

---

# Sistemas Operativos I

*“LILO 21.6 Loading linux.....”*

Clase: Introducción

Rafael Ignacio Zurita <[rafa@fi.uncoma.edu.ar](mailto:rafa@fi.uncoma.edu.ar)>

---

**Advertencia:** Estos slides traen ejemplos.

**No copiar (ctrl+c) y pegar en un shell o terminal los comandos aquí presentes.**

**Algunos no funcionarán, porque al copiar y pegar también van caracteres “ocultos” (no visibles pero que están en el pdf) que luego interfieren en el shell.**

**Sucedio en vivo :)**

**Conviene “escribirlos” manualmente al trabajar.**

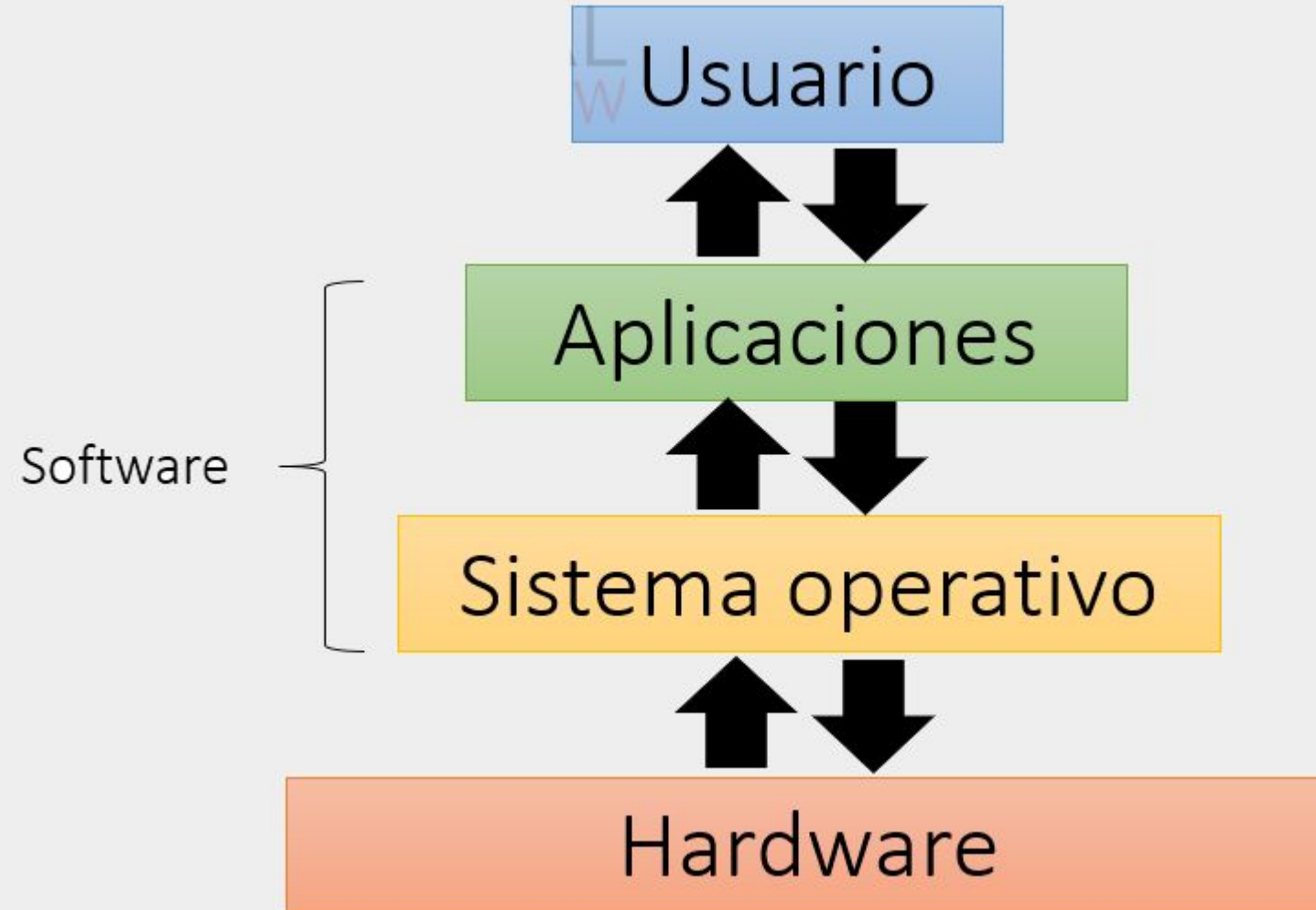
## Contenido:

- **Detalles administrativos**
- **Sistema de computación**
- **Qué es un Sistema Operativo**
- **Servicios de los SO**
- **System Calls**
- **Tipos de Sistemas Operativos**
- **Estructura Interna**

# Sistemas Operativos I

---

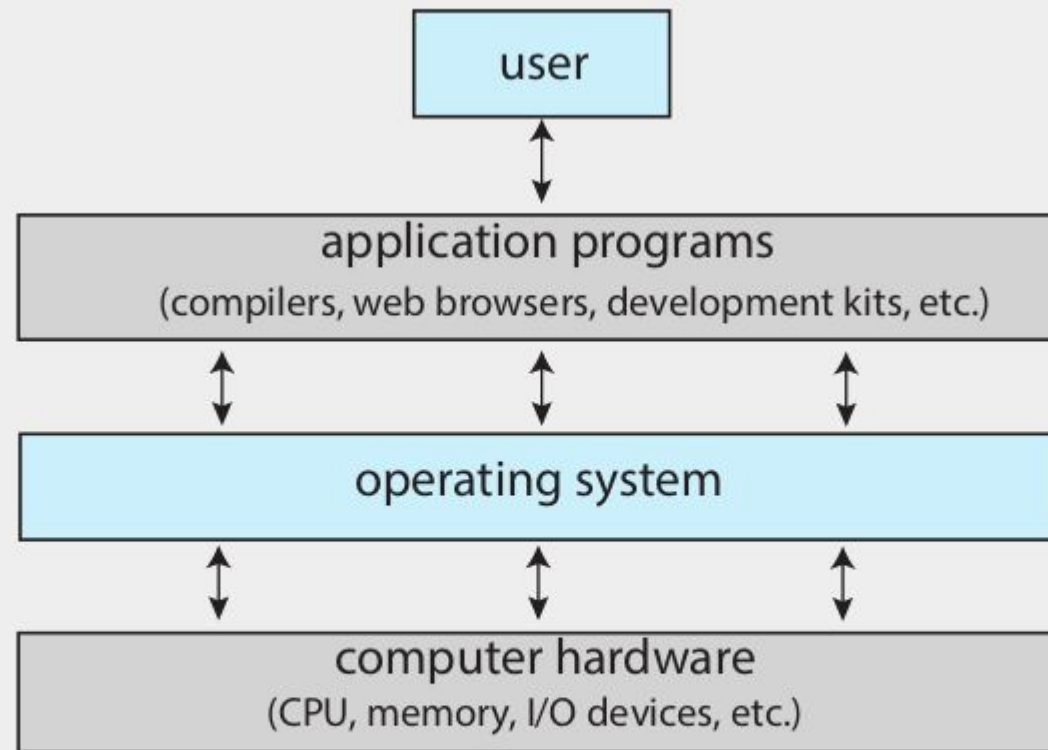
## Sistema de Computación



# Sistemas Operativos I

---

## Sistema de Computación



**Figure 1.1** Abstract view of the components of a computer system.

# Sistemas Operativos I

---

## ¿Qué es un sistema operativo?

Un sistema operativo (SO) **es el conjunto de programas** de un sistema informático **que gestiona los recursos de hardware y provee servicios** a los programas de aplicación. (by Wikipedia)

---

Is a software which performs two essentially unrelated functions:

- providing application programmers (and application programs, naturally) a clean abstract set of resources instead of the messy hardware ones and
  - managing these hardware resources. (by Andrew Tanenbaum)
- 

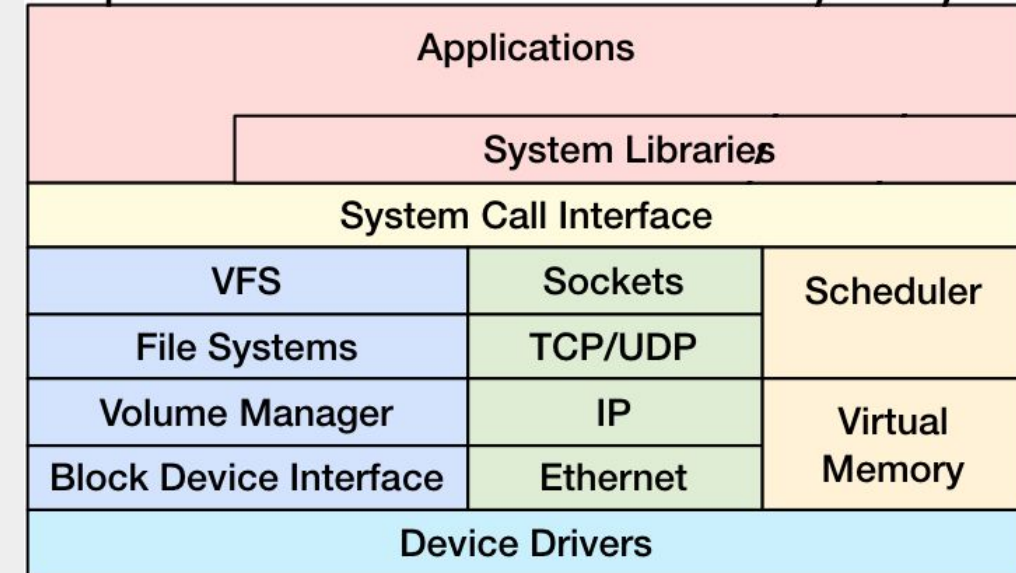
The fundamental goal of computer systems is to execute programs and to make solving user problems easier. Computer hardware is constructed toward this goal. Since bare hardware alone is not particularly easy to use, application programs are developed. These programs require certain common operations, such as those controlling the I/O devices. The common functions of controlling and allocating resources are then brought together into one piece of software: the operating system. (by Abraham Silberschatz)

# Sistemas Operativos I

---

## Componentes de un Sistema Operativo

- Kernel
- Shell o Interfaz de Usuario GRÁFICA
- Programas de sistema  
(que se encuentran ligado al sistema operativo: sysadmin, compiladores, bibliotecas)



# Sistemas Operativos I

---

## Funciones o metas de un Sistema Operativo

### Asignación de recursos

**El SO administra y asigna recursos con ciertos criterios.**

### Abstracción

**El SO oculta los detalles de bajo nivel del hardware, y presenta servicios de alto nivel.  
(Ej: disco y archivo)**

### Control (protección, privacidad, seguridad, etc)

**El SO supervisa constantemente la ejecución de los programas de usuario y el uso de los recursos para detectar posibles fallas y/o errores.**



## Servicios PRINCIPALES un sistema operativo

**Gestión de Procesos**

**Gestión de Memoria**

**Gestión de la E/S**

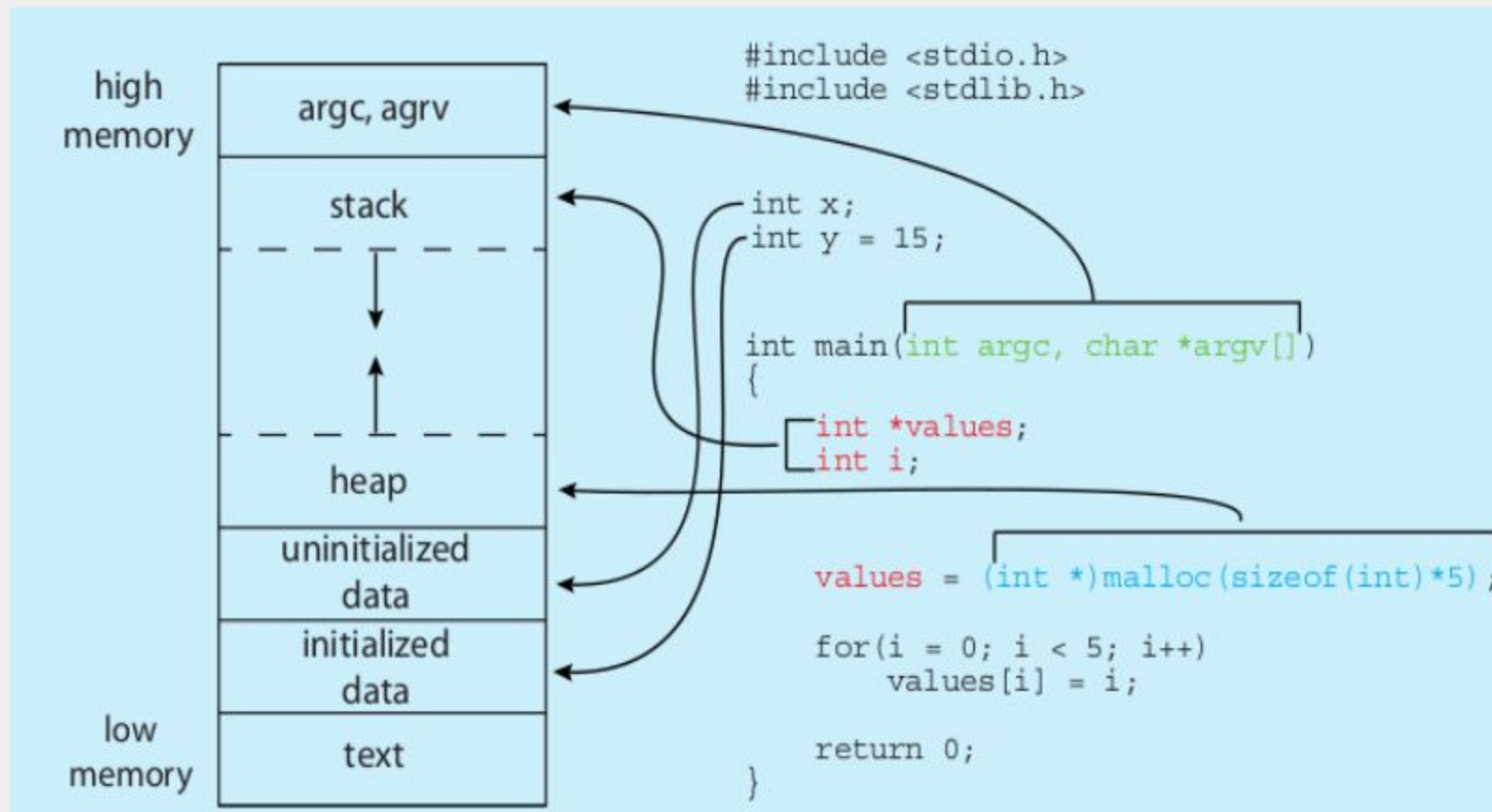
**Sistemas de Archivos**

**Protección**

**Gestión de Redes**

# Sistemas Operativos I

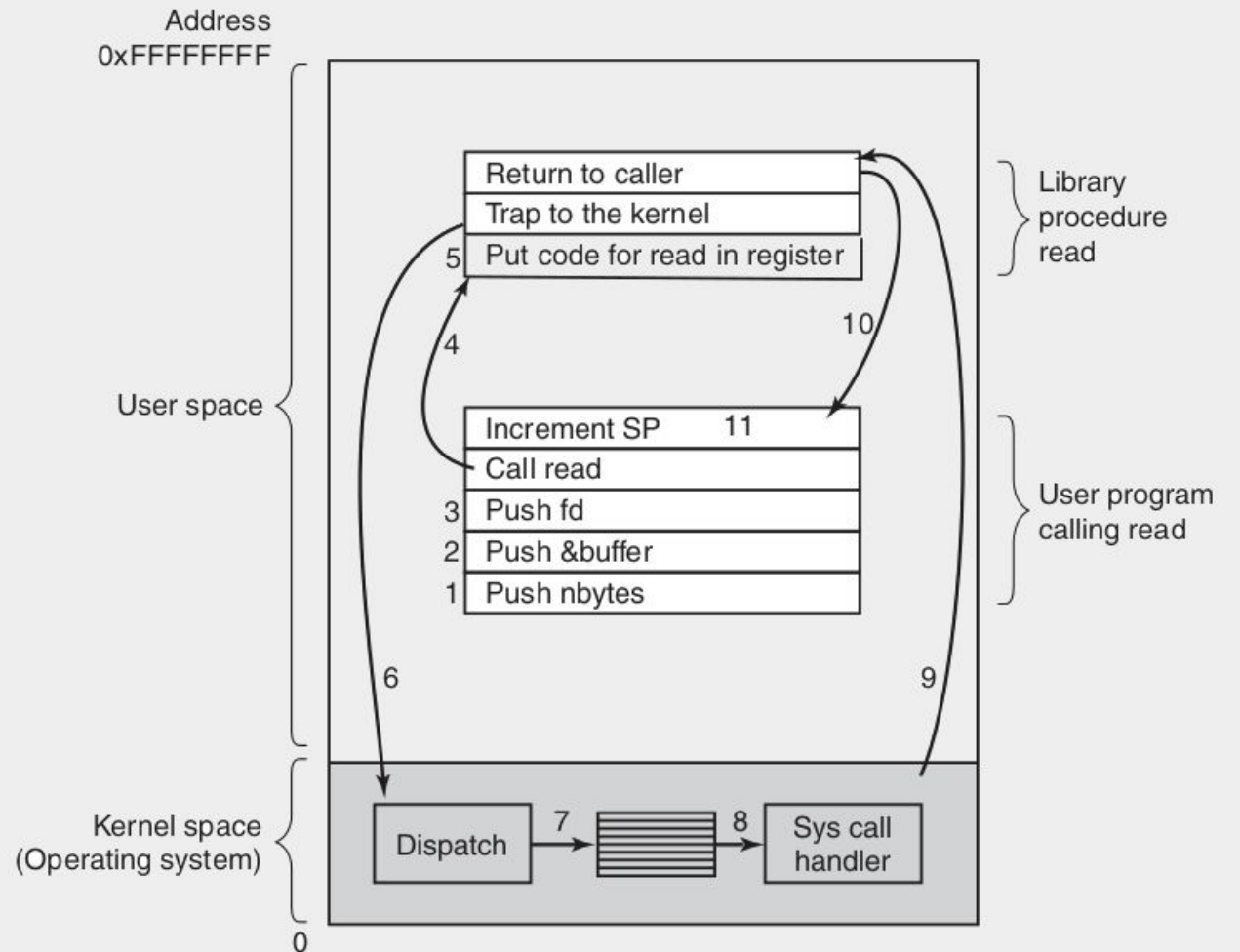
## Introducción a System Calls



## Llamada al Sistema (syscall o system call)

Ejemplo:

```
c = read(fd, buffer, nbytes);
```



# Sistemas Operativos I

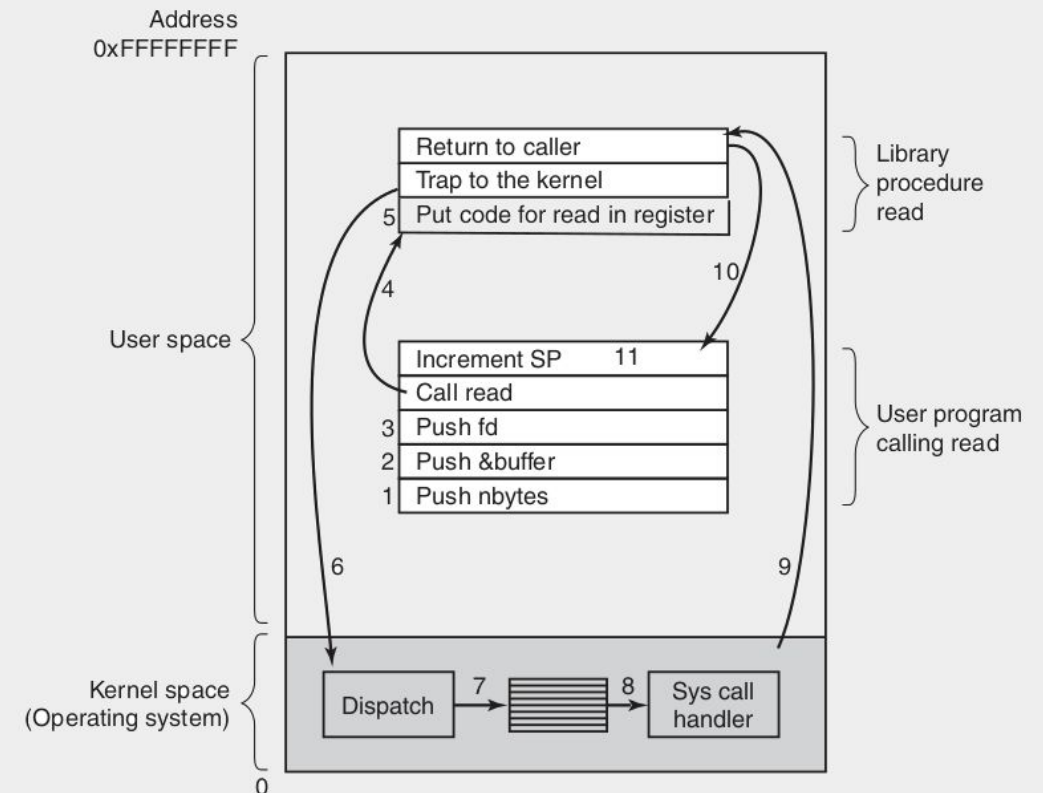
## Llamada al Sistema (syscall o system call)

### Ejemplo:

```
c = read(fd, buffer, nbytes);
```

### Soporte del hardware:

- CPU con 2 modos: **supervisor y usuario**
- **Instrucciones y registros especiales**
- **Interrupciones (exepciones/trap)**
- **Memoria Virtual**



# Sistemas Operativos I

---

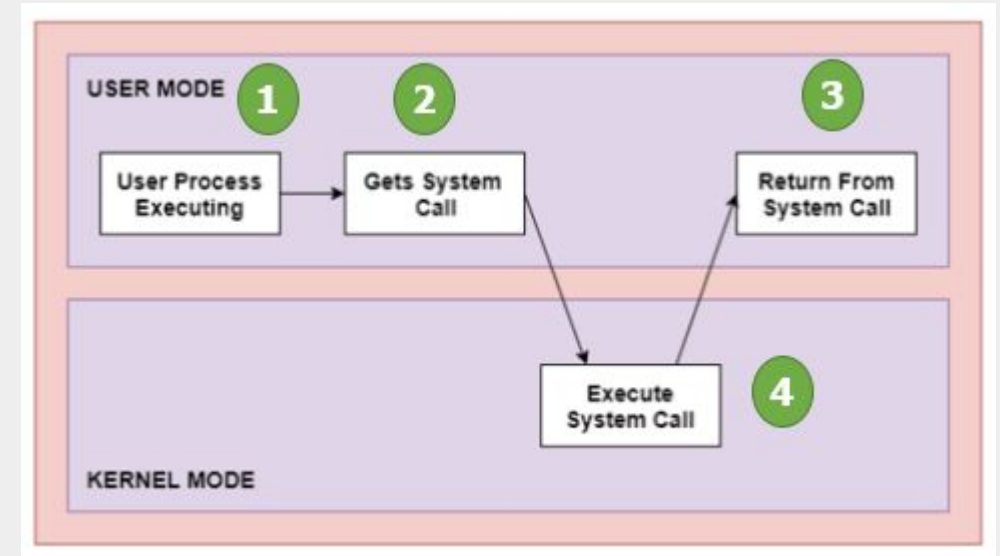
## Llamada al Sistema (syscall o system call)

### Ejemplo:

```
c = read(fd, buffer, nbytes);
```

### Soporte del hardware:

- CPU con 2 modos: **supervisor y usuario**
- **Instrucciones y registros especiales**
- **Interrupciones (exepciones/trap)**
- **Memoria Virtual**



# Sistemas Operativos I

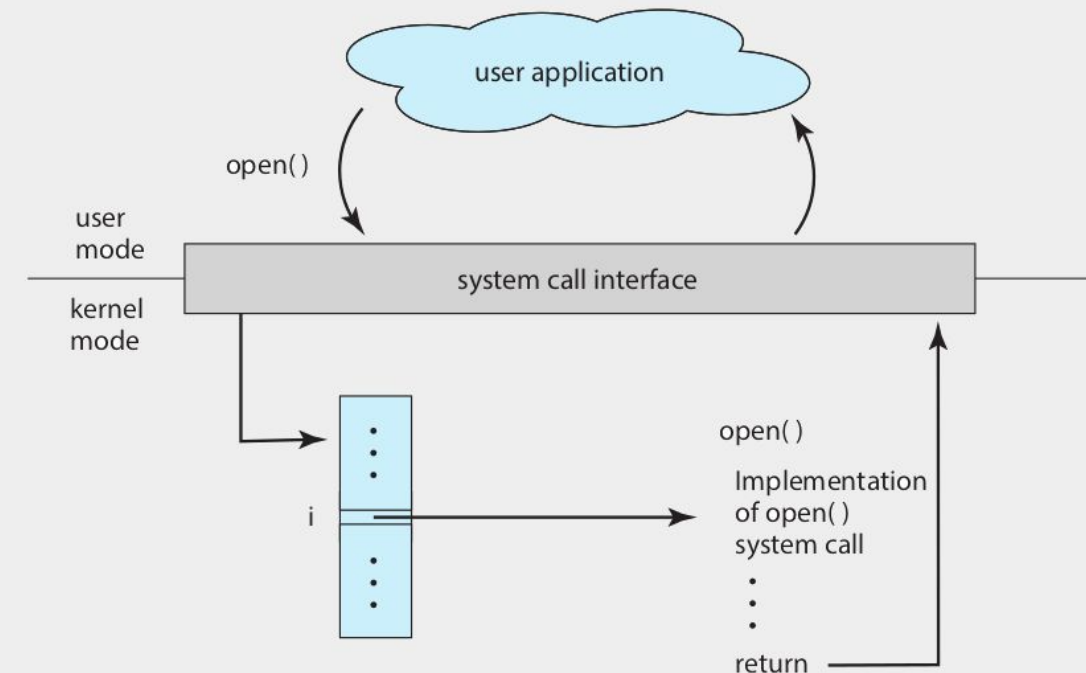
## Llamada al Sistema (syscall o system call)

### Ejemplo:

```
c = read(fd, buffer, nbytes);
```

### Soporte del hardware:

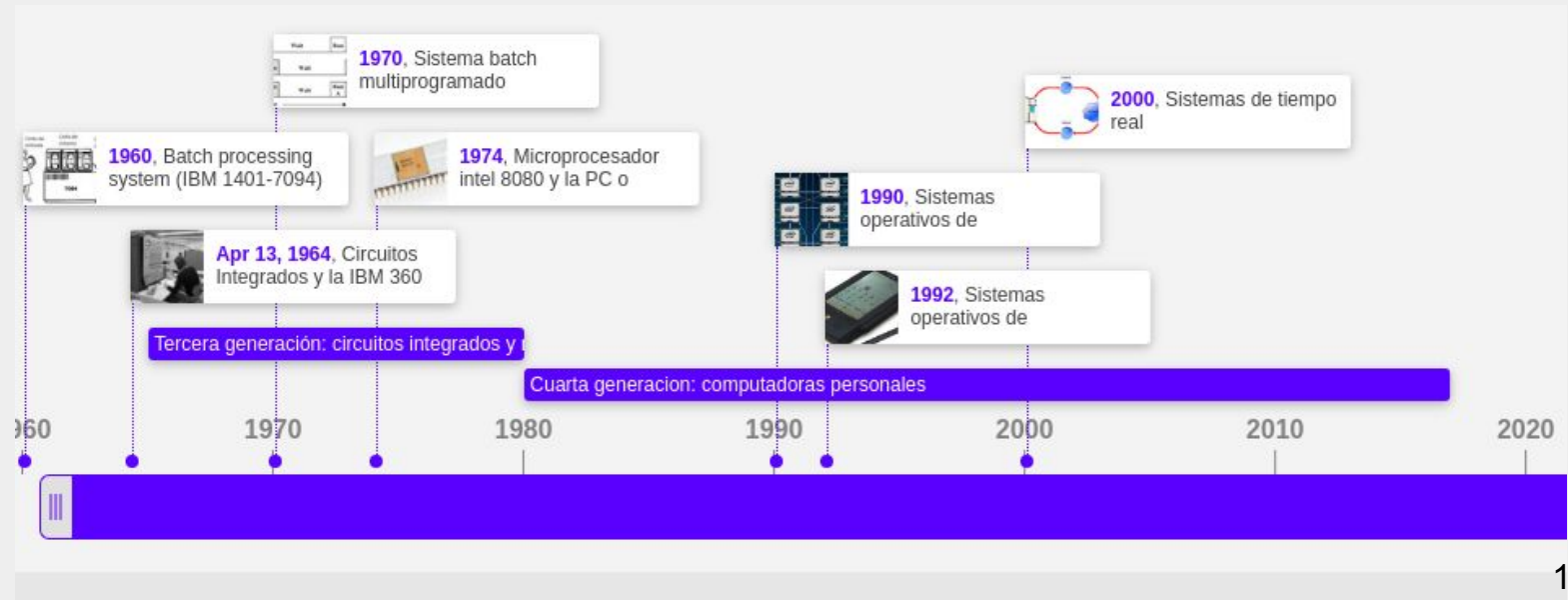
- CPU con 2 modos: **supervisor y usuario**
- **Instrucciones y registros especiales**
- **Interrupciones (exepciones/trap)**
- **Memoria Virtual**



# Sistemas Operativos I

## Evolución de los Sistemas Operativos

- Máquina sin S.O. (década del 50)
- Sistemas batch simples
- Sistemas batch multiprogramados
- Sistemas de tiempo compartido
- Sistemas desktop
- Sistemas multiprocesadores
- Sistemas de tiempo real
- Sistemas Embebidos

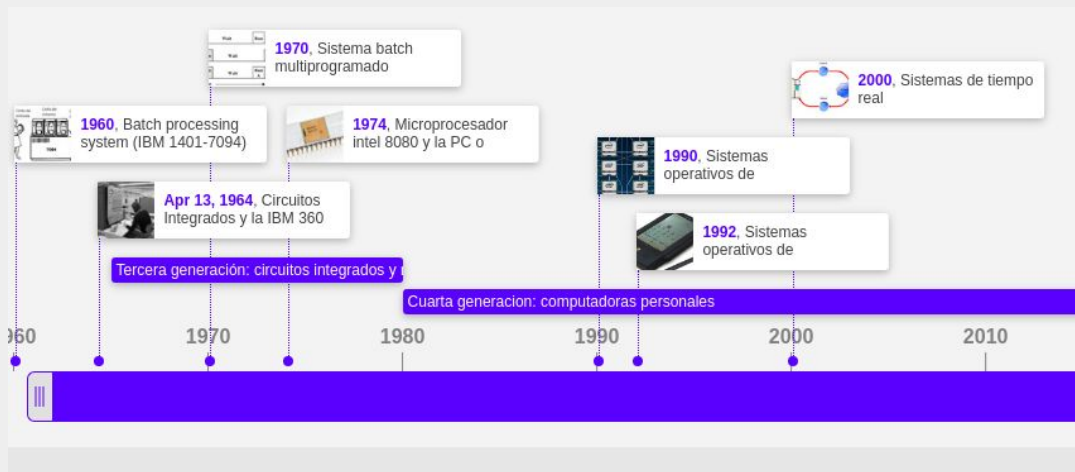




# Sistemas Operativos I

## Evolución de los Sistemas Operativos

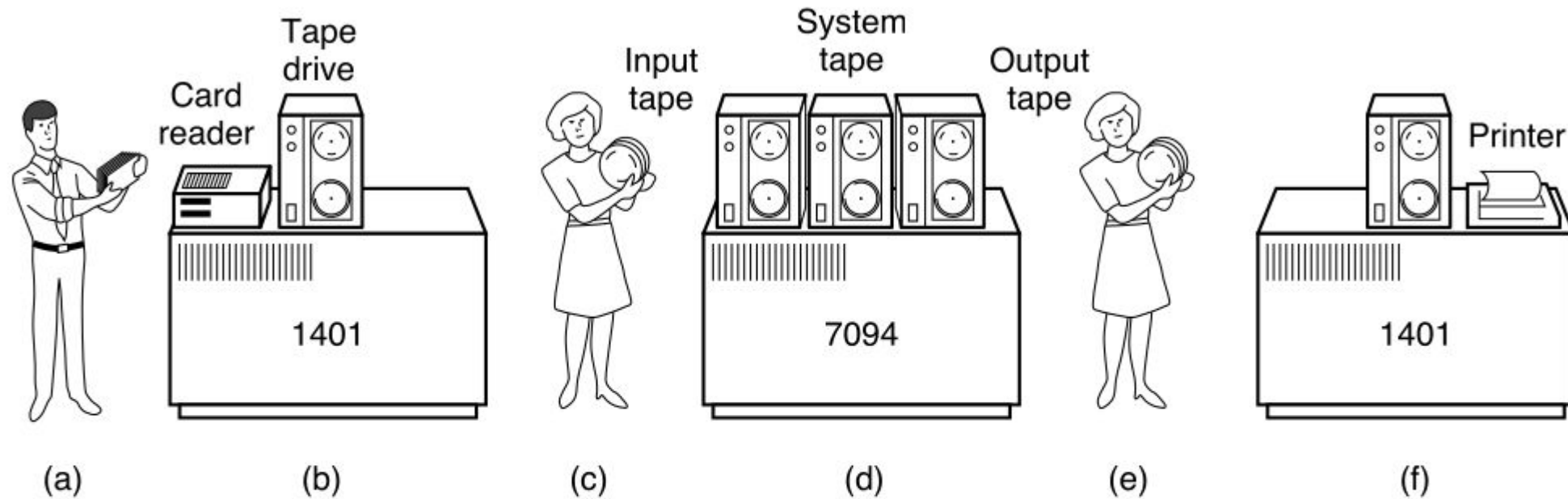
- Sistemas batch simples
- Sistemas batch multiprogramados





# Sistemas Operativos I

## Evolución de los Sistemas Operativos - Sistemas batch simples - Sistemas batch multiprogramados

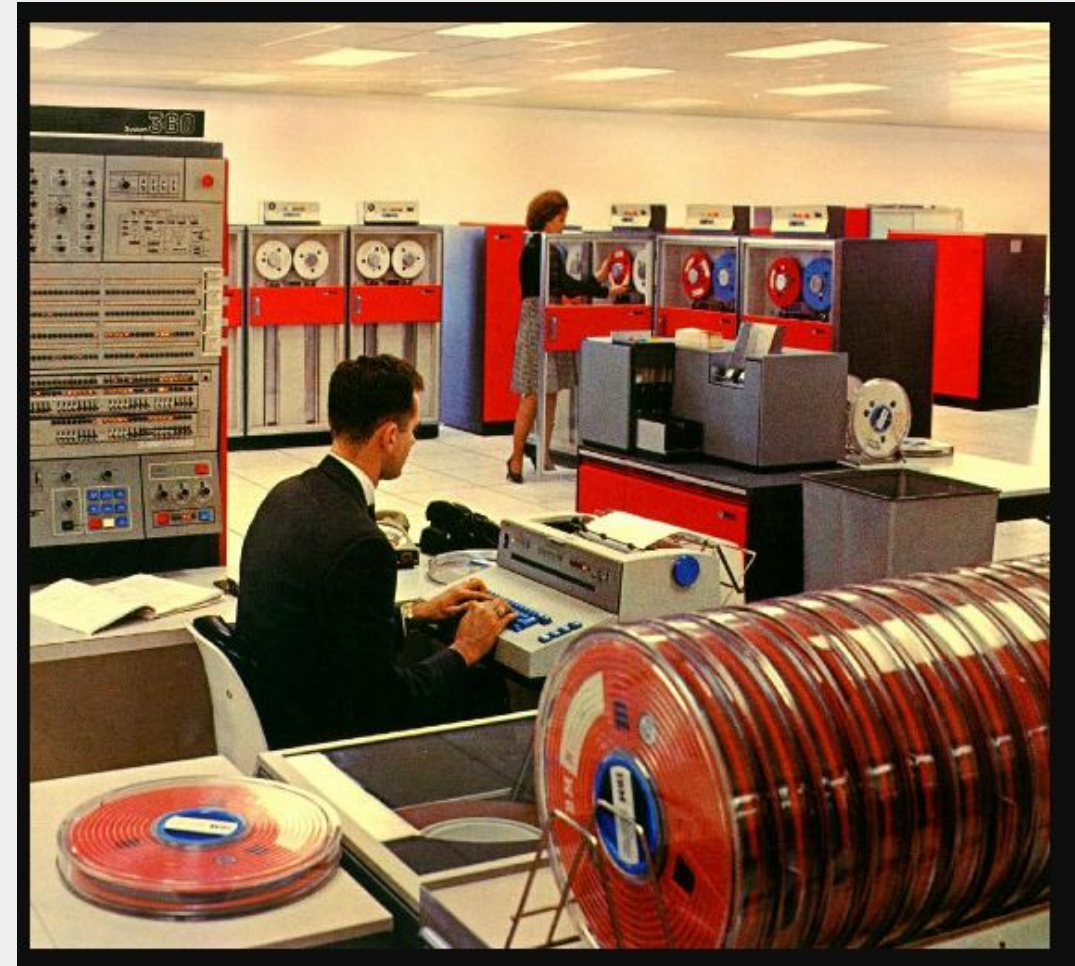
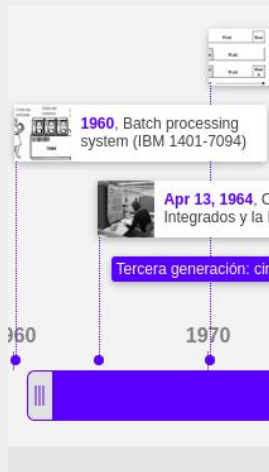


**Figure 1-3.** An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

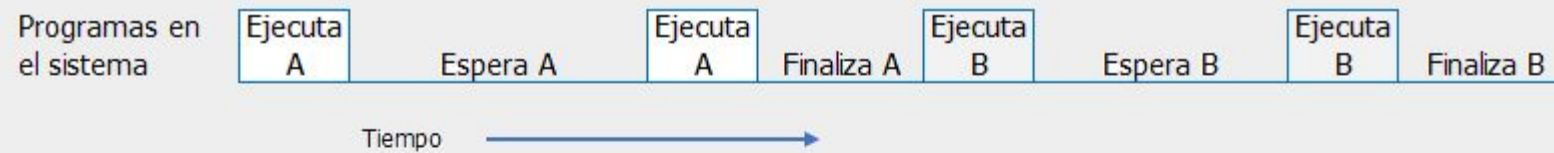
# Sistemas Operativos I

## Evolución de los Sistemas Operativos

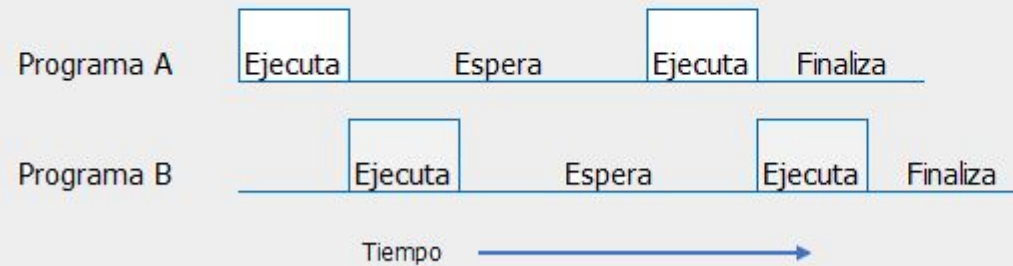
- Sistemas batch simples (2da generacion: transistores) 1955
- Sistemas batch multiprogramados (3ra generacion ICs) 1965



## Evolución de los Sistemas Operativos: Sistemas batch



a) Monoprogramación



b) Multiprogramación



# Sistemas Operativos I

## Evolución de los Sistemas Operativos: Era de la computadora personal (4ta. generación. LSI. 1980)

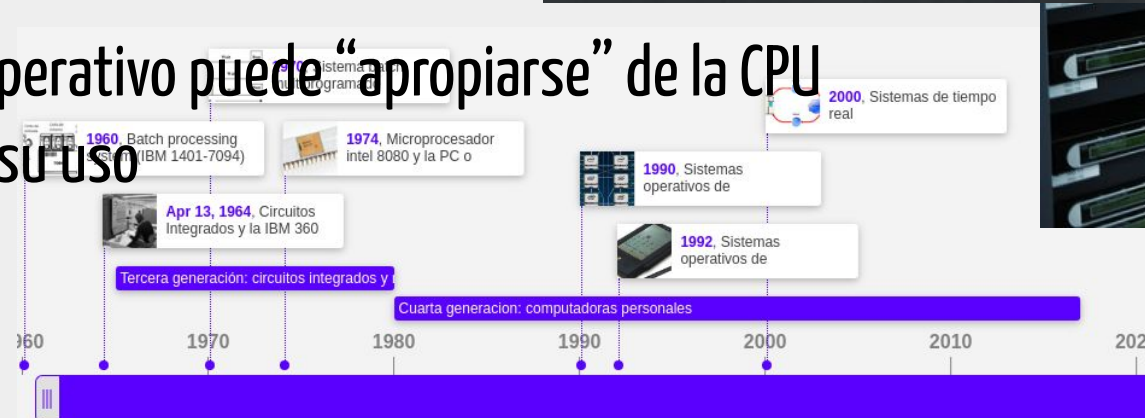
(large scale integration)

Tiempo compartido. Un proceso puede

- Estar en ejecución
- Esperar por una entrada salida
- Esperar por un evento



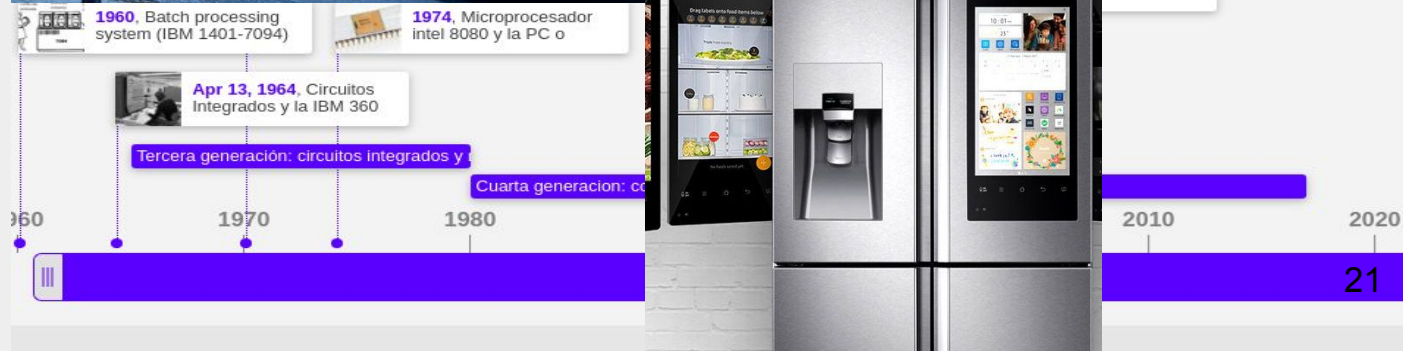
- El sistema operativo puede “apropriarse” de la CPU y conmutar su uso



# Sistemas Operativos I

## Evolución de los Sistemas Operativos: **Sistemas embebidos**

- Se encuentra el sistema operativo y una aplicación dedicada
- El software controla el hw y contiene una aplicación dedicada

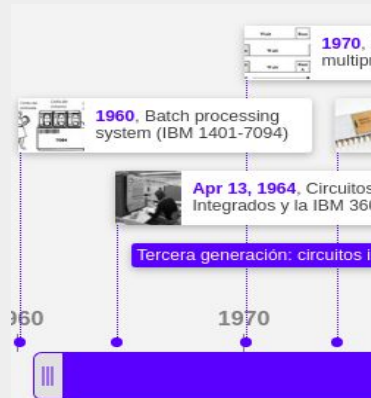




# Sistemas Operativos I

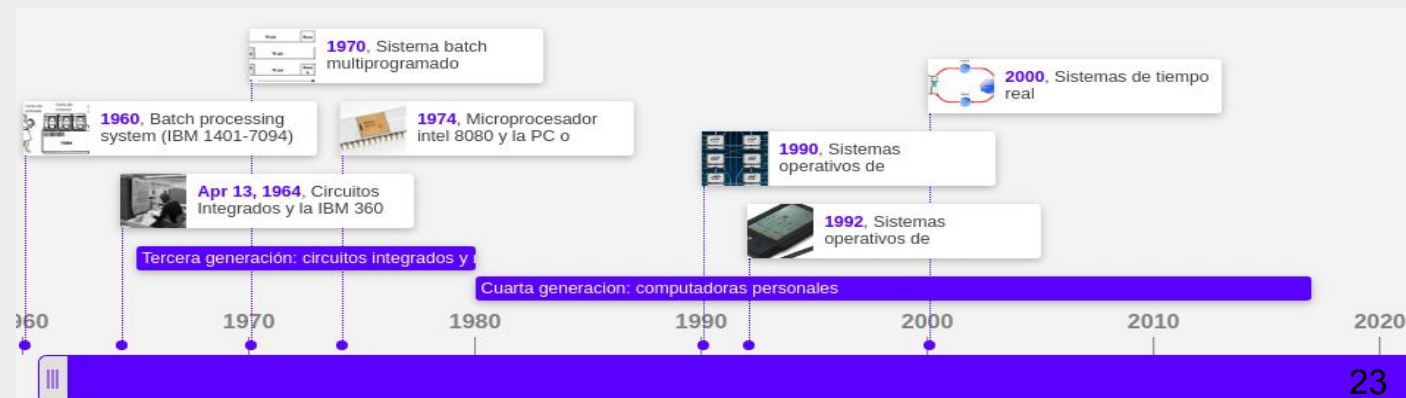
## Evolución de los Sistemas Operativos: **Sistemas de tiempo REAL**

- Se encuentra el sistema operativo y una aplicación dedicada
- La aplicación cuenta con tareas y un RTOS
- El sistema debe computar los resultados correctamente, y además, hacerlo en un tiempo menor a tiempos de respuestas límites

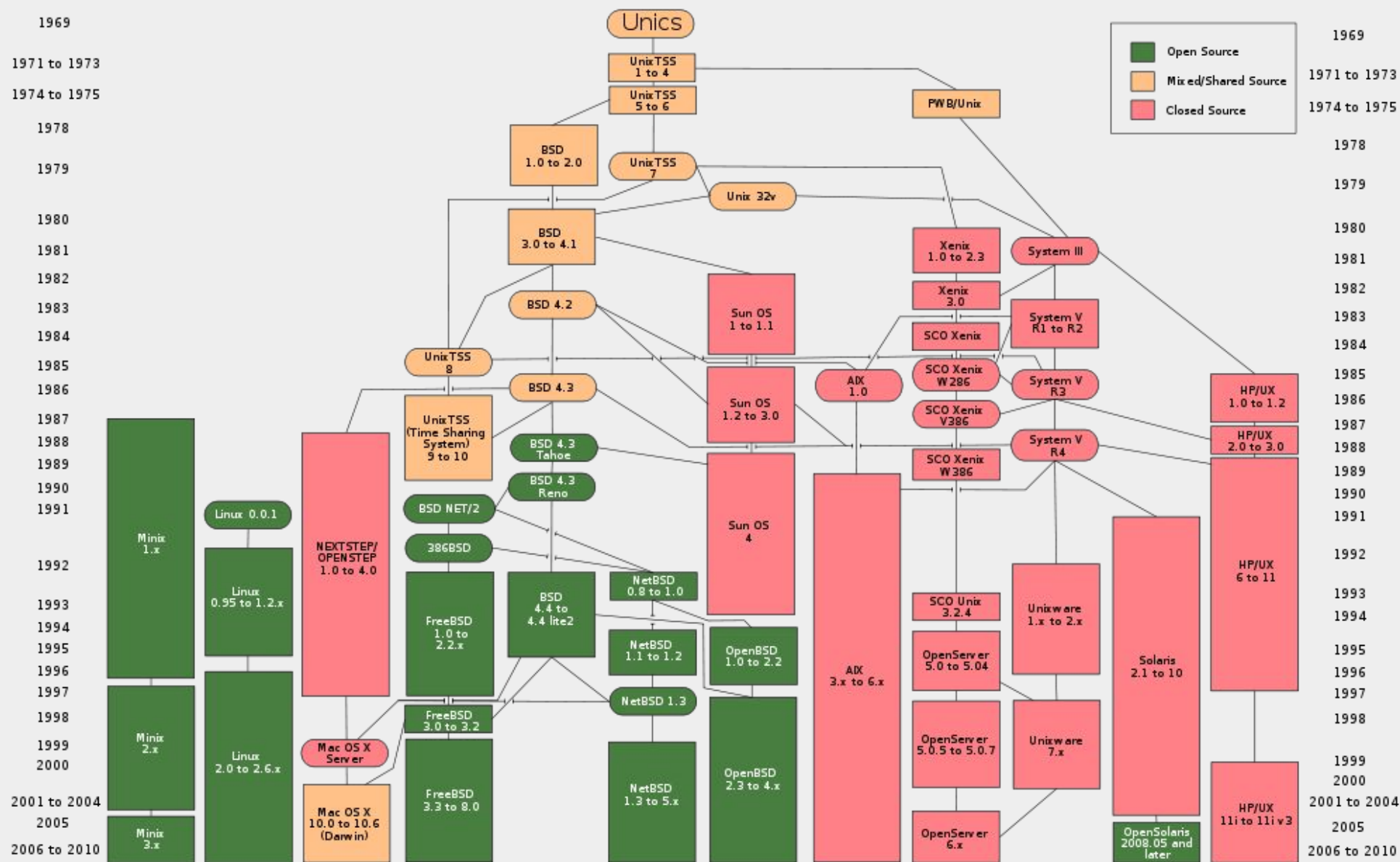


## Evolución de los Sistemas Operativos: **Sistemas distribuidos**

- Una aplicación, en donde, porciones del código, están distribuidos en diferentes sistemas de cómputo (posiblemente con diferentes sistemas operativos).
- ejemplo de herramientas: RPC y RMI



# Sistemas Operativos I





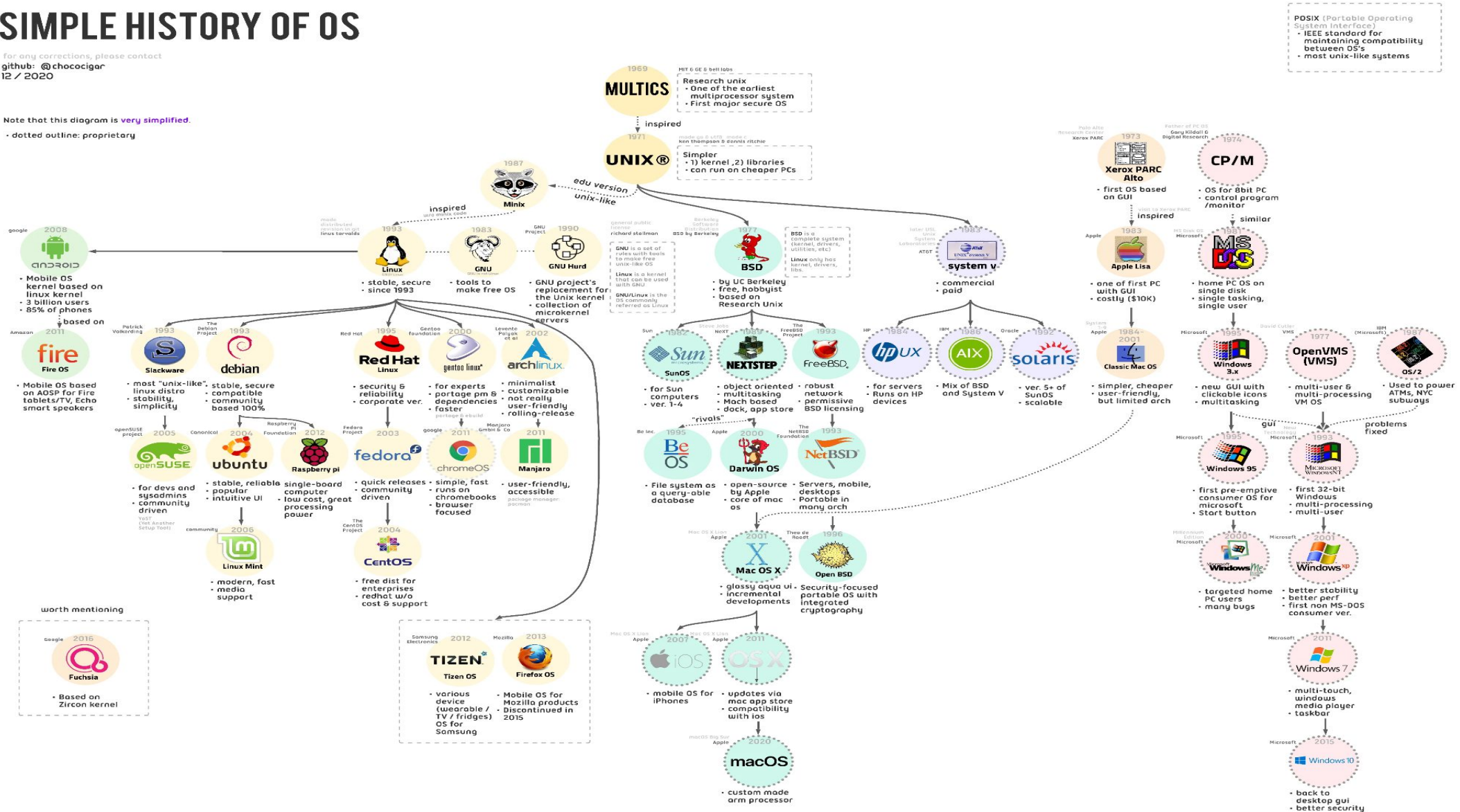
# Sistemas Operativos I

## SIMPLE HISTORY OF OS

for any corrections, please contact  
github: @chococigar  
12 / 2020

Note that this diagram is **very simplified**.

• dotted outline: proprietary

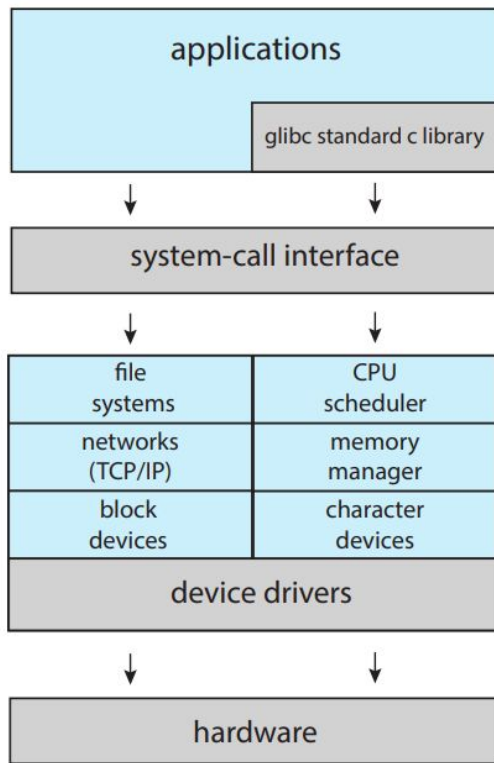


## Estructura de los Sistemas Operativos

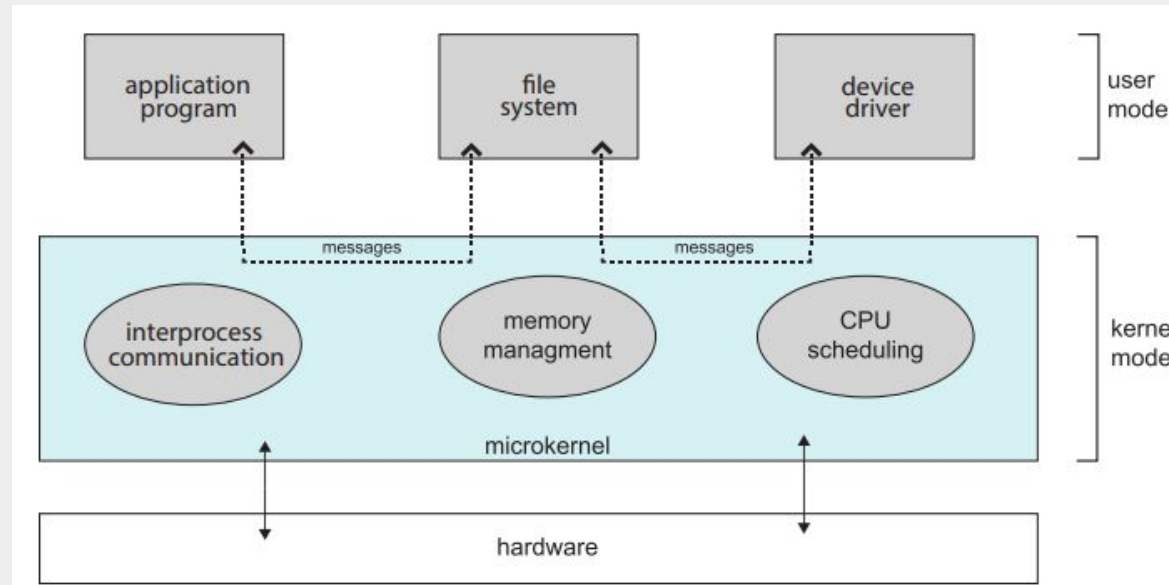
- **Monolítico**
- **Jerarquía de Múltiples Niveles**
- **Microkernel**
- **Máquina Virtual**

# Sistemas Operativos I

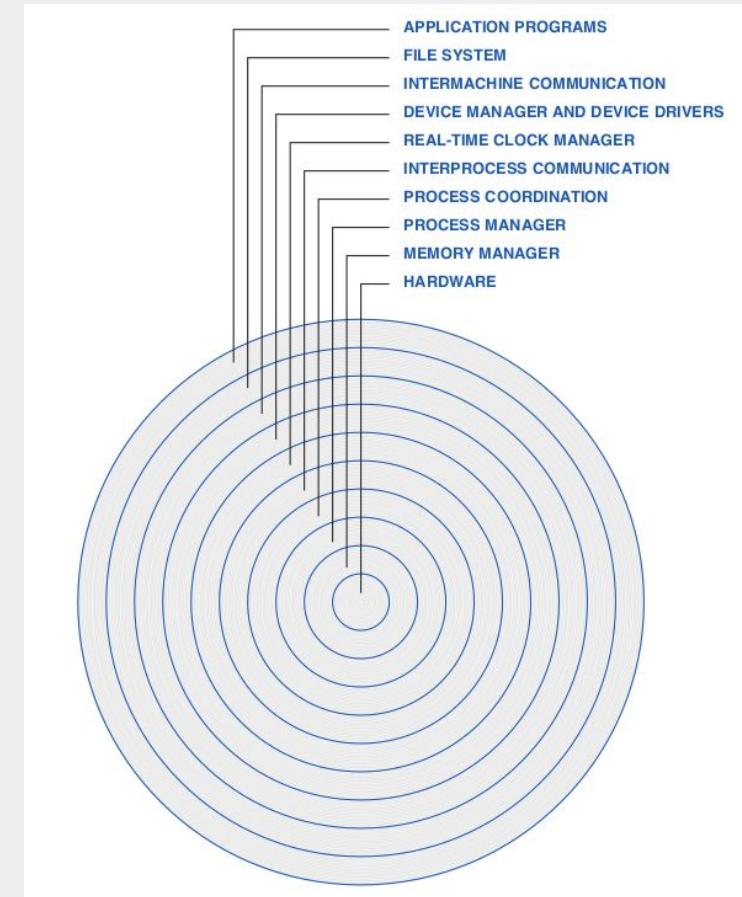
## Estructura de los Sistemas Operativos



monolítico



microkernel



múltiples niveles

# Sistemas Operativos I

---

## Preguntas de repaso

- ¿Cómo busco información de cómo utilizar la función `strlen()` de la biblioteca de C?
- ¿Qué es una llamada al sistema?
- ¿Que es modo supervisor de una CPU?
- ¿Puede una aplicación acceder al espacio de memoria del kernel?
- ¿Qué es un sistema operativo?
- ¿Cuáles son las categorías de servicios que provee un sistema operativo?
- ¿Cuáles son los componentes de un sistema operativo?