MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Data handling
Operating Systems

Alejandro Alonso alejandro.alonso@upm.es

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

Objetivos del tema

- Describir las características principales de los SO
- Describir los principios básicos de su operación
- Identificar los componentes principales de los SO
- Interacción con dispositivos hardware: memoria, E/S, discos

© 2021 Grupo STRAST - UPM

Contenidos

© 2021 Grupo STRAST - UPM

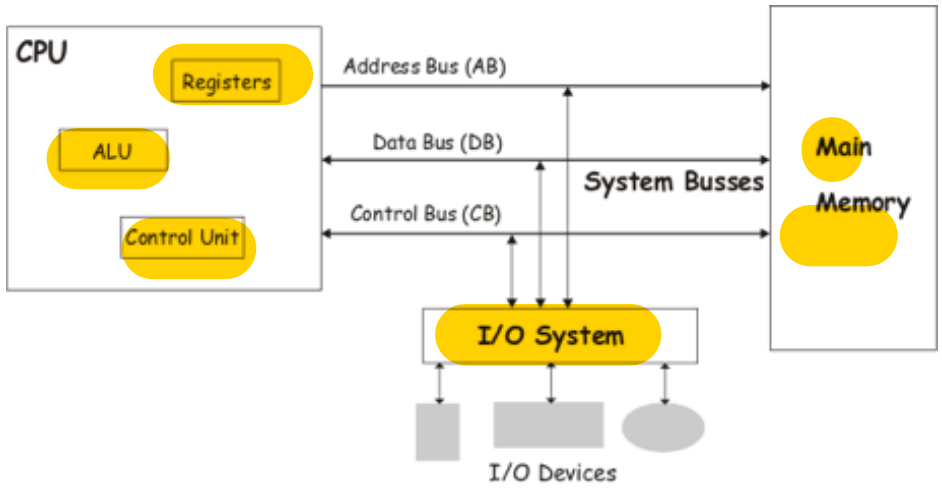
Planteamiento

ADCS | Experiments | Platform | TC/TM | OBDH

Program, task/thread

- How is executed a program or thread?
- How is assigned memory or UCP?
- How are used I/O devices?
- How do tasks communicate?

Operating System



Hardware

© 2021 Grupo STRAST - UPM

On-board computers: Specific requirements

- **embedded in spacecraft**
  - specific I/O devices, limited power, size and weight
- **real-time operation**
  - functions have to be performed on time
- **high reliability requirements**
  - repair is not possible in space
  - Spacecraft can fail to work or to be lost
- **harsh environment**
  - radiation, temperature, vibration, acceleration
- **Applicable technology may be restricted**
  - Often resulting in less powerful technology being used

© 2021 Grupo STRAST - UPM

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Introducción a los Sistemas Operativos

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

1.1 Introducción a los sistemas operativos

- **¿Qué es un sistema operativo?**
  - Difícil de definir. Una posible definición es:

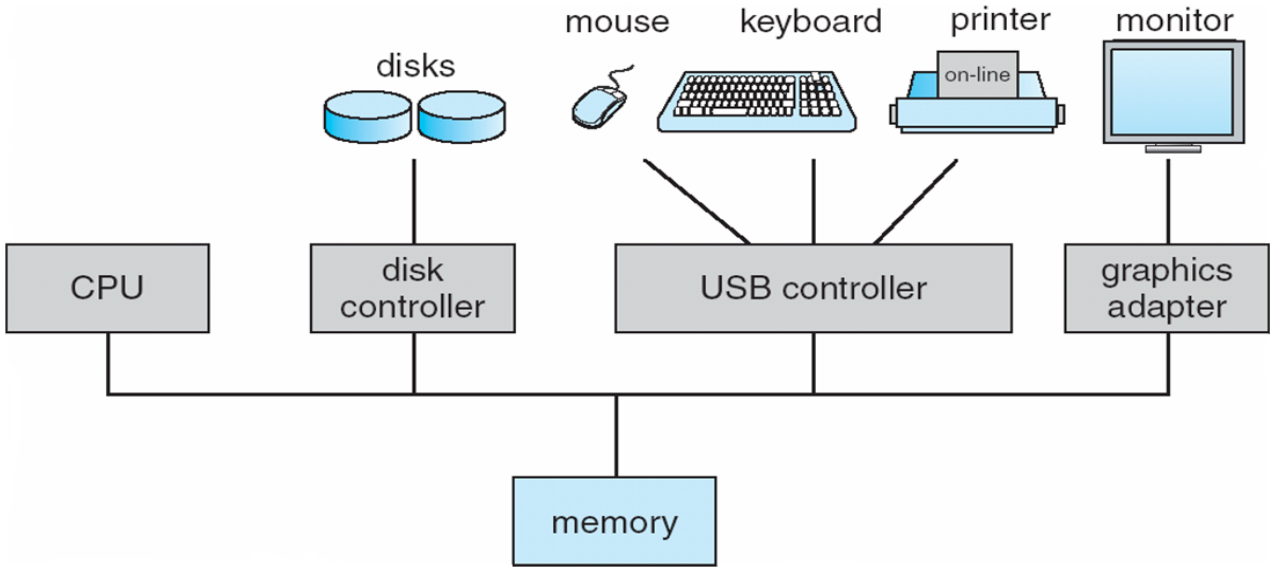
Programa que actúa como intermediario entre el hardware y los usuarios de un computador

- **Objetivos del sistema operativo**
  - Gestión de recursos:
    - Oculta complejidad del hardware
    - Eficiencia de uso de recursos y control de intentos de acceso simultáneos
  - Facilitar el uso del ordenador: abstracciones adecuadas
    - Mayor nivel de abstracción que el hardware
    - Abstracciones de alto nivel: procesos, ficheros

© 2021 Grupo STRAST - UPM

Gestión de los recursos de un Computador

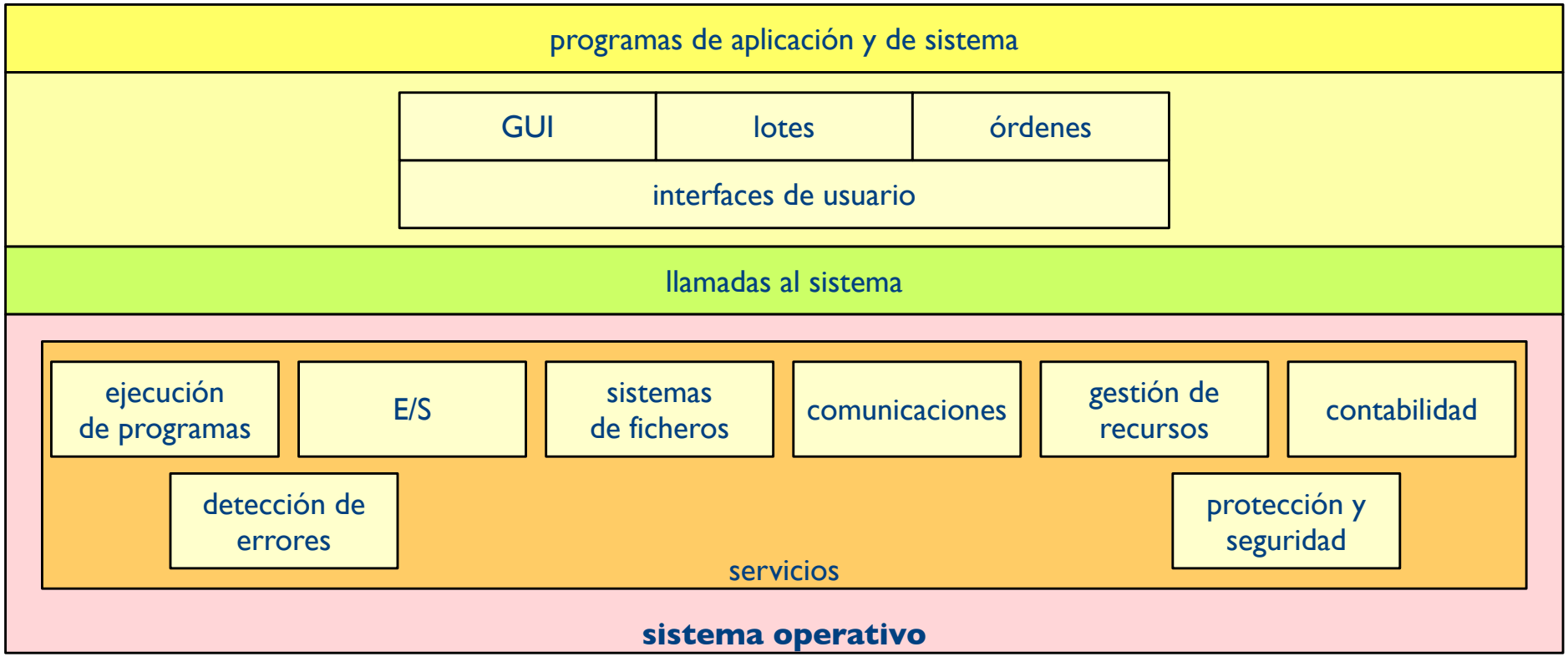
- **Controladores de dispositivos muy variados y complejos**
  - Interfaz de alto nivel: ocultar las peculiaridades del HW
  - Ficheros: Sucesión de registros lógicos. En Unix: bytes
- **Contención en el uso de recursos**
  - Varios usuarios tratan de usar simultáneamente recursos de HW, como procesador, memoria o dispositivos de E/S



© 2021 Grupo STRAST - UPM

Servicios del sistema operativo

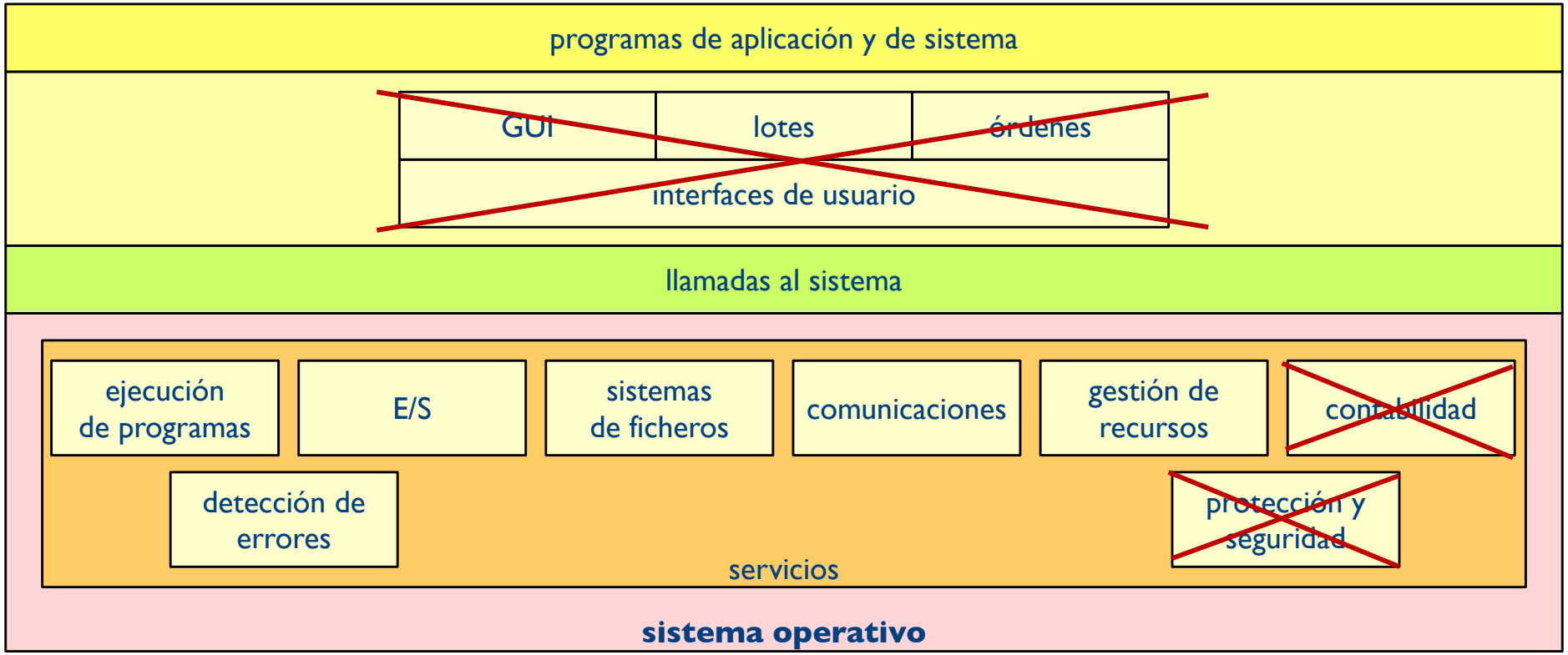
- **SO proporciona un entorno para ejecutar programas**
  - se realiza mediante un conjunto de llamadas al sistemas
- **Los programas o bibliotecas para usuarios**



© 2021 Grupo STRAST - UPM

Servicios del SO: Sistema empotrado

- Sistemas empotrados en aplicaciones espaciales:
  - Algunos tipos de servicios no se proporcionan
  - Otros servicios proporcionan una funcionalidad simplificada



© 2021 Grupo STRAST - UPM

Servicios básicos

- interfaz de usuario
  - gráfica o textual
- ejecución de programas
  - carga en memoria y arranque ejecución
- operaciones de E/S
  - lectura y escritura desde los programas
- gestión de ficheros
  - organización y acceso a la información
- comunicaciones
  - entre tareas (local o distribuida)
- detección de errores
  - errores de hardware o de software
  - acciones de corrección para preservar la

Sist. empotrados (espacio)

No necesario

Sólo al iniciar el sistema

Dispositivos dedicados a la misión

O no se necesita o un gestor sencillo

Mecanismos básicos

Necesarios

© 2021 Grupo STRAST - UPM

Servicios básicos (continuación)

Sist. empotrados (espacio)

- gestión de recursos
  - CPU, memoria, espacio en disco, etc.
- contabilidad
  - datos sobre uso de recursos
- protección y seguridad
  - los procesos no deben interferir entre ellos
  - los usuarios no deben poder acceder a los recursos sin autorización

Necesario

No necesario

Las tareas colaboran.

No hay usuarios.

© 2021 Grupo STRAST - UPM

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Funcionamiento de un Sistema Operativo

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

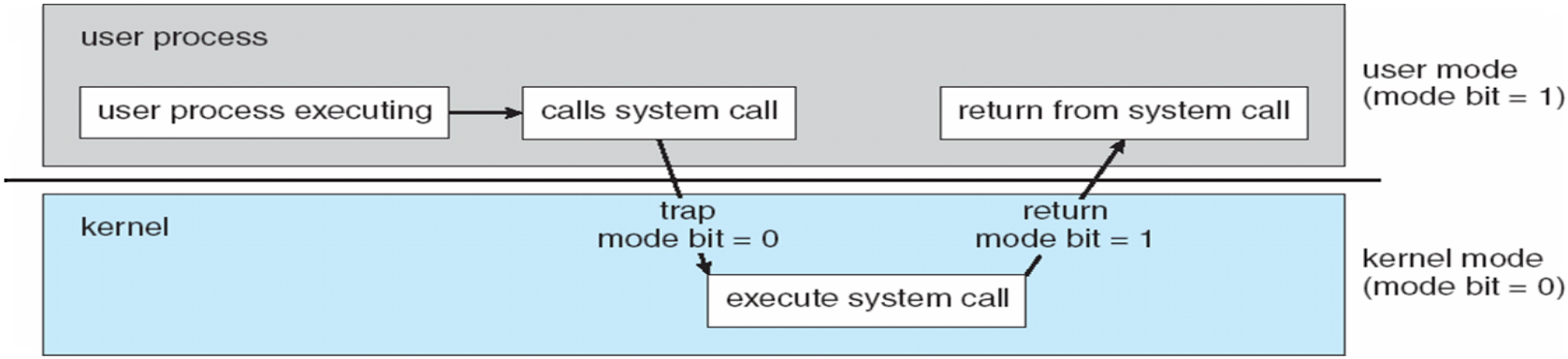
1.2. Funcionamiento del Sistema Operativo

- **Dirigido por interrupciones:**
  - Hardware: al interactuar con dispositivos
  - Software
    - servicios al usuario (llamadas al sistema)
    - errores de ejecución: División por cero, instrucciones ilegales
- **SO se basa en el modelo dual del procesador:**
  - Dos modos: usuario y núcleo o privilegiado
    - El hardware proporciona un bit de modo
  - Algunas instrucciones se tienen que ejecutar en modo privilegiado
  - El núcleo se ejecuta en modo privilegiado
  - Fundamental para proteger al sistema

© 2021 Grupo STRAST - UPM

Tratamiento de interrupciones

- **Cuando se produce una interrupción:**
  - Se salva el contexto de la ejecución en curso
  - Se pasa a modo supervisor / modo núcleo
  - Se inhabilitan las interrupciones para que no se pierdan
  - Transfiere control a la rutina de tratamiento correspondiente
    - Se suele emplear vectores de interrupciones



© 2021 Grupo STRAST - UPM

Sistemas empotrados

- **Los sistemas operativos, dirigidos por interrupciones**

- **No es imprescindible usar los dos modos de funcionamiento:**
  - No hay que proteger de otros usuarios, ni de accesos del exterior
  - Las llamadas al sistema se realizan como llamada a procedimiento

© 2021 Grupo STRAST - UPM

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Interfaz de programación

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

Interfaz de programación: llamadas al sistema

- Los programas pueden acceder a los servicios del sistema invocando llamadas al sistema
- Especificación de operaciones del sistema en términos de un lenguaje de programación
  - funciones, subprogramas o métodos con parámetros
  - sintaxis de un lenguaje de programación (ej. Lenguaje C)
- **Ejemplo (Java): java.io.OutputStream.write:**

```
public abstract void write (byte[] b,int off,int len)
  throws IOException
```

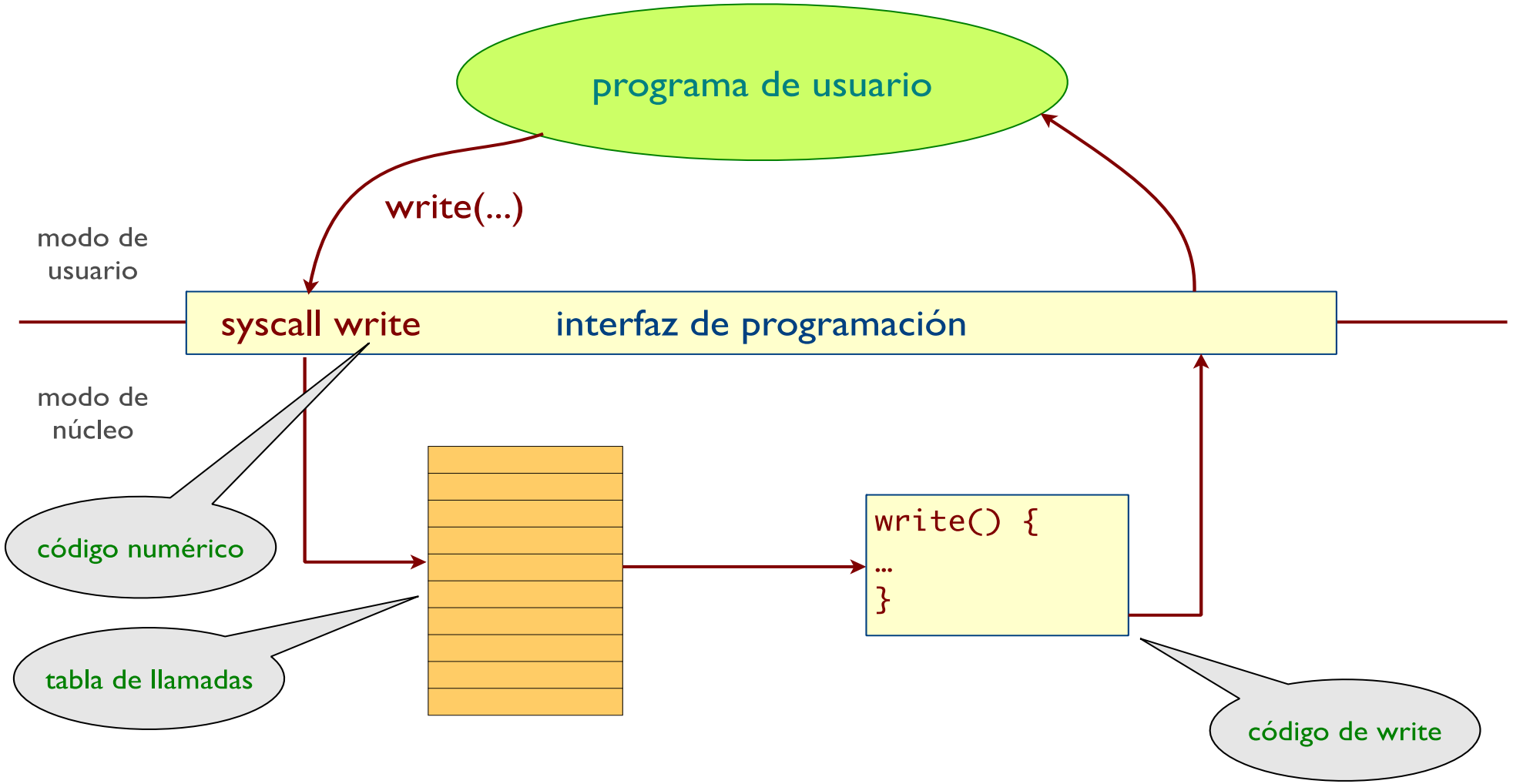
- **Ejemplo (C):**

```
ssize_t write(int fd, const void *buf, size_t count)
```

- **Algunas APIs comunes: Win32, POSIX, Java API**

© 2021 Grupo STRAST - UPM

Ejecución de llamadas al sistema



© 2021 Grupo STRAST - UPM

Tipos de llamadas al sistema

	Windows	Unix

Gestión de procesos	`CreateProces()`
	`ExitProcess()`
	`WaitForSingleObject()`
		`fork()`
		`exit()`
		`wait()`

Gestión de ficheros	`CreateFile()`
	`ReadFile()`
	`WriteFile()`
	`CloseHandle()`
		`open()`
		`read()`
		`write()`
		`close()`

Gestión de dispositivos	`SetConsoleMode()`
	`ReadConsole()`
	`WriteConsole()`
		`ioctl()`
		`read()`
		`write()`

Mantenimiento de información	`GetCurrentProcessId()`
	`SetTimer()`
	`Sleep()`
		`getpid()`
		`alarm()`
		`sleep()`

Comunicaciones	`CreatePipe()`
	`CreateFileMapping()`
	`MapViewOfFile()`
		`pipe()`
		`shmget()`
		`mmap()`

Protección	`SetFileSecurity()`
	`InitializeSecurirty`
	`Descriptor()`
	`SetSecurityDescriptorGroup()`
		`chmod()`
		`umask()`
		`chown()`

© 2021 Grupo STRAST - UPM

POSIX

- **POSIX (Portable Operating System Interface) es un estándar que define una interfaz de programación para sistemas operativos**
  - IEEE Std 1003.1-2008 — ISO /IEC 9945:2009
  - API para C y otros lenguajes basada en Unix
  - define servicios de SO y la API para las llamadas al sistema correspondientes
  - los servicios son opcionales
    - un SO no tiene por qué tener todos los servicios definidos en POSIX
    - pero si los tiene debe seguir la API de POSIX
- **Algunos SO compatibles con POSIX**

© 2021 Grupo STRAST - UPM

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Estructura de los Sistemas Operativos

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

Estructura de sistemas operativos

- **Generalmente se divide en componentes organizados jerárquicamente**
  - de mayor a menor nivel de abstracción
- **Ejemplo: estructura original de Unix**

© 2021 Grupo STRAST - UPM

The Open Ravenscar Kernel

- **ORK is a real-time kernel for ERC-32 computers**
  - open source alternative to commercial kernels and OS
  - small kernel supporting only Ravenscar profile functionality
- **ORK is integrated with GNAT**
- **The full GNAT/ORK compilation system includes other tools:**
  - Libraries, debugger, remote monitor and loader for embedded targets

© 2021 Grupo STRAST - UPM

GNAT

- **GNAT (GNU/New York University Ada Translator) is a free compilation system for Ada 95**
  - initially developed under DoD funding
  - later taken over by ACT

- **Characteristics**
  - source code available
  - fully integrated with GNU software development toolkit
  - supports full core Ada plus annexes
    - Annex C: System programming
    - Annex D: Real Time systems

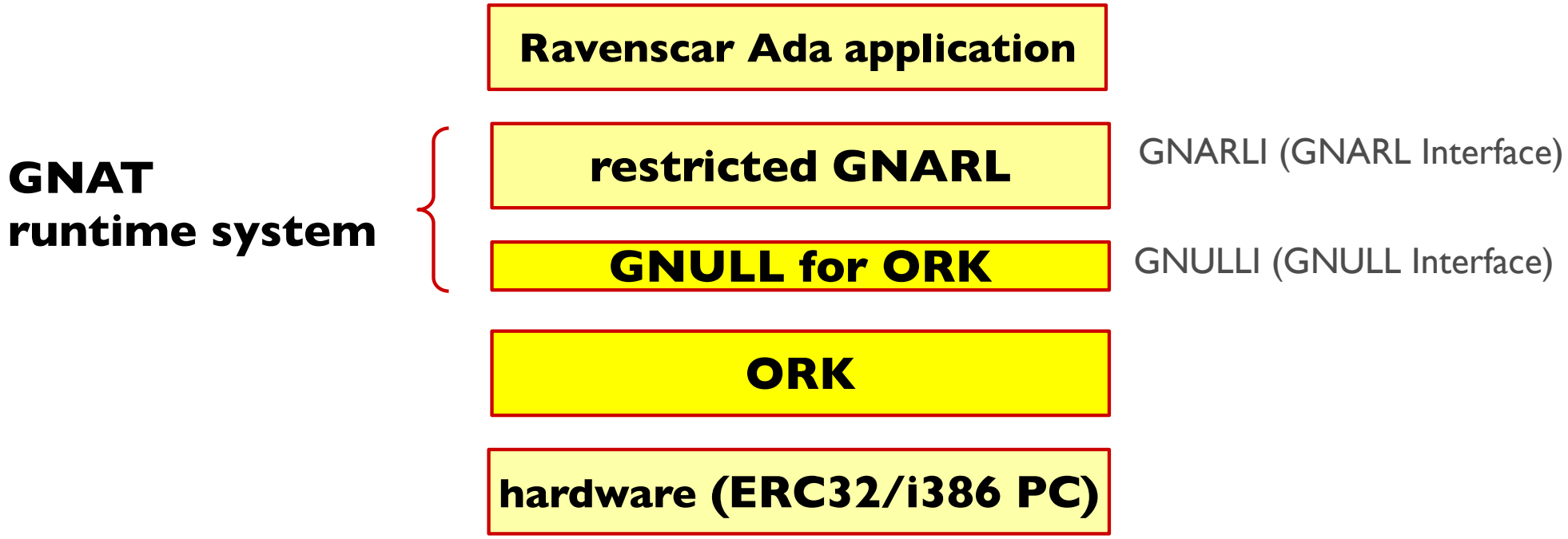
© 2021 Grupo STRAST - UPM

Tasking implementation in GNAT

- **Ada tasking is implemented by means of GNARL (GNU Ada Runtime Library)**
  - a platform-independent collection of Ada packages
- **Low level dependencies are isolated by means of GNULL (GNU Low-level Library)**
  - different versions of GNULL for different platforms
  - tasking at this level is usually based on an underlying POSIX threads layer
- **Highly modular, portable approach**
  - but efficiency may be improved for particular targets
  - "abstraction inversion" – Ada tasks provide the same basic functionality as POSIX threads

© 2021 Grupo STRAST - UPM

GNAT run-time architecture



© 2021 Grupo STRAST - UPM

Configuración y Arranque del sistema

- **Se pate de un disco o partición de arranque, con:**
  - El código del SO, Ficheros de configuración del SO
  - El sistema de fichero de raíz del SO,
    incluyendo programas y ficheros del sistema y usuarios
- **Carga del núcleo (boot)**
  - cargador inicial, arranca por hardware: bootstrap loader
    - en ROM/EPROM (): comprobación de dispositivos básicos
    - Cargador general en bloque de disco predeterminado (GRUB)
  - a continuación se carga, el cargador completo
- **En sistemas empotrados:**
  - Se enlaza el SO y programas, se guarda en ROM
  - Al iniciar, se pasa cargar para ejecutar

© 2021 Grupo STRAST - UPM

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Gestores en un sistema de operativo

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

2. Gestores en un sistema de operativo

- El SO se puede dividir desde un punto de vista funcional:
  - Gestor de Procesos
  - Gestor de Memoria
  - Gestor de Almacenamiento Secundario
    - Sistemas de ficheros
  - Gestor de Entrada Salida
  - Protección y Seguridad

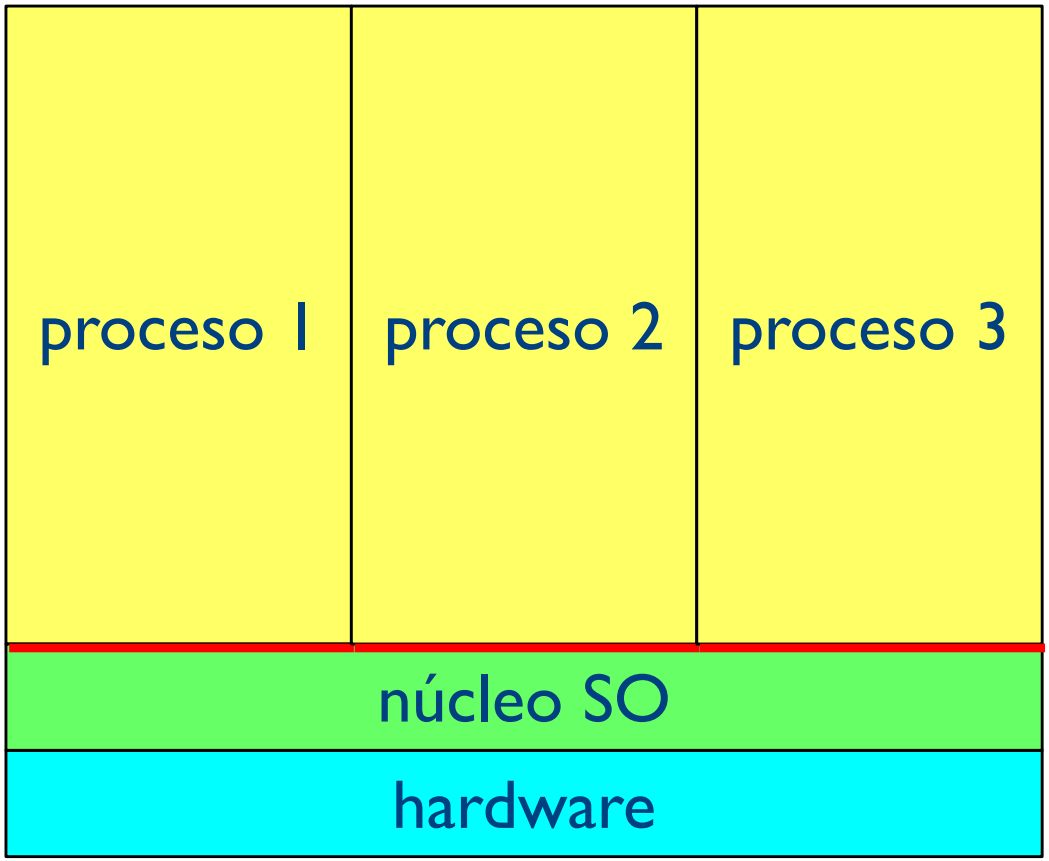
© 2021 Grupo STRAST - UPM

2.1 Gestión de Procesos

- **La mayoría de los sistemas operativos permiten ejecutar varios programas a la vez**
  - Ejemplos: procesadores de textos, navegadores, etc.
  - Programas de sistema: colas de impresión, comunicación…
- **Un proceso es un programa en ejecución**
  - con toda la información referente a su estado
- **El SO multiplexa la ejecución de los procesos**
  - la CPU ejecuta alternativamente las instrucciones de los procesos
- **Asigna un espacio de memoria propio al proceso**
  - Abstracción/Virtual: cada proceso se ejecuta como si tuviera una CPU y una memoria para él solo

© 2021 Grupo STRAST - UPM

Estructura de un SO: Gestión de Procesos



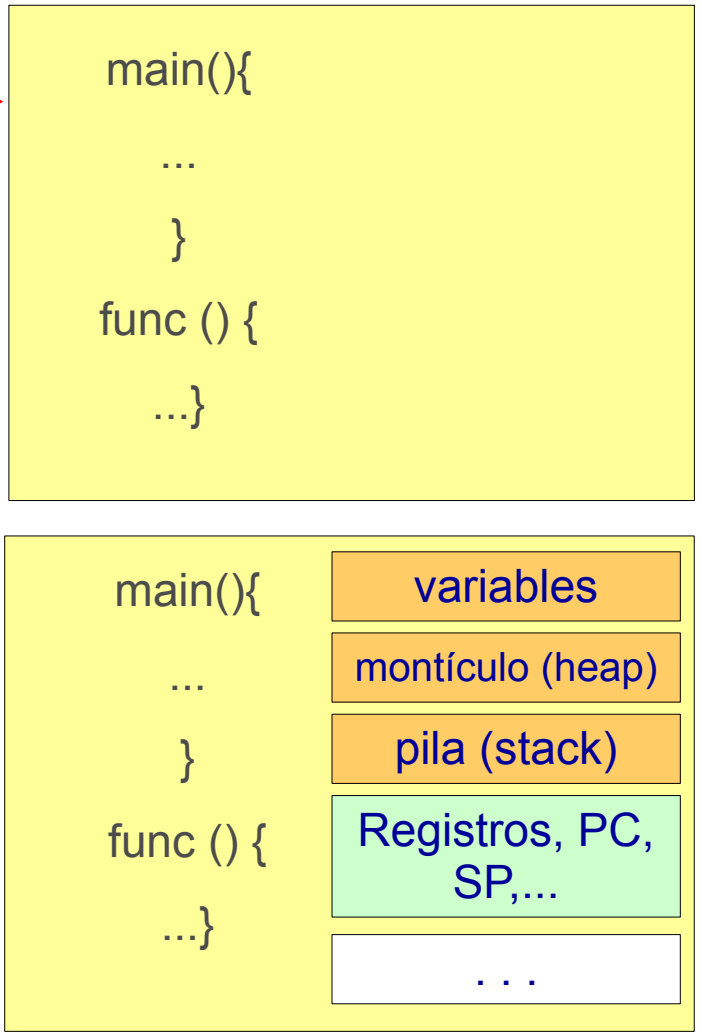
Esquema de ejecución

Mapa de memoria

© 2021 Grupo STRAST - UPM

Gestión de Procesos

- **Proceso: programa en ejecución**
- **Programa: código y datos estáticos**
- **Imagen del proceso en SO:**
  - Código
  - Estado del proceso:
    - Área de memoria: valores de variables, pila, etc.
    - Registros: contador de programa, puntero de pila, ...
    - . . .



© 2021 Grupo STRAST - UPM

Gestión de Procesos:

- **Servicios básicos:**
  - Creación, destrucción de procesos
- **Mecanismos de interacción entre procesos**
  - Sincronización y comunicación
- **Gestión de recursos:**
  - Procesador: qué proceso debe ejecutarse.
  - Memoria: asignación a procesos
  - Gestión de dispositivos de entrada/salida.
- **Los procesos están protegidos:**
  - Un proceso no puede acceder a la memoria de otro directamente

© 2021 Grupo STRAST - UPM

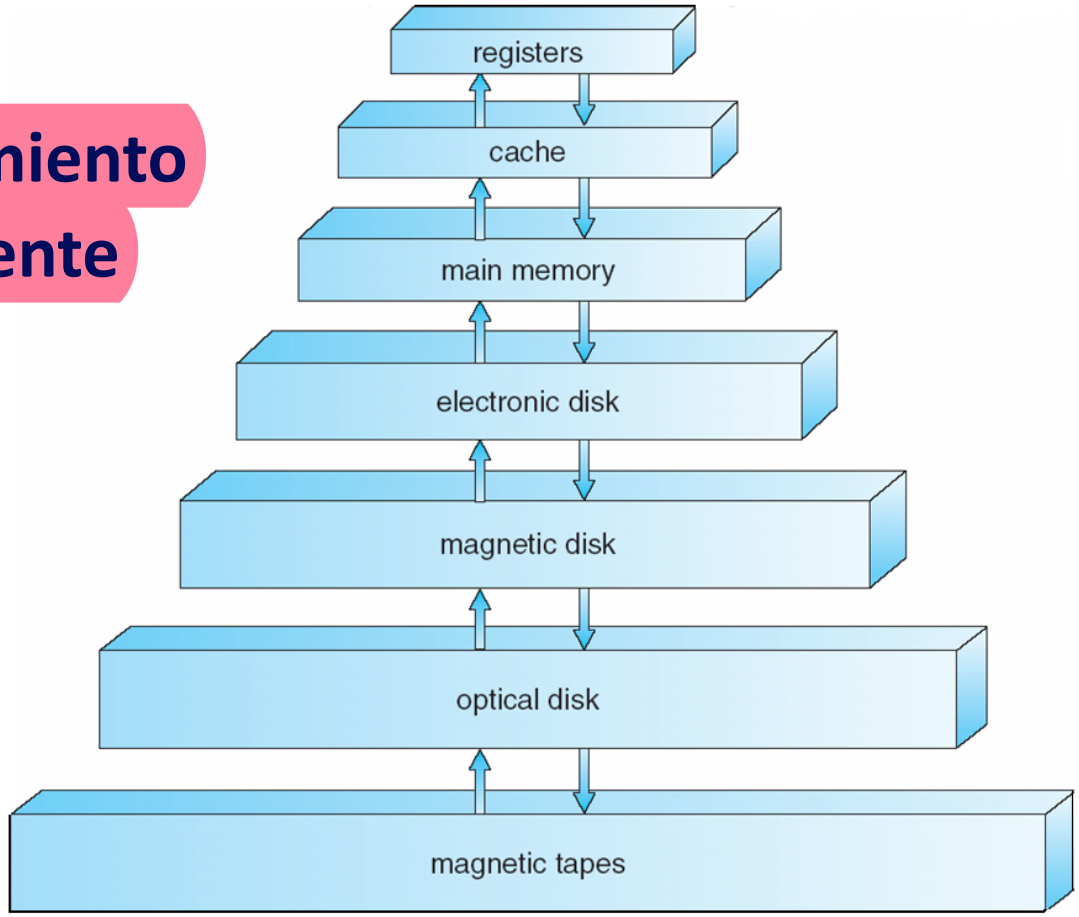
2.2 Gestión de Memoria

- **Las instrucciones y los datos de un proceso tienen que estar en memoria para ejecutarse**
- **El gestor decide cuáles son los contenidos de la memoria:**
  - Mapa de memoria, qué (partes) procesos, ...
  - Se trata de optimizar el uso de la CPU y la respuesta a los usuarios
- **Actividades del gestor de memoria:**
  - Memoria libre ocupada y quién la usa
  - Qué procesos (o partes) se mantienen en memoria

© 2021 Grupo STRAST - UPM

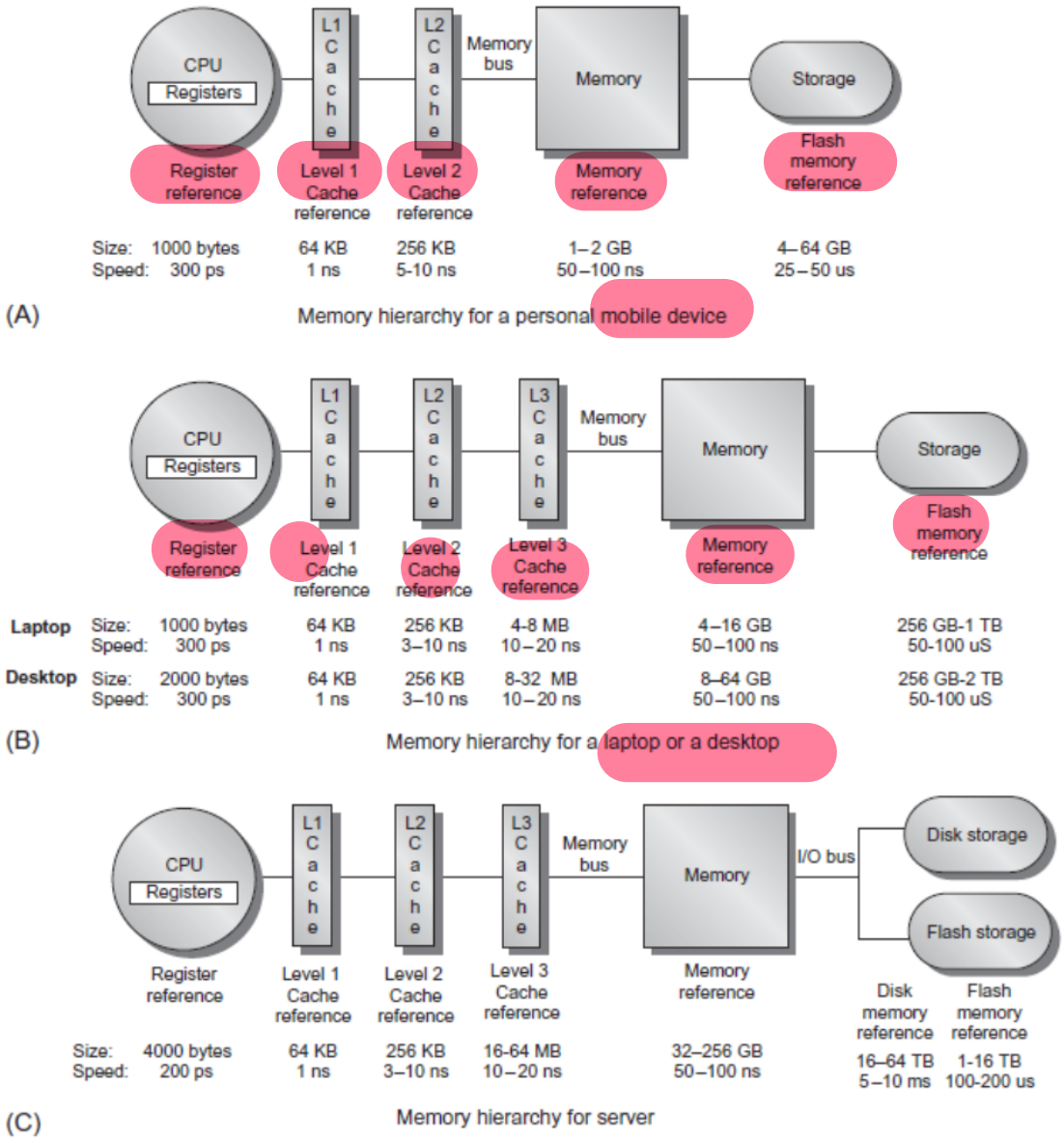
Jerarquía del Almacenamiento

- **El sistema de almacenamiento se organiza jerárquicamente**
  - Velocidad
  - Coste
  - Volatilidad



© 2021 Grupo STRAST - UPM

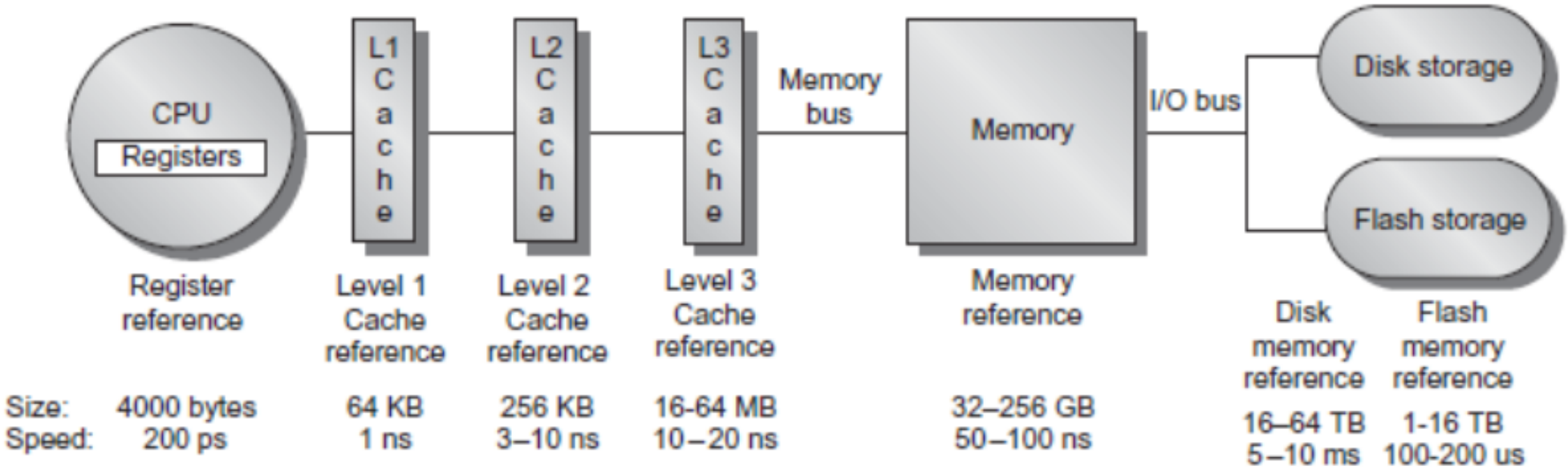
Jerarquía de memorias



Computer Architecture: A Quantitative Approach, J.L. Hannessy, D.A. Patterson,

© 2021 Grupo STRAST - UPM

Jerarquía de memorias

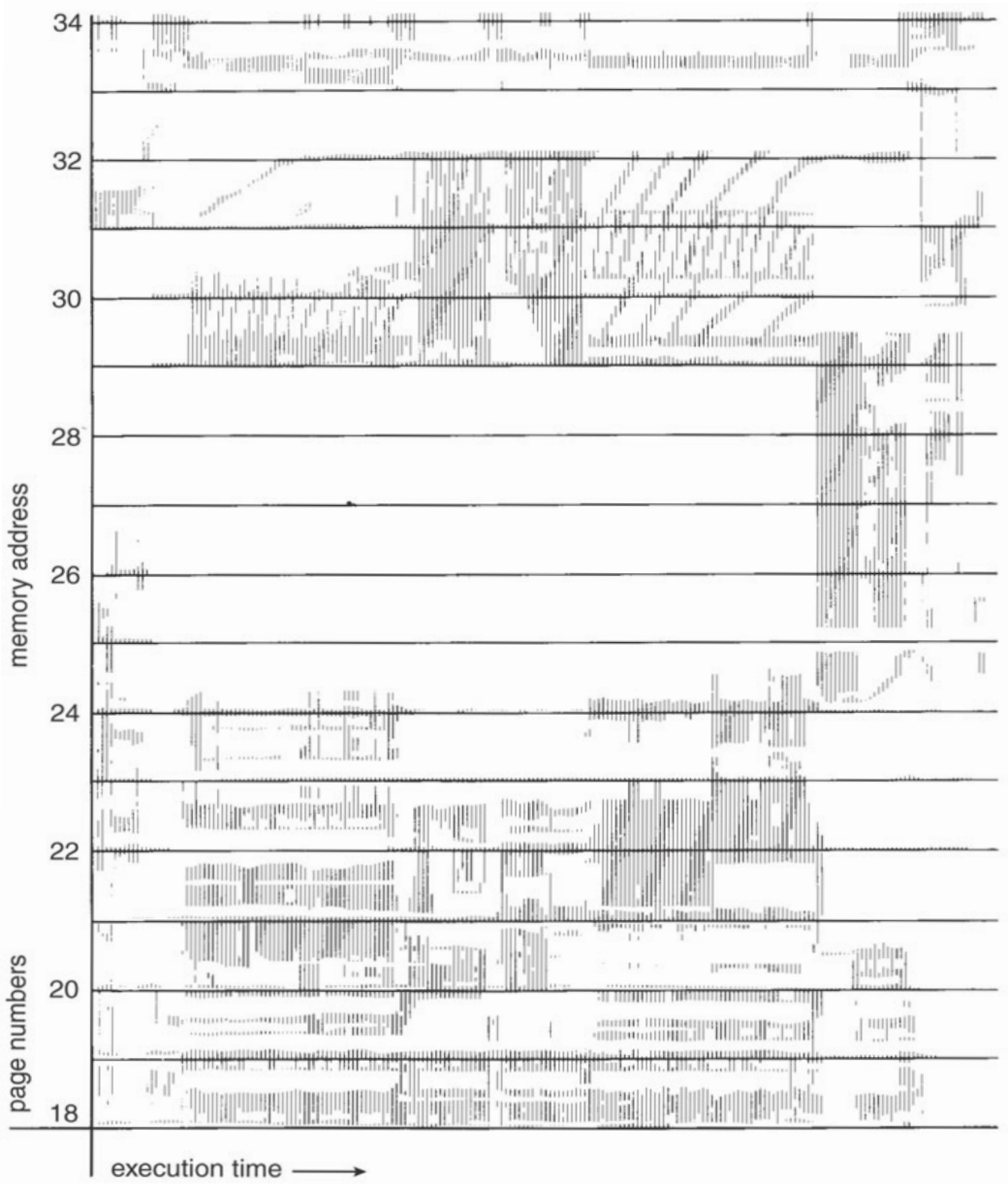


Memory hierarchy for server

Compilador Hardware Sistema operativo

© 2021 Grupo STRAST - UPM

Vecindad en la referencias a memoria



Silberschatz et al. 2010

© 2021 Grupo STRAST - UPM

2.3 Gestión de Ficheros

- **Proporciona una visión uniforme y virtual del almacenamiento de información**
  - Fichero: abstracción de almacenamiento de información que oculta los detalles de los dispositivos físicos
  - Cada dispositivo tiene características diferentes
- **Sistema de ficheros**
  - Los ficheros se suelen organizar en directorios
  - Control de acceso a la información
  - Operaciones del SO:
    - Creación y borrado de ficheros y directorios
    - Operaciones para manipular ficheros y directorios
    - Almacenar los ficheros en los dispositivos
    - Copias de seguridad en dispositivos no volátiles

© 2021 Grupo STRAST - UPM

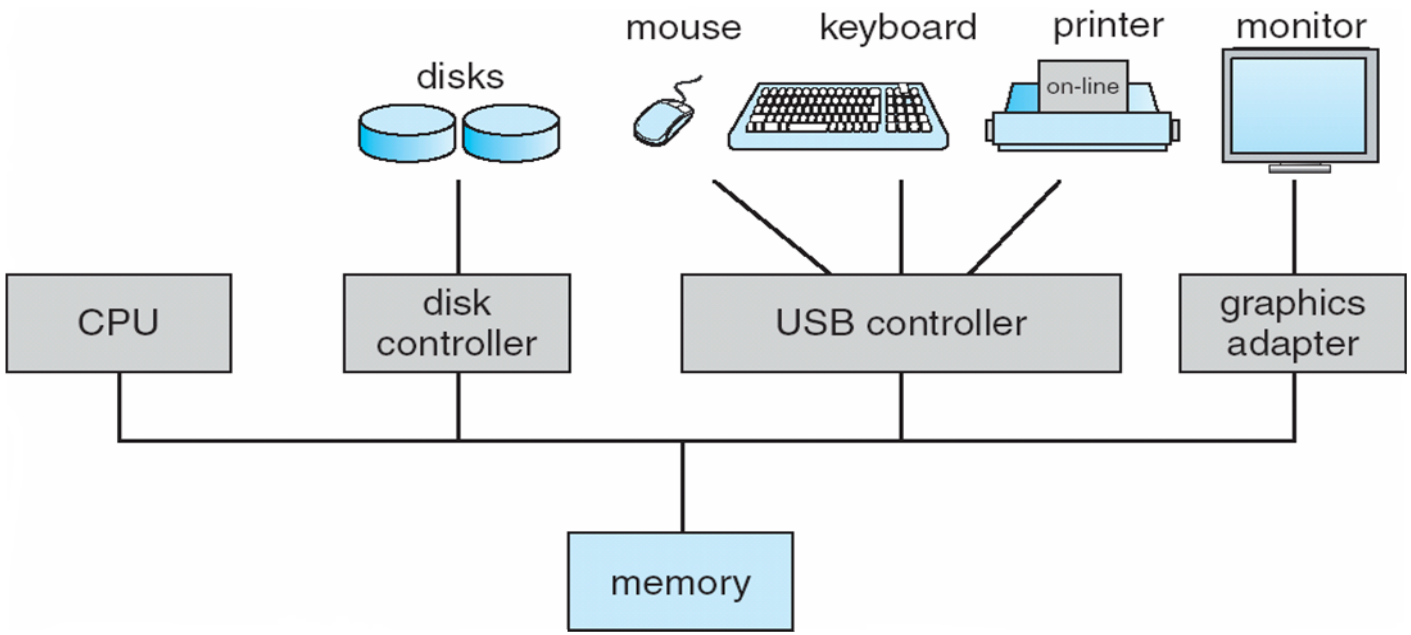
2.4 Gestión de E/S

- **Un objetivo del SO es ocultar los detalles y peculiaridades de los dispositivos de E/S**

- **Responsabilidades del SO:**
  - Gestión de la memoria de E/S:
  - Almacenamiento temporal de datos, mientras se transfieren
  - Caché de disco
  - Spooling, etc
- **Interfaz con los gestores de dispositivos**
- **Gestores para los diferentes dispositivos hardware**

© 2021 Grupo STRAST - UPM

Computador de Von Neumann



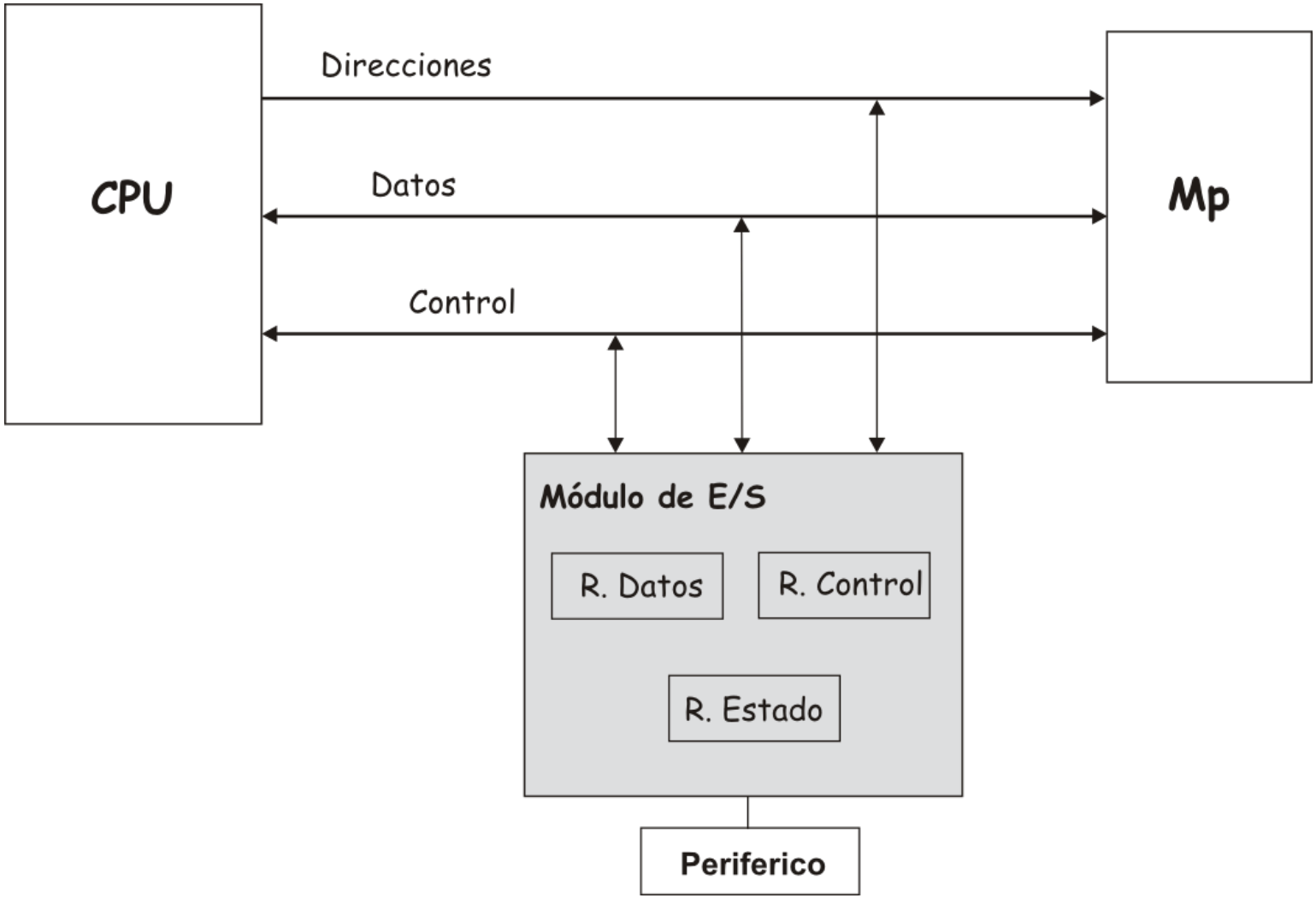
© 2021 Grupo STRAST - UPM

Técnicas de entrada/salida

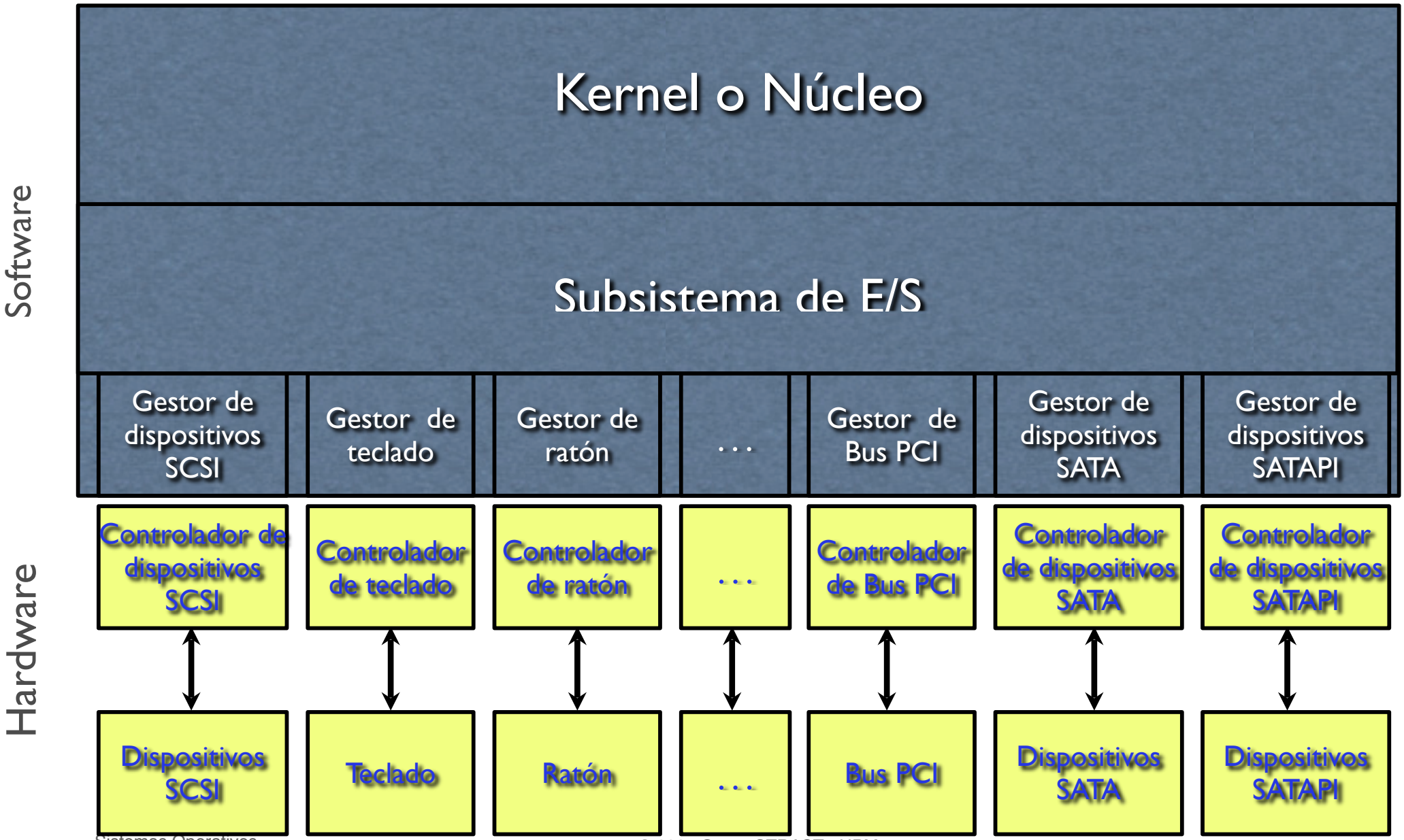
- **Los periféricos son más lentos que la CPU y memoria.**
- **E/S y periférico son independientes y trabajan en paralelo.**
- **Técnicas de E/S:**
  - E/S programada
  - E/S por interrupciones
  - E/S por DMA
- **El grado de participación de la CPU varía**

© 2021 Grupo STRAST - UPM

Sistema de entrada/salida



© 2021 Grupo STRAST - UPM



© 2021 Grupo STRAST - UPM

MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

Introducción a las máquinas virtuales

Some rights reserved. This work is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

STRAST

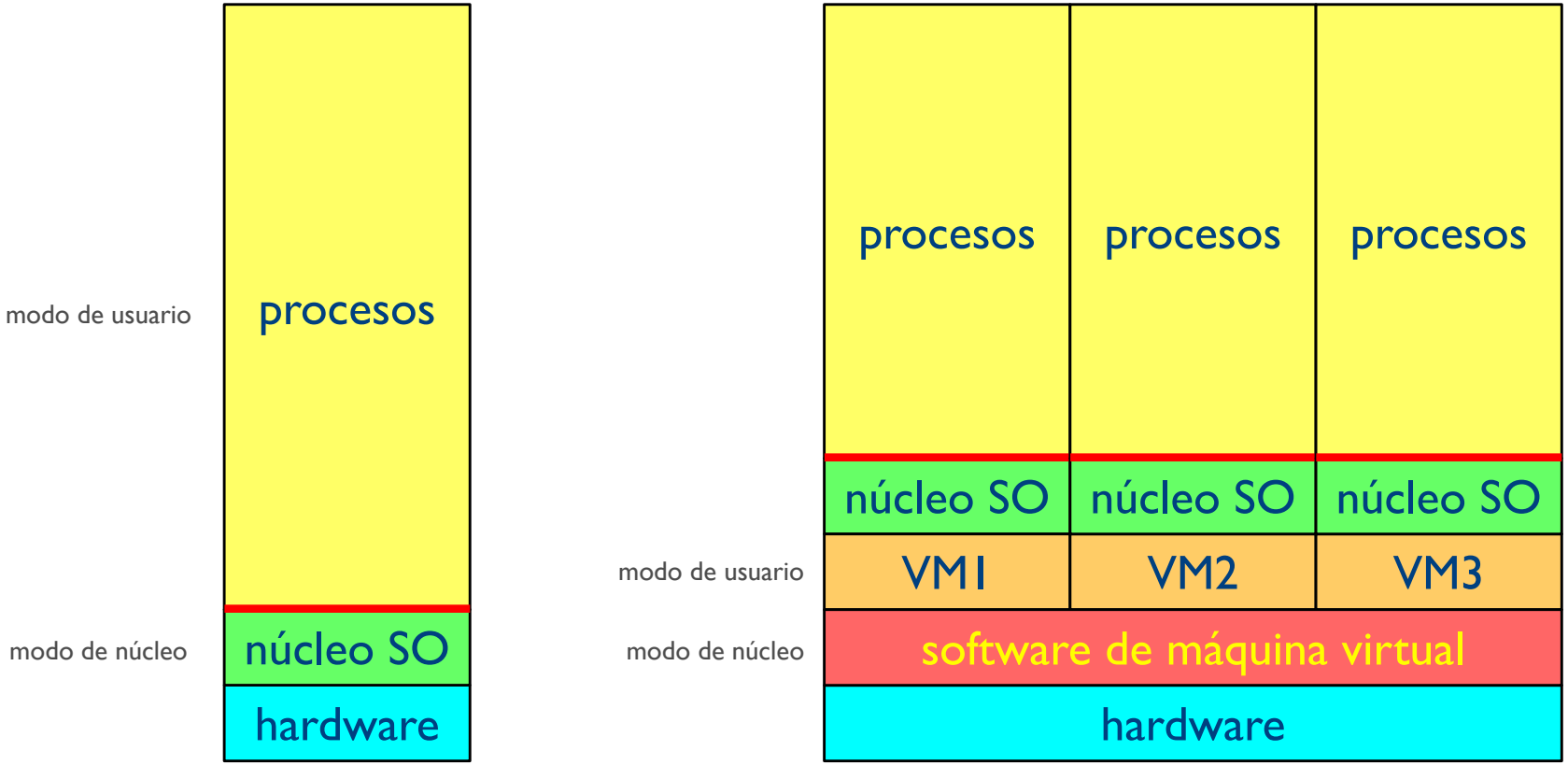
3. Introducción a las máquinas virtuales

- **Máquina virtual (virtual machine, VM)**
  - Software que ejecuta programas como si fuera la máquina física
- **Se abstrae el hardware y se representa mediante una capa de software**
  - interfaz idéntica al hardware
- **Apariencia de varios procesadores idénticos**
- **Anfitrión (host): el hardware real**
  - posiblemente con un SO nativo
- **Invitado (guest): el SO que se instala en la máquina virtual**
  - puede haber varios SO invitados, cada uno en una

© 2021 Grupo STRAST - UPM

Arquitectura con máquinas virtuales: Tipo 1

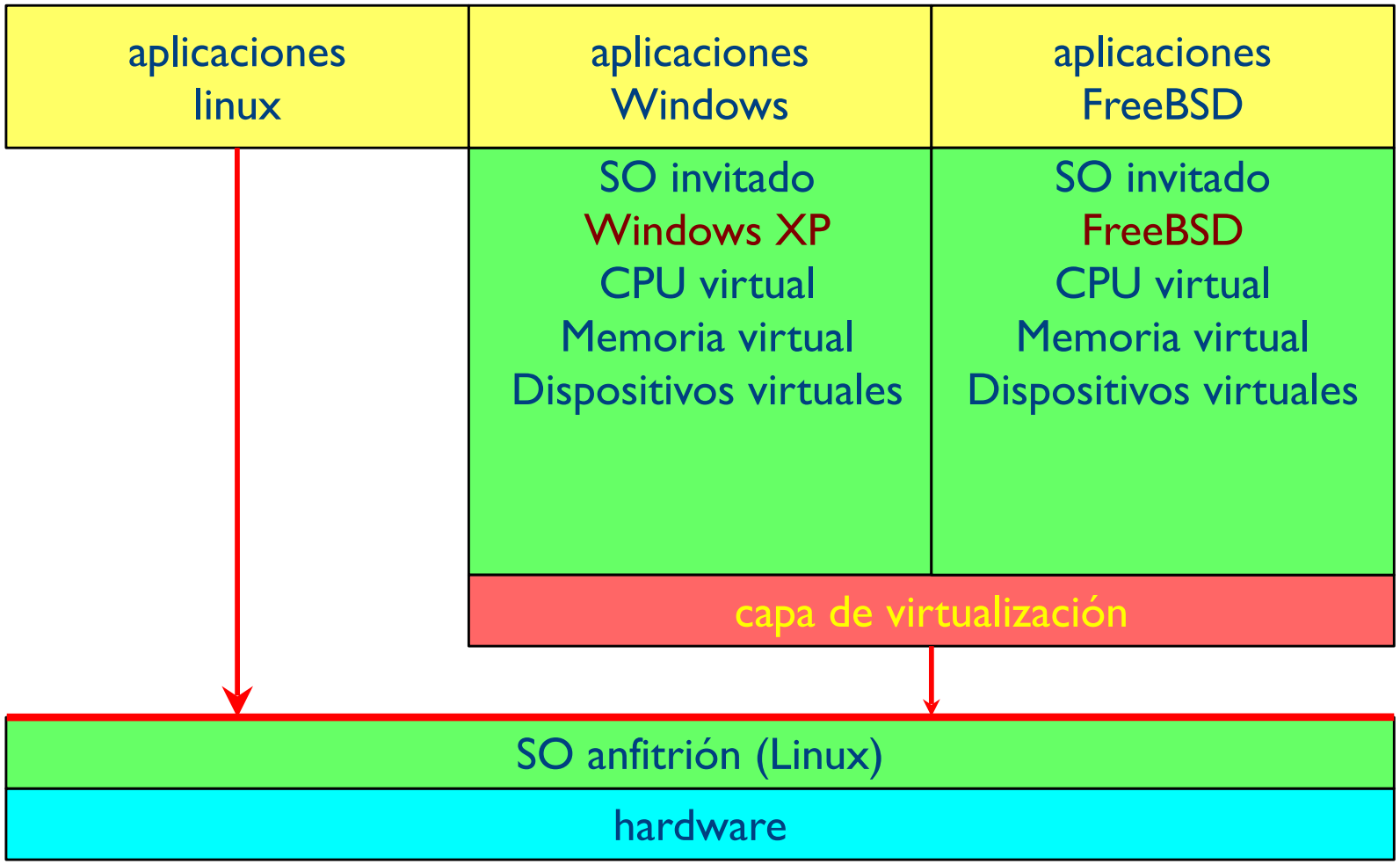
- **El hipervisor ejecuta sobre el hardware**



© 2021 Grupo STRAST - UPM

Arquitectura con Vmware WS: Tipo 2

- **El hipervisor ejecuta sobre un SO**



© 2021 Grupo STRAST - UPM

Técnicas relacionadas

- **Emulación**
  - se simula un procesador distinto del anfitrión
  - muy costoso en tiempo de ejecución
    - hay que interpretar todas las instrucciones de máquina del procesador invitado
  - útil para ejecutar software antiguo
- **Para-virtualización**
  - la plataforma de máquina virtual presenta una interfaz similar, pero no idéntica, a la arquitectura de máquina original
  - el software invitado debe modificarse para ejecutarse en la máquina virtual

© 2021 Grupo STRAST - UPM

Resumen

- **Objetivos SO:**
  - Gestión eficiente de recursos
  - Interfaz adecuado con usuario
- **Sistema Operativo**
  - funcionamiento de computadores
  - Dirigido por interrupciones
  - Núcleo: se ejecuta en modo protegido
- **Componentes fundamentales**
  - Gestión de procesos
  - Gestión de memoria
  - Gestión de E/S

© 2021 Grupo STRAST - UPM

Referencias

- A. Silberstchatz, P. B. Galvin, G. Gagne: *Operating System Concepts*, Global ed. John Wiley, 2019
- A. Silberschatz, P. Galvin y G. Gagne: ***Operating System Concepts with Java***. 8ª edición. Addison Wesley, 2011. (Safari)
- G. Fernández, ***Conceptos básicos de Arquitectura y Sistemas Operativos***, 5ª Edición. Publicaciones ETSIT, 2005.

© 2021 Grupo STRAST - UPM