



Sesión 3: Problema de Diseño

TOPICOS ESPECIALES SISTEMAS OPERATIVOS

DOCENTE

Ing. Jorge Bravo

<https://sites.google.com/site/jorgedavidbravoe>



UNIVERSIDAD NACIONAL
DE TRUJILLO
*Rumbo a la
Excelencia Académica*



Escuela de
INFORMÁTICA

INDICE

TEMAS

- PROBLEMA EN EL DISEÑO
- DISEÑO SO VS DISEÑO SW
- DISEÑO INTERFACES
- TRABAJO GRUPAL 02



PROBLEMA

EN EL DISEÑO DE SISTEMAS OPERATIVOS

PROBLEMA DE DISEÑO

- La Ley de Moore enuncia que el hardware de computadora mejora por un factor de 100 cada década,
- pero no hay ley alguna que declare el mejoramiento de los sistemas operativos en un factor de 100 cada diez años... o que siquiera mejoren.
 - De hecho, se puede considerar que algunos de ellos son peores, en ciertos sentidos clave (como la confiabilidad), que la versión 7 de UNIX, de la década de 1970.

¿Por qué?

- La mayor parte de las veces la inercia y el deseo de obtener compatibilidad inversa tienen la culpa, y también el no poder adherirse a los buenos principios de diseño.
- Pero hay más que eso. Los sistemas operativos son esencialmente distintos en ciertas formas de los pequeños programas de aplicación que se venden en las tiendas por 49 (dólares).
- Vamos a ver ocho de las cuestiones que hacen que sea mucho más difícil diseñar un sistema operativo que un programa de aplicación.

DISEÑO DE SISTEMAS

SISTEMAS OPERATIVOS VS SOFTWARE GENERAL

1 Los sistemas operativos se han convertido en programas extremadamente extensos.

- Ninguna persona se puede sentar en una PC y escribir un sistema operativo serio en unos cuantos meses.
 - Todas las versiones actuales de UNIX sobrepasan los 3 millones de líneas de código.
 - Windows Vista tiene más de 5 millones de líneas de código del kernel (y más de 70 millones de líneas de código en total). Nadie puede entender de 3 a 5 millones de líneas de código, mucho menos 70 millones.
- Cuando tenemos un producto que ninguno de los diseñadores puede esperar comprender por completo, no debe sorprender que los resultados estén con frecuencia muy alejados de lo óptimo.

1 Los sistemas operativos se han convertido en programas extremadamente extensos.

- Los sistemas operativos no son los más complejos de todos.
- Por ejemplo, las empresas de transporte aéreo son mucho más complicadas, pero se pueden particionar en subsistemas aislados para poder comprenderlos mejor.
 - Las personas que diseñan los inodoros en una aeronave no tienen que preocuparse por el sistema de radar.
 - Los dos subsistemas no tienen mucha interacción.
- En un sistema operativo, el sistema de archivos interactúa a menudo con el de memoria en formas inesperadas e imprevistas.

2 Concurrency

- Los sistemas operativos tienen que lidiar con la concurrencia.
- Hay varios usuarios y dispositivos de E/S activos al mismo tiempo.
- En esencia, es mucho más difícil administrar la concurrencia que una sola actividad secuencial.
- Las condiciones de carrera y los interbloqueos son sólo dos de los problemas que surgen.

3 Seguridad

- En tercer lugar, los sistemas operativos tienen que lidiar con usuarios potencialmente hostiles que desean interferir con la operación del sistema o hacer cosas prohibidas, como robar los archivos de otro usuario.
- El sistema operativo necesita tomar las medidas necesarias para evitar que estos usuarios se comporten de manera inapropiada.
- Los programas de procesamiento de palabras y los editores de fotografías no tienen este problema.

4 Compartir Recursos

- En cuarto lugar, a pesar del hecho de que no todos los usuarios desconfían de los otros, muchos de ellos desean compartir parte de su información y recursos con otros usuarios seleccionados.
- El sistema operativo tiene que hacer esto posible, pero de tal forma que los usuarios maliciosos no puedan interferir.
- De nuevo, los programas de aplicaciones no se enfrentan a ningún reto similar.

5 Proyección

- En quinto lugar, los sistemas operativos viven por mucho tiempo. UNIX ha estado en operación durante un cuarto de siglo; Windows, durante más de dos décadas y no muestra signos de desaparición.
- En consecuencia, los diseñadores tienen que pensar sobre la forma en que pueden cambiar el hardware y las aplicaciones en un futuro distante, y cómo deben prepararse para ello.
- Por lo general, los sistemas que están demasiado encerrados en una visión específica del mundo desaparecen.

6 Uso

- En sexto lugar, los diseñadores de sistemas operativos en realidad no tienen una buena idea sobre la forma en que se utilizarán sus sistemas, por lo que necesitan proveer una generalidad considerable.
- Ni UNIX ni Windows se diseñaron con el correo electrónico o los navegadores Web en mente, y aun así hay muchas computadoras que utilizan estos sistemas operativos y casi todo el tiempo utilizan estas dos aplicaciones.
- Nadie le dice a un diseñador de barcos cómo crear uno sin especificarle que desean un bote de pesca, un crucero o un buque de guerra.

7 Portabilidad

- En séptimo lugar, por lo general los sistemas operativos modernos están diseñados para ser portables, lo cual significa que deben ejecutarse en varias plataformas de hardware.
- También tienen que admitir miles de dispositivos de E/S, y todos están diseñados de manera independiente, sin ningún tipo de relación con los demás.
- Un ejemplo en donde esta diversidad ocasiona problemas, se podría ver de manera constante en MS-DOS, cuando los usuarios trataban de instalar, por ejemplo, una tarjeta de sonido y un módem que utilizaban los mismos puertos de E/S o las mismas líneas de petición de interrupción.
- Pocos programas además de los sistemas operativos tienen que lidiar con los problemas que ocasionan las piezas de hardware en conflicto.

8 Compatibilidad inversa

- En octavo y último lugar, está la frecuente necesidad de tener compatibilidad inversa con cierto sistema operativo anterior.
- Ese sistema puede tener restricciones en cuanto a las longitudes de las palabras, los nombres de archivos u otros aspectos que los diseñadores ya consideran obsoletos pero que deben seguir utilizando.
- Es como convertir una fábrica para producir los autos del próximo año en vez de los de este año, pero seguir produciendo los autos de este año a toda su capacidad.

DISEÑO DE INTERFACES

EN EL DISEÑO DE SISTEMAS OPERATIVOS

¿Dónde se puede empezar?

- El mejor lugar para iniciar sea pensar sobre las interfaces que va a proporcionar.
- Un sistema operativo proporciona un conjunto de abstracciones, que en su mayor parte se implementan mediante tipos de datos (por ejemplo, archivos) y las operaciones que se realizan en ellos (por ejemplo, read).
- En conjunto, estos dos elementos forman la interfaz para sus usuarios.
- Hay que considerar que en este contexto, los usuarios del sistema operativo son programadores que escriben código que utiliza llamadas al sistema, y no personas que ejecutan programas de aplicación.

Interfaces Adicionales

- Además de la interfaz principal de llamadas al sistema, la mayoría de los sistemas operativos tienen interfaces adicionales. Por ejemplo, algunos programadores necesitan escribir drivers de dispositivos para insertarlos en el sistema operativo. Estos drivers ven ciertas características y pueden realizar llamadas a ciertos procedimientos.
- Estas características y llamadas también definen una interfaz, pero es muy distinta de la que ven los programadores de aplicaciones.
- Todas estas interfaces se deben diseñar con cuidado, para que el sistema tenga éxito.