

Sistemas Operativos y Redes

Jurgen Heysen



¿Por qué estudiar Sistemas Operativos?

Porque definen en qué formas podemos **interactuar** con los computadores para realizar distintas tareas

Ofrecen **interfaces** que ayudan a no preocuparse de todos los detalles de bajo nivel con ciertos temas

El software que uno construye es **ejecutado y administrado** por el Sistema Operativo

Sistemas Operativos permiten

Interactuar con piezas del computador usando una interfaz conocida y uniforme

Administrar la memoria del computador

Ejecutar más de un programa a la vez

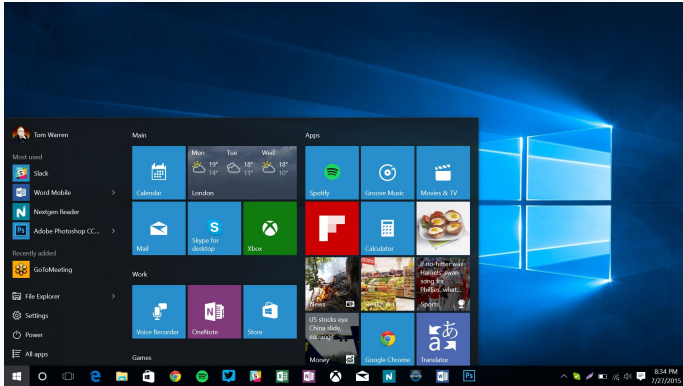
Componentes de un Sistema Operativo

Interfaces de usuario

Servicios de Sistema

Kernel

Interfaces de Usuario



```
cruz@hercules:~$ cd iic2333/  
cruz@hercules:iic2333$ ls  
examples  
cruz@hercules:iic2333$ cd examples/  
cruz@hercules:examples$ ls  
fork2.cpp fork3.c fork.c shmem-cons.c shmem-prod.c waitZombies.c  
cruz@hercules:examples$
```

Servicios de Sistema

Corresponde a **interfaces de software** y herramientas que entrega el sistema operativo para que el software que opera con él no deba **repetir procesos** que son muy comunes

Por ejemplo, administrador de tareas, servicio de audio, manejo de sistema de archivos, comunicación por red

Kernel

“El verdadero” Sistema Operativo

Programa con **acceso completo** al hardware que **gobierna la ejecución** de **todos** los programas en el computador

Se encarga de coordinar los distintos programas en ejecución de modo tal que todos puedan ser **atendidos** por la CPU

Se encarga de mantener los **espacios de memoria** de todos los programas

Concepto clave: Llamada a Sistema (**Syscall**)

Proceso

Unidad de asignación de recursos del Sistema Operativo

Además de las instrucciones que componen el programa que se ejecuta contiene información de estado

Contiene información administrativa para el Sistema Operativo

Thread

Corresponde a una división dentro de un proceso que comparte memoria con otros Threads del mismo proceso

Representa el estado de ejecución del programa

Scheduling

Muchos procesos (y Threads) quieren utilizar los recursos del computador **al mismo tiempo**

En muchos casos la sumatoria de los recursos solicitados **excede** a la capacidad máxima que aguanta el computador

La **única** forma de poder atenderlos a todos es ingresando a ejecución ordenadamente los hilos de ejecución, de modo que la cantidad de recursos requerida **nunca** exceda la capacidad máxima del computador en este caso.

Filesystem

Una operación común es querer leer y guardar datos, y el Sistema Operativo se encarga de proveer este funcionamiento

Los datos guardados en disco queremos leerlos con alguna **estructura** y **ordenados** de alguna forma

Dicho servicio es provisto como el Filesystem (Sistema de Archivos)

Ejemplos: EXT4, APFS, NTFS, FAT

Redes

Por qué estudiar redes

Hoy por hoy todo el software funciona con red de alguna forma

Porque las redes son parte trascendental de nuestras vidas

Para comprender cómo funciona la red más grande de todas: [Internet](#)

Receta para hacer redes

Tener un **medio físico** por el que transmitir **señales**

Tener **dispositivos input/output** en los computadores **miembros de la red** que les permitan **enviar y recibir** señales en este medio

Tener un **acuerdo** sobre cómo enviar señales en el medio e interpretarlo como **string binario**

Tener acuerdo sobre **cómo leer** este string binario

Ejemplo: WiFi

Medio físico: Aire

Dispositivos Input/Output: Antenas

Acuerdo sobre cómo enviar las cosas: IEEE 802.11

Acuerdo sobre cómo leer los datos: IP

Muchas partes a distintos niveles

Podemos ver que hay **separaciones** en la receta que son marcadas

Cada separación requiere un **acuerdo distinto** de cómo hacer las cosas

Potencialmente, esa decisión es **independiente** del resto

Redes se estructuran por **capas**

Modelo OSI de capas

1. Física: Transmisión binaria de los datos
2. Enlace: Acceso a los medios
3. Red: Direccionamiento y ruteo de los datos
4. Transporte: Conexión de extremo a extremo
5. Sesión: Comunicación entre hosts
6. Presentación: Representación de los datos
7. Aplicación: Procesos de red a aplicaciones

Ejemplo: Conexión a webserver en LAN

1. Física: Cable UTP
2. Enlace: Ethernet
3. Red: IPv4
4. Transporte/Sesión/Presentación: TCP
5. Aplicación: HTTP

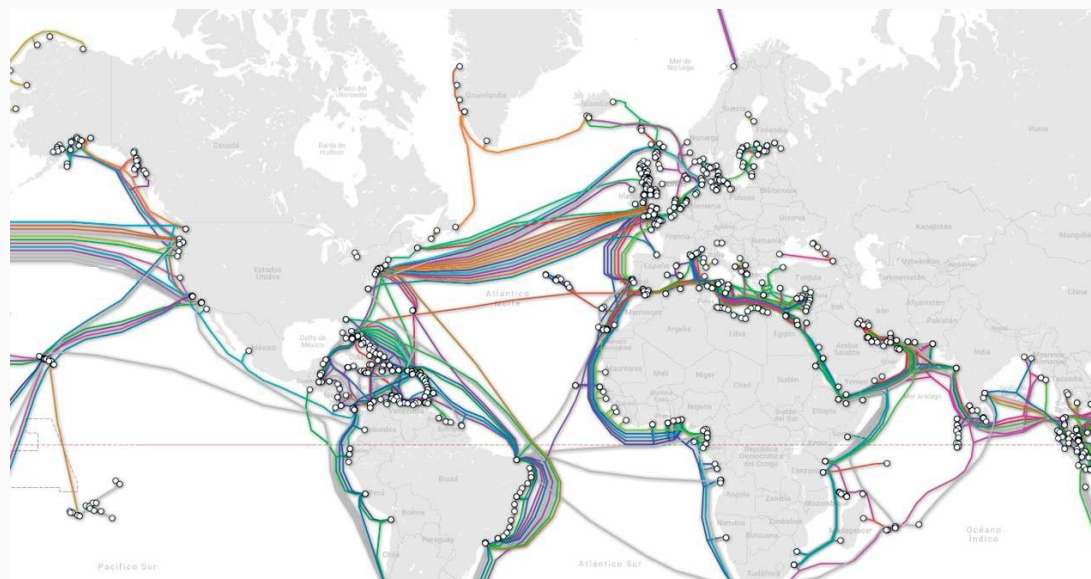
¿Y si el webserver está en internet?

Debemos cruzar múltiples redes para alcanzarlo

Para cada red que cruzamos, se han definido los puntos de la receta

No todas las redes comparten necesariamente los mismos puntos de la receta

Esfuerzo de estandarización logra que todos se entiendan e internet funcione



Debemos cruzar múltiples redes

Para que internet funcione necesitamos cruzar por múltiples redes, que deben poder saber quiénes son sus vecinos

Aparece el concepto de [Sistema Autónomo](#)

Se establecen protocolos de comunicación entre Sistemas Autónomos

Sistemas Autónomos

Redes que pertenecen a [una organización en particular](#)

Identificados por el [ASN](#), que es entregado por ICANN

Se comunican entre sí en lugares conocidos como [Puntos de Intercambio de Tráfico](#)

Un protocolo domina la comunicación entre ellos: [BGP](#)

Sistemas Operativos y Redes

Jurgen Heysen

