

# Resumen Introducción

Tomás Felipe Melli

Noviembre 2024

## Índice

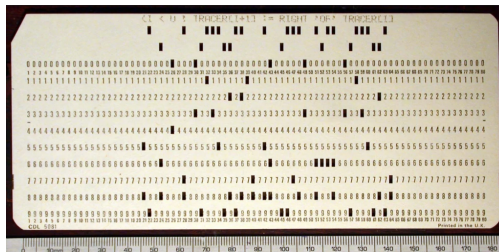
<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Historia</b>	<b>2</b>
2.1	Tarjetas Perforadas . . . . .	2
2.2	Sistema Batch . . . . .	2
2.3	Multiprogramación . . . . .	2
2.4	Contención . . . . .	2
<b>3</b>	<b>De qué se compone un Sistema Operativo ?</b>	<b>3</b>
<b>4</b>	<b>Cómo está diseñado y organizado el kernel en relación con las funcionalidades del sistema operativo ?</b>	<b>4</b>

# 1 Introducción

El **Sistema Operativo** es una pieza de Software que hace de intermediario entre el Hardware y los programas de usuario. La función principal es que los programas corran correctamente y con un buen rendimiento. Para lograrlo, debe gestionar adecuadamente todos los recursos del sistema : CPU, memoria, almacenamiento, dispositivos de I/O,.. por ello corre con **máximo nivel de privilegio**.

## 2 Historia

### 2.1 Tarjetas Perforadas



La tarjeta perforada era el programa. Ejemplo :

```
1      PROGRAM SUMA
2      INTEGER A, B, C
3      A = 10
4      B = 20
5      C = A + B
6      PRINT *, C
7      END
```

Y la forma de que la compu antigua lo interpretase como tal era, perforar en una tarjeta (que representaba una línea de código) y por cada hueco, un sensor eléctrico convertía a binario. Y así se corría el programa. Normalmente estaba en FORTRAN, por lo que el compilador lo traducía a código máquina. Si el programa necesitaba DATOS, se insertaban a continuación en un lector de tarjetas. En el principio, el usuario tenía que escribir el programa, si fallaba, tenía que estar presente durante la ejecución para corregirlo, entre otras cosas. Por supuesto que el operador de la compu estaba encima para organizar las tarjetas y para detener la ejecución en caso de errores.

### 2.2 Sistema Batch

Con el sistema Batch, los usuarios no tenían que estar presentes durante la ejecución. Eso mejoró. En ese momento, se entregaban también tarjetas o luego las cintas, pero la diferencia es que ahora entregaban el **lote** y se ejecutaba secuencialmente y en caso de errores, la máquina producía un código de error. En conclusión, lo que cambió fue que el usuario no tenía que estar pendiente durante la ejecución. Sólo el operador.

### 2.3 Multiprogramación

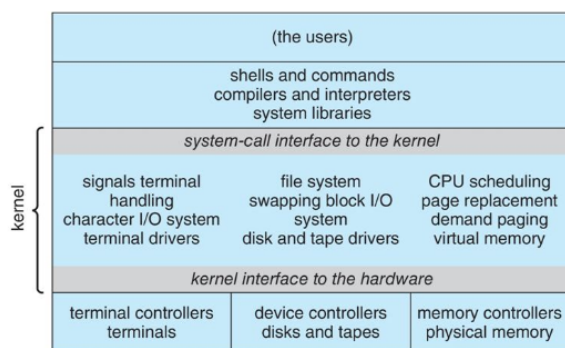
Luego de unos años, como al leer de una cinta o imprimir un resultado el proce estaba IDLE, se quería intentar aprovechar ese tiempo, ya que es caro. Esto deriva en el concepto de **Multiprogramación**. La capacidad que tiene un sistema para ejecutar tareas diferentes en el mismo procesador. Esto logra que el tiempo de una tarea en ejecutarse sea el mismo, pero que en la suma sea menor. Aumentando el THROUGHPUT del sistema.

### 2.4 Contención

Otro concepto que nace con esto anterior es la **contención**, es decir, la disputa entre varios procesos por hacer uso del mismo recurso.

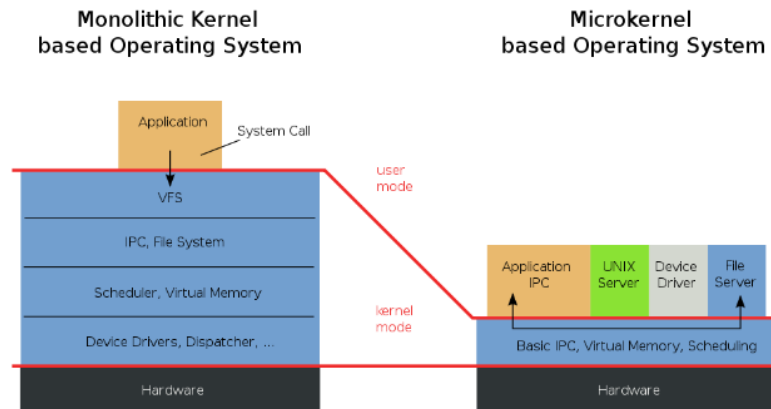
### 3 De qué se compone un Sistema Operativo ?

1. El **Núcleo o KERNEL** es el programa en sí. El mismísimo SO. Gestiona los recursos del hardware y proporciona servicios esenciales a las aplicaciones y procesos en ejecución. Se encarga de la comunicación entre el hardware y el software.
2. Los **Drivers** son programas que permiten la interacción entre dispositivos de I/O y el SO. Ellos contienen todos los detalles de bajo nivel de los dispositivos.
3. El **Shell o Intérprete de Comandos** es un programa que le permite al usuario interactuar con el SO.
4. Un **Proceso** es un programa en ejecución con su espacio de memoria asociado y ciertos atributos.
5. Un **Archivo** es una secuencia de bits con nombre y atributos que indican, entre muchas otras cosas, permisos.
6. Un **Directorio** es una colección de archivos y directorios que contiene un nombre y un orden jerárquico.
7. Un **Dispositivo Virtual** es una abstracción de un dispositivo físico bajo la forma de un archivo con la finalidad de poder abrirlo, leerlo, escribirlo, ..., entre otras cosas.
8. El **Sistema de Archivos o File System** es la forma en la que se organizan los datos en disco para gestionar su acceso.
9. Los **Directorios del Sistema** son los directorios donde el propio sistema operativo guarda los archivos para poder funcionar. Como por ejemplo /boot, /devices, ...
10. El **Binario del Sistema** es un conjunto de archivos que están dentro de los directorios del sistema que **no son parte del KERNEL**, pero tienen funciones muy importante como por ejemplo : /usr/bin/who que nos dice qué usuarios están sesionados en el sistema.
11. Los **Archivos de Configuración** son archivos de los cuales el sistema operativo saca de información necesaria para funcionar. Como por ejemplo, /etc/passwd.
12. Un **Usuario** es la representación, dentro del propio SO, de las personas o entidades que pueden usarlo. Sirve principalmente como una forma de aislar información entre sí y de establecer limitaciones.
13. Un **Grupo** es una colección de usuarios.



## 4 Cómo está diseñado y organizado el kernel en relación con las funcionalidades del sistema operativo ?

Existen dos estructuras diferentes :



La diferencia reside en el espacio de memoria en el que se ejecutan las funcionalidades del SO. El **Monolítico** es aquel en el que todas las funcionalidades se ejecutan en un único espacio de memoria. Por el contrario, el **Micro-Kernel** surgió para contrarrestar algunos de los problemas que puede tener un monolítico como que se caiga todo el sistema si falla un servicio, o mismo para reducir la cantidad de código privilegiado. Entonces se propuso una estructura en la que el kernel sólo se encargue del *manejo básico de I/O y Memoria* y la *IPC*; el resto, son servicios. En la práctica, los **Micro-Kernels** resultaron más lentos por la excesiva comunicación entre el Kernel y los servicios del espacio del usuario.