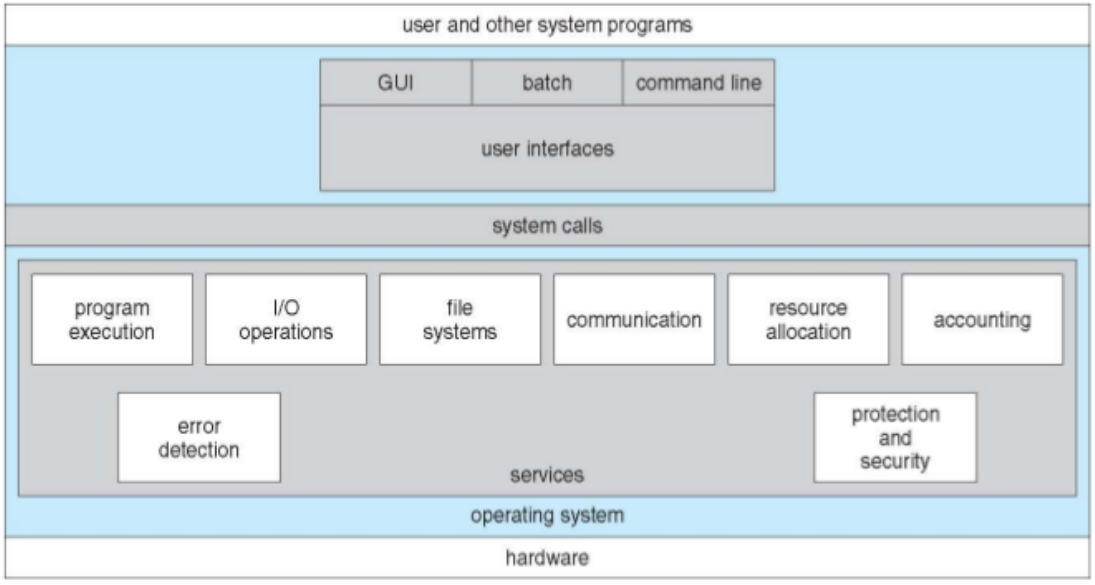
- **Comunicación**: Internamente, la comunicación puede ser vía memoria compartida o a través de paso de mensajes.

Servicios



- **Detección de errores**: Las facilidades de debugging pueden ayudar a la productividad de programadores y a utilizar eficientemente el sistema.

Servicios



- **Asignación de recursos**: Necesario cuando existen múltiples usuarios o múltiples tareas concurrentes.

Interfaz Usuario

- El intérprete de comandos (CLI) permite interactuar directamente con el SO
- CLI puede estar en el kernel o a través de programas de sistema (shells)
- Otra opción es la interfaz gráfica (GUI), que utiliza mouse, teclado y monitor

Interfaz Usuario

- El intérprete de comandos (CLI) permite interactuar directamente con el SO
- CLI puede estar en el kernel o a través de programas de sistema (shells)
- Otra opción es la interfaz gráfica (GUI), que utiliza mouse, teclado y monitor

Interfaz Usuario

- El intérprete de comandos (CLI) permite interactuar directamente con el SO
- CLI puede estar en el kernel o a través de programas de sistema (shells)
- Otra opción es la interfaz gráfica (GUI), que utiliza mouse, teclado y monitor

Estructura de un SO

- Las formas de estructurar un SO son:
  - Estructuras simples
  - Estructuras de capas
  - Micro kernels
  - Módulos

Estructura de un SO

- Las formas de estructurar un SO son:
  - Estructuras simples
  - Estructuras de capas
  - Micro kernels
  - Módulos

Estructura de un SO

- Las formas de estructurar un SO son:
  - Estructuras simples
  - Estructuras de capas
  - Micro kernels
  - Módulos

Estructura de un SO

- Las formas de estructurar un SO son:
  - Estructuras simples
  - Estructuras de capas
  - Micro kernels
  - Módulos

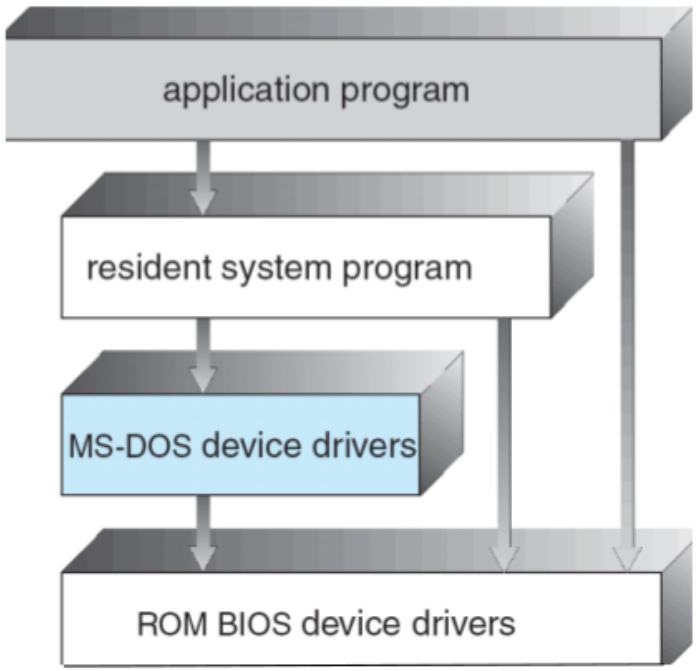
Estructura simple

- Unix y MS-DOS son ejemplos de estructuras desarrolladas inicialmente en hardware con funcionalidades limitadas
- MS-DOS fue escrito para proporcionar la mayor funcionalidad en el mínimo espacio, no está dividido en módulos.

Estructura simple

- Unix y MS-DOS son ejemplos de estructuras desarrolladas inicialmente en hardware con funcionalidades limitadas
- MS-DOS fue escrito para proporcionar la mayor funcionalidad en el mínimo espacio, no está dividido en módulos.

Estructura de niveles en MS-DOS



UNIX

- Al igual que MS-DOS fue desarrollado inicialmente en hardware con funcionalidades limitadas
- Se divide en programas de sistemas y kernel, que a su vez es separado en un conjunto de subsistemas
- Kernel: todo lo que está entre las llamadas al sistema y hardware. Provee entre otras cosas el sistema de archivos, itineración de CPU y gestión de memoria.

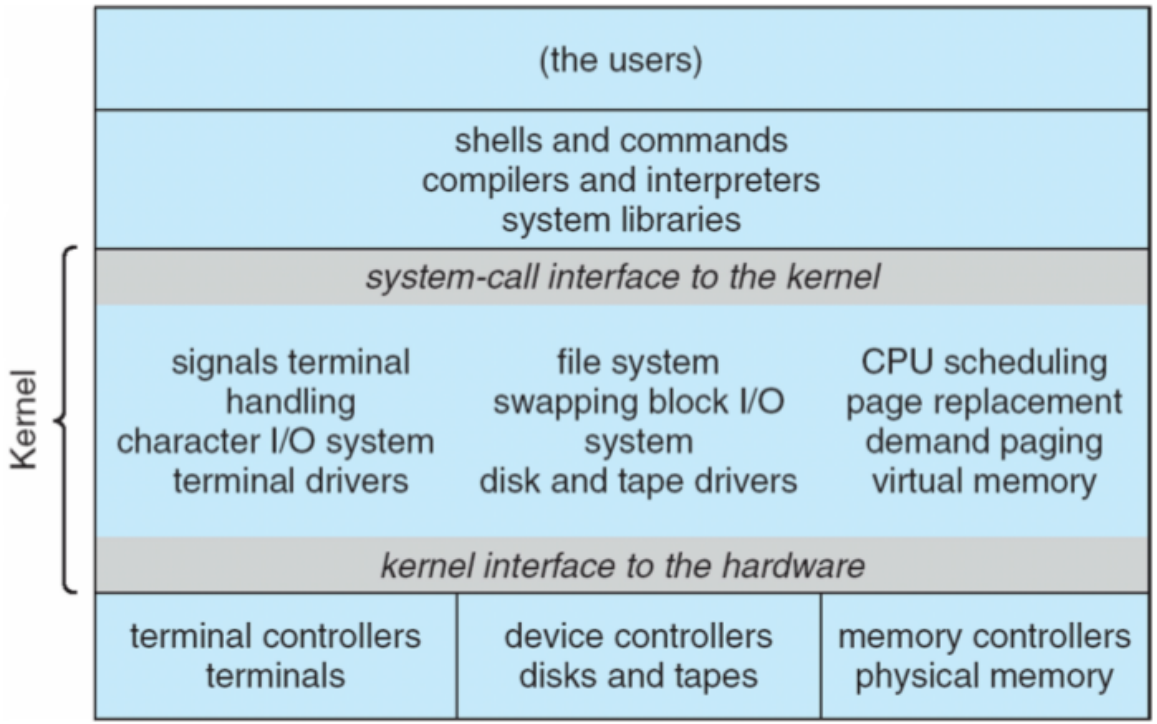
UNIX

- Al igual que MS-DOS fue desarrollado inicialmente en hardware con funcionalidades limitadas
- Se divide en programas de sistemas y kernel, que a su vez es separado en un conjunto de subsistemas
- Kernel: todo lo que está entre las llamadas al sistema y hardware. Provee entre otras cosas el sistema de archivos, itineración de CPU y gestión de memoria.

UNIX

- Al igual que MS-DOS fue desarrollado inicialmente en hardware con funcionalidades limitadas
- Se divide en programas de sistemas y kernel, que a su vez es separado en un conjunto de subsistemas
- Kernel: todo lo que está entre las llamadas al sistema y hardware. Provee entre otras cosas el sistema de archivos, itineración de CPU y gestión de memoria.

Estructura de un sistema UNIX tradicional



Estructuración por capas

- El SO se divide en un número de capas o niveles
- La capa 0 (inferior) corresponde al hardware.
- La capa N (más alta) es la interfaz usuario
- Con modularidad, las capas se seleccionan de forma tal que cada una utiliza funciones y servicios **sólo** de las capas inferiores

Estructuración por capas

- El SO se divide en un número de capas o niveles
- La capa 0 (inferior) corresponde al hardware.
- La capa N (más alta) es la interfaz usuario
- Con modularidad, las capas se seleccionan de forma tal que cada una utiliza funciones y servicios **sólo** de las capas inferiores

Estructuración por capas

- El SO se divide en un número de capas o niveles
- La capa 0 (inferior) corresponde al hardware.
- La capa N (más alta) es la interfaz usuario
- Con modularidad, las capas se seleccionan de forma tal que cada una utiliza funciones y servicios **sólo** de las capas inferiores

Estructuración por capas

- El SO se divide en un número de capas o niveles
- La capa 0 (inferior) corresponde al hardware.
- La capa N (más alta) es la interfaz usuario
- Con modularidad, las capas se seleccionan de forma tal que cada una utiliza funciones y servicios **sólo** de las capas inferiores

Estructura microkernel

- La idea es mover todo lo posible desde el kernel al espacio de usuario
- La comunicación se desarrolla entre los módulos usuarios por paso de mensajes

Estructura microkernel

- La idea es mover todo lo posible desde el kernel al espacio de usuario
- La comunicación se desarrolla entre los módulos usuarios por paso de mensajes

Estructuración por módulos

- La mayoría de los SO modernos implementan el kernel en base a módulos
- Cada módulo se carga en la medida que se necesita (módulos de carga dinámica)

Estructuración por módulos

- La mayoría de los SO modernos implementan el kernel en base a módulos
- Cada módulo se carga en la medida que se necesita (módulos de carga dinámica)

Estructuración por módulos