Sistemas Operativos

Diego Enrique Fontán, CosasDePuma

2017-18

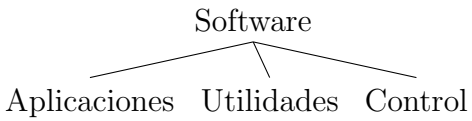
Índice

1. Niveles de software **1**
1.1. Concepto de Sistema Operativo 1
1.1.1. Características del Sistema Operativo 1
1.1.2. Funciones principales del Sistema Operativo 2

2. Tipos de Sistemas **3**
2.1. Sistemas monolíticos 3
2.2. Sistemas en estratos 3
2.3. Máquinas virtuales 4
2.4. Modelo Cliente-Servidor 4
2.5. Estructura orientada al objeto 5
2.6. Sistemas híbridos 5

1. Niveles de software

Se denomina **soporte lógico** o **software** a todo aquel conjunto de programas asociados a un ordenador.



Niveles de software

1.1. Concepto de Sistema Operativo

Un **Sistema Operativo** es un programa (o conjunto de subprogramas o módulos) de control que tiene como finalidad facilitar el uso del ordenador y conseguir que se utilice eficientemente.

1.1.1. Características del Sistema Operativo

- Actúa como interfaz entre el usuario y la máquina física.

- Controla la ejecución de otros programas.

- Gestiona y mantiene ficheros.

- Contabiliza la utilización de los recursos.

- Protege los datos y los programas.

- Gestiona y asigna directamente los recursos hardware.

⋮

1.1.2. Funciones principales del Sistema Operativo

El Sistema Operativo debe inicializar la máquina y preparar el ordenador para su funcionamiento mediante una **inicialización total** (Initial Program Loading, Bootstrapping) o mediante una **inicialización parcial**.

También servirá de **máquina extendida o virtual** con el fin de ocultar los detalles del hardware al usuario y proporcionar un entorno más cómodo. De esta forma se logran varios objetivos:

- Evitar que la ejecución de los programas se interfieran unos entre otros, proporcionando así **seguridad**.

- Construir recursos virtuales de alto nivel a partir de recursos físicos de de más bajo nivel.

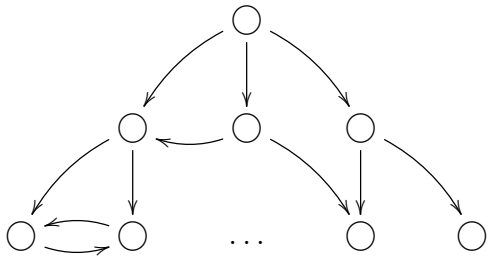
Otra de las funciones principales de un Sistema Operativo es la de **administrar los recursos para su funcionamiento**, ya sea *asignandole* todos los que requiera un programa para su ejecución o *controlando* el uso corrector de estos.

Además, es importante que un Sistema Operativo sea **determinista**, para que un mismo programa ejecutado con los mismo datos de los mismos resultados en cualquier momento y en cualquier ejecución, y que sea **indeterminista**, para que pueda responder a circunstancias que ocurren de manera impredecible.

2. Tipos de Sistemas

2.1. Sistemas monolíticos

No tienen una estructrura definida. Se componen de un conjunto de procedimientos donde cada uno de ellos puede llamar a todos los demás.



2.2. Sistemas en estratos

Se organiza en una jerarquía de estratos, estando construído cada uno de ellos sobre el otro que tiene menor jerarquía que él.

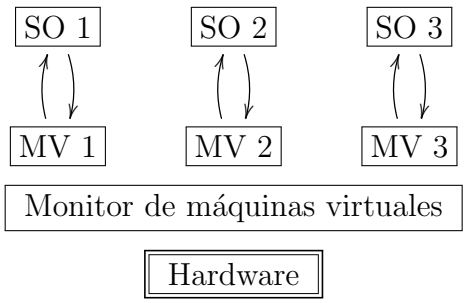
Operador del SO
Programas de usuario
Administración de E/S
Comunicación operador-procesos
Administración de memoria y tambor
Distribución del procesador y multiprogramación

Ejemplo de sistema THE

2.3. Máquinas virtuales

Máquinas similares a la máquina real pero de carácter virtual.

El programa de control se ejecuta sobre el propio hardware y ofreec al nivel superior varias máquinas virtuales.



2.4. Modelo Cliente-Servidor

Su objetivo es minimizar el kernel desplazando el código de todos sus servicios a estratos lo más superiores posibles.

Para ello, la mayoría de sus funciones se implementan como procesos del servidor, denominados **procesos servidores**, de forma que cuando un proceso de usuario, llamado **proceso cliente**, necesita un servicio del SO, lo que hace es enviar un mensaje al servidor correspondiente (el cual realiza el trabajo y devuelve la respuesta).

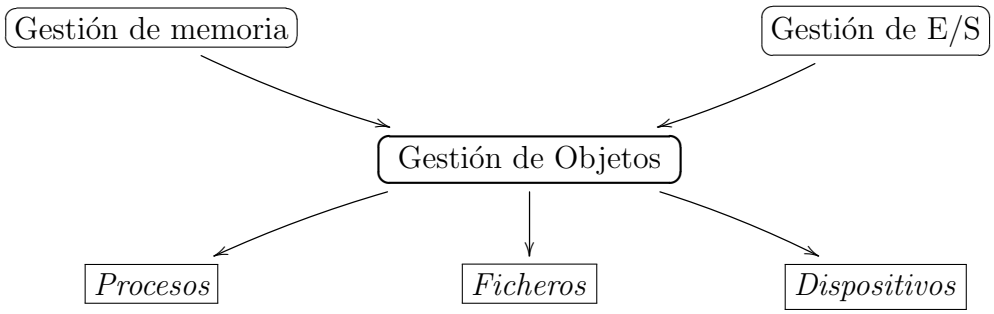
El kernel. lo único que hace es implementar la comunicación entre cliente-servidor y entre servidor-hardware.

2.5. Estructura orientada al objeto

Se basan en una colección de objetos, donde las funciones del sistema son ficheros, dispositivos, etc.

La interacción entre dichos objetos viene determinada por las capacidades que cada uno tenga para actuar con el otro.

El kerner es el responsable del mantenimiento de las definiciones de los tipos de objetos soportados y del control de los privilegios de acceso de los mismos.



2.6. Sistemas híbridos

Son similares a los sistemas cliente-servidor, aunque añaden ciertas funcionalidades al kernel para que se ejecute más rápido que si permanecen en el espacio de usuario.

Se les llama híbridos porque usan mecanismos o conceptos de arquitectura de los sistemas monolíticos y de los sistemas cliente-servidor.