

## Tema 1. ESTRUCTURAS DE SISTEMAS OPERATIVOS

### 1. INTRODUCCION

### 2. SISTEMAS OPERATIVOS DE PROPOSITO ESPECÍFICO

### 3. ESTRUCTURA DE UN SISTEMA OPERATIVO

#### Bibliografía

Silberschatz, A.; *Fundamentos de Sistemas Operativos (7/e)*, McGraw-Hill, 2006.

Stallings, W.; *Sistemas Operativos. Aspectos Internos y Principios de Diseño (5/e)*, Prentice Hall.

Tanenbaum, A.S.; *“Sistemas Operativos Modernos” (3/e)*, Pearson Prentice Hall, 2

1

## 1. INTRODUCCION

*Stallings apartados 2.1 a 2.4: Descripción de conceptos como objetivos y funciones de un SO, interfaz del SO,... Sobre lo que habla sobre la historia: solo definición de conceptos como multiprogramación, tiempo compartido,...*

#### \* Multiprogramación

Capacidad de un Sistema Operativo (SO) de ejecutar simultáneamente varios programas que residen simultáneamente en Memoria Principal (MP).

Necesita interrupciones para las operaciones de Entrada/Salida (E/S)

El SO deberá...

- proporcionar rutinas de E/S para proteger los dispositivos. El software que accede a los dispositivos forma parte del SO
- gestionar la MP, puesto que los programas han de residir en MP
- planificar la CPU

#### \* Tiempo compartido

Capacidad de un SO de que varios procesos **interactivos** se ejecuten simultáneamente  
Presupone multiprogramación

2

### \* Características de un SO

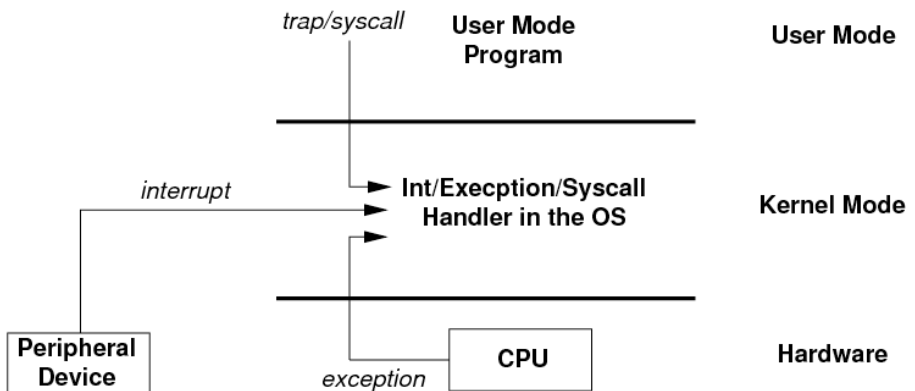
- corrección
- eficiencia
- fiabilidad
  - debe responder de forma predecible a condiciones de error, incluidos fallos hardware
  - debe proporcionar protección, tanto a programas de usuario como al propio software del SO, ante software erróneo o con mala intención.
- extensibilidad (añadir, eliminar o modificar una parte supone un coste mínimo)
- tolerancia a fallos (el fallo de una parte no inhabilita totalmente al sistema a seguir funcionando)
- reentrancia (<http://www.ganssle.com/articles/areentra.htm>)

Conseguir propiedades anteriores es particularmente difícil en un SO debido a su carácter de sistema concurrente.

Eventos que provocan concurrencia en el SO:

- llamadas al sistema
- interrupciones
- excepciones
- partes del SO que se ejecutan como programas concurrentes

3



4

## 2. SISTEMAS OPERATIVOS DE PROPÓSITO ESPECÍFICO:

### **Sistemas operativos de tiempo real**

(Silberschatz pág. 633) Un SO de tiempo real requiere no solo que los resultados sean correctos desde un punto de vista lógico, sino que se produzcan dentro de un tiempo especificado

“SO de tiempo real estricto” (Hard Real Time): es absolutamente imperativo que las respuestas se produzcan en el tiempo especificado

“SO de tiempo real no estricto” (Soft Real Time): los tiempos de respuesta son importantes pero el sistema seguirá funcionando correctamente aunque los tiempos límite no se cumplan ocasionalmente

### **Sistema operativo en red – sistema operativo distribuido**

Sistema informático distribuido:

conjunto de ordenadores sin memoria ni reloj común interconectados a través de un sistema de comunicación que podría permitir que estén geográficamente dispersos.

Este sistema de comunicación constituye la única posibilidad de comunicar información.

La comunicación se realiza mediante operaciones de enviar/recibir mensajes, siendo la duración del envío de un mensaje grande respecto al tiempo de ejecución de una instrucción máquina.

Un sistema distribuido tiene como objetivo compartir recursos o permitir la dispersión geográfica.

5

(Stallings 13.0; Tanenbaum pág 18)

#### **SO en red:**

Cada computador tiene su propio sistema operativo;  
el SO en red es un añadido al sistema operativo local añadiendo posibilidades de comunicación.

Se necesita software de bajo nivel para las E/S en la red, pero ello no cambia esencialmente la estructura del SO.

El usuario conoce la existencia de múltiples computadores y debe trabajar con ellos de forma explícita:

El usuario conoce dónde se ubican los archivos y dónde se ejecutan los procesos.

#### **SO distribuido:**

Es un sistema operativo común compartido por una red de computadores.

A los usuarios les parece un sistema operativo tradicional con una sola máquina, pero les proporciona acceso transparente a los recursos de diversas máquinas:

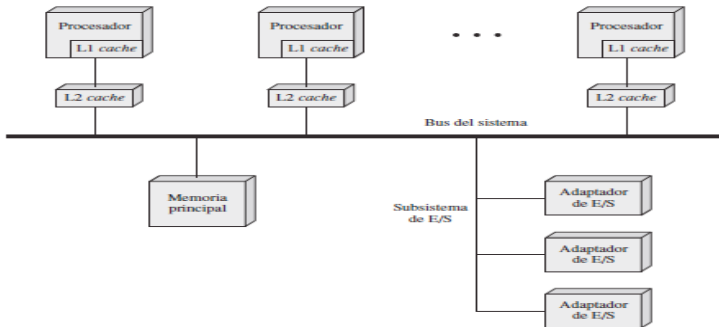
El usuario no conoce dónde se ubican los archivos ni dónde se ejecutan los procesos.

Característica: transparencia

6

### Sistemas operativos paralelos

(Stallings 4.2) Nos situamos en un sistema multiprocesador en que el objetivo es aumentar la velocidad de ejecución al aumentar el nº de procesadores (Figura 4.9 de Stallings):



Proporciona primitivas básicas para dividir una tarea en múltiples actividades paralelas.

Proporciona una comunicación y sincronización eficiente entre esas actividades

7

### 3. ESTRUCTURA DE UN SISTEMA OPERATIVO.

(Stallings 4.3; Silberschatz 2.7; Tanenbaum 1.7.1-1.7.5)

#### 3.1 Sistemas monolíticos

Todo el SO se ejecuta como un único ejecutable en modo kernel

El SO se escribe como una colección de procedimientos enlazados entre sí en un único programa binario ejecutable extenso.

Cada procedimiento del sistema tiene la libertad de llamar a cualquier otro.

Todos los elementos comparten el mismo espacio de direcciones. No hay ocultamiento de la información.

Características:

- poco fiable: un error en una parte puede provocar la caída del sistema
- poco manejable, difícil de comprender y de mantener
- poca tolerancia a fallos
- poca extensibilidad

#### 3.2 Estructura de capas

El sistema se organiza como una jerarquía de capas, cada una construida utilizando la funcionalidad de la capa sobre la que se apoya.

8

### 3.3 Estructura cliente-servidor o estructura microkernel

Definición de “Núcleo” (Stallings, glosario):

*Parte del SO que incluye las porciones de software utilizadas más frecuentemente, tiene estas características:*

*Generalmente, el núcleo se mantiene permanentemente en memoria principal.*

*Se ejecuta en modo privilegiado*

*Frecuentemente se ejecuta con interrupciones inhibidas, o con un alto nivel de inhibición de interrupciones.*

*Responde a llamadas al sistema realizadas por los procesos e interrupciones de dispositivos.*

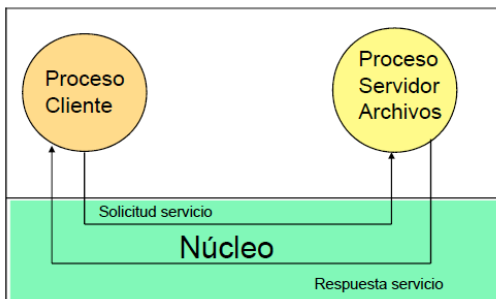
*Ha de estar altamente depurado, no puede contener errores.*

Idea central de la estructura cliente-servidor:

Reducir al mínimo la funcionalidad del núcleo e implementar las restantes partes del SO como procesos.

9

Para solicitar un servicio, el proceso de usuario (cliente) envía un mensaje al proceso servidor, que realiza el servicio y devuelve al cliente una respuesta.



Ventajas:

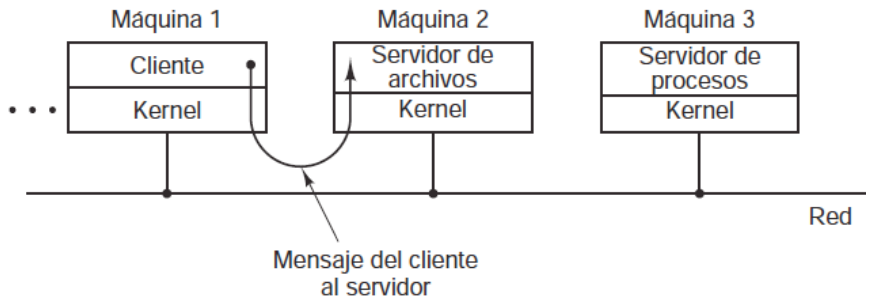
- mayor tolerancia a fallos y extensibilidad
- adaptable a sistemas distribuidos

Inconvenientes:

- peor rendimiento que las estructuras anteriores: para resolver un servicio se realizan más cambios de contexto entre procesos y se desencadenan más ejecuciones de otros servicios como paso de mensajes

10

Figura 1-25 de Tanenbaum:



**Figura 1-27.** El modelo cliente-servidor sobre una red.

### 3.4 Máquinas virtuales

(Se propondrá la realización de un trabajo personal sobre esta materia, por lo que este punto no se incluye en este documento –ni entra en examen–)