

SISTEMAS OPERATIVOS

Arquitectura de un sistema operativo comercial



Lorena Cáceres Arias
Jesús García León
Javier Gómez López
José Alberto Hoces Castro

Índice

- *1. Introducción al sistema operativo ReactOS*
 - ◆ *Creador y propósito*
 - ◆ *Un poco de historia*
 - ◆ *Versión 0.4.13 (Particularidades)*
- *2. Historia de ReactOS*
- *3. Arquitectura de ReactOS*

1. INTRODUCCIÓN

Un sistema operativo es un conjunto de programas que habilita una administración eficaz de los recursos de un ordenador. El sistema operativo comienza a trabajar apenas se enciende el equipo al tener que gestionar el hardware desde los niveles más básicos. A lo largo de la historia, múltiples sistemas operativos han sido desarrollados con diferentes propósitos. Algunos de los más conocidos son Windows, Linux y MacOS. Sin embargo, existen muchos otros más de poca popularidad como Haiku, AROS, Visopsys, etc. En este documento nos centraremos en el estudio del sistema operativo ReactOS, concretamente en su última versión (0.4.13).

Creador y propósito

El fundador del proyecto es Jeff Knox. Tal y como pone en la página web oficial de ReactOS, el objetivo de este proyecto es proporcionar un sistema operativo que sea compatible a nivel binario con Windows, de forma que los usuarios acostumbrados a la interfaz de Windows puedan hacer uso de ReactOS sin problema. De hecho, la palabra *react* se debe a la insatisfacción y la reacción al monopolio de Microsoft. Así se le abrió la posibilidad de renunciar al uso de software propietario comercial (sin acceso libre a su código fuente) sin necesidad de cambiar a Linux a todos los usuarios de Windows que se encontraban cómodos con dicha interfaz.

Versión 0.4.13 (Particularidades)

Esta es la última versión de ReactOS, la cual fue lanzada el 9 de abril de 2020. Las diferencias con las anteriores versiones son una serie de mejoras:

- ❖ El sistema USB contaba con errores, por lo que fue reescrito para esta versión, consiguiendo soporte HID y dispositivos de almacenamiento.
- ❖ Búsqueda de archivos en el shell gráfico, de forma que se puede conocer la ubicación exacta de un archivo sin necesidad de buscarlo manualmente.
- ❖ Mayor velocidad en FreeLoader (cargador de arranque de ReactOS) en particiones FAT y en el puerto Xbox (permite cargar ReactOS en dicha consola).
- ❖ Aparición de un Accessibility Utility Manager, que maneja la accesibilidad de las aplicaciones

La próxima versión será la 0.5, cuya fecha de lanzamiento está aún por determinar y trae mejoras como soporte de escritura para NTFS, soporte para drivers WDM, impresoras y DirectX.

2. Historia de ReactOS

El objetivo principal del proyecto ReactOS es proporcionar un sistema operativo que sea compatible a nivel binario con Windows... de tal manera que las personas acostumbradas a la familiar interfaz de usuario de Windows la encontrarán sencilla usando ReactOS. El objetivo final de ReactOS es permitir quitar Windows e instalar ReactOS sin que el usuario final note el cambio.

En el año 1996 un grupo de programadores y desarrolladores de free-software comenzaron con el proyecto FreeWin95 que consistía en implementar un clon de Windows 95. El proyecto arrancó con grandes expectativas pero tras dos años de trabajo no había ninguna release de el proyecto.

Se decidió darle un nuevo impulso, rebautizándolo como ReactOS en 1998, como crítica al monopolio de sistemas operativos de Microsoft de la época.

Sufrieron varios altibajos, como el proyecto Ekush OS. En 2004 se encontró una violación de derechos de autor cuando alguien distribuyó una copia de ReactOS bajo el nombre Ekush OS.

Para evitar problemas de derechos de autor, en 2006 el director del proyecto declaró que la línea del proyecto había dado un rumbo radical, presentando ReactOS como una alternativa a Windows y no un derivado de este. Tras ciertas auditorías internas, el grupo fue demandado y dejado de lado por algunas compañías, como la propia Microsoft o como Wine.

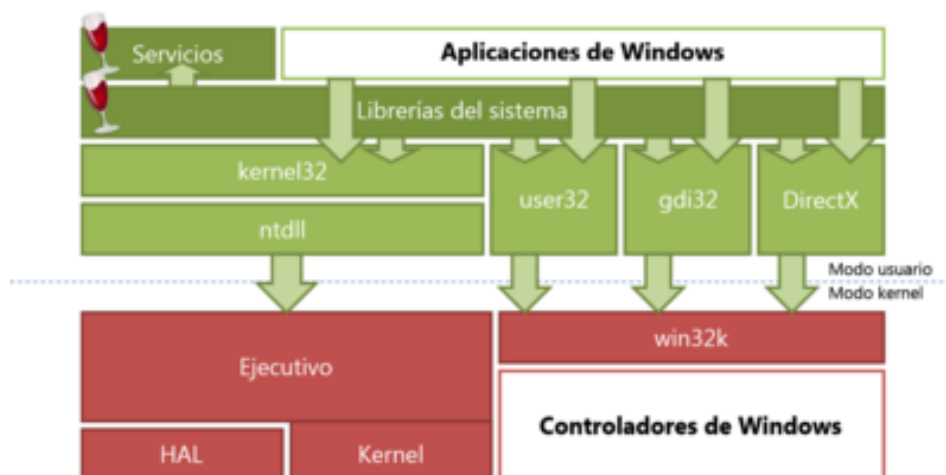
Esto generó un gran rechazo en la comunidad de software libre y debido a ello el proyecto se pudo retomar. La mayor parte del código ensamblador, que al parecer fue copiado, ha sido sustituido como una progresión natural en el desarrollo de ReactOS, con los desarrolladores habiendo reimplementado la funcionalidad en C por razones de portabilidad.

Desde entonces, el proyecto trata de financiarse por distintos medios como el Proyecto Thorium Core Cloud Desktop, algún Hackfest o participaciones en Google Summer of Code.

3. Arquitectura de ReactOS

En este apartado vamos a estudiar la arquitectura de este SO, y tras analizarla podremos realizar una comparación con lo estudiado en clase para clasificarla como monolítico o mononúcleo.

Esquema

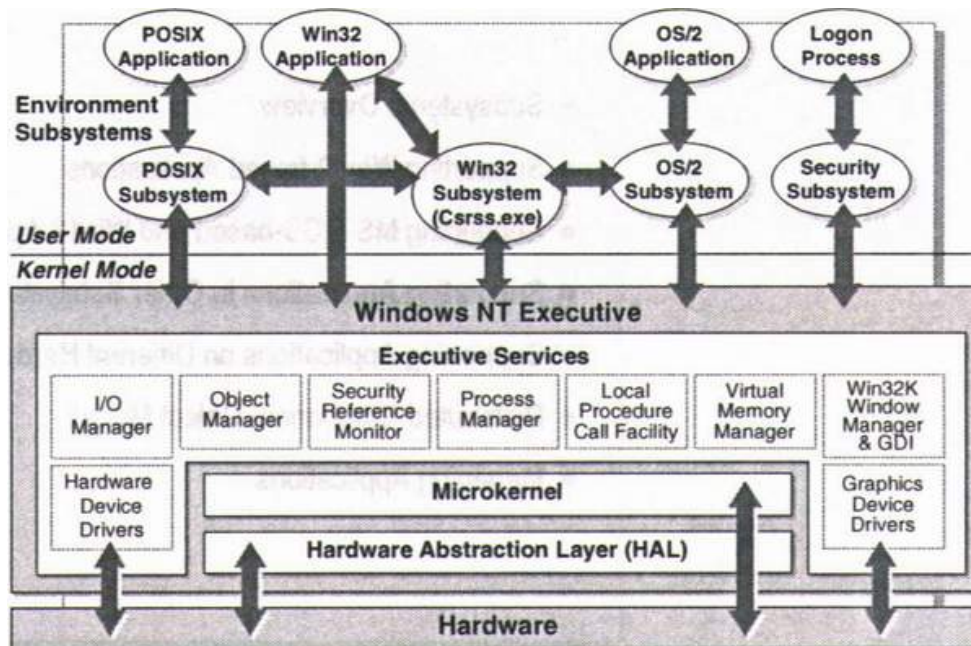


En el esquema vemos los distintos elementos del modo usuario y el modo kernel.

En la parte del kernel podemos observar elementos como el HAL (Hardware Abstraction Level), el Kernel, los drivers, etc.

En la zona del usuario tenemos que destacar la colaboración de Wine, un proyecto con objetivo similar con el que comparte partes del código para lograr ejecutar aplicaciones de Windows. En esta parte también se encuentran las librerías del sistema y módulos como user32 o DirectX.

Podemos ver que es prácticamente igual a la de windows, puesto que busca ser una especie de sustituto de este.



Clasificación:

Viendo lo similar que es a Windows, no es extraño que podamos colocarlos en la misma categoría a ambos. A grandes rasgos, podemos decir que son sistemas microkernel, en concreto microkernel híbrido, que incorporan parte de código no esencial en el kernel por términos de eficiencia. Pues asignan pocas funciones esenciales al núcleo, como espacios de direcciones, comunicación entre procesos y planificación básica, pero el grueso de los servicios se ejecutan en modo usuario.