

Sistemas Operativos

Grado en Ingeniería Informática del Software

Tema 1: Introducción

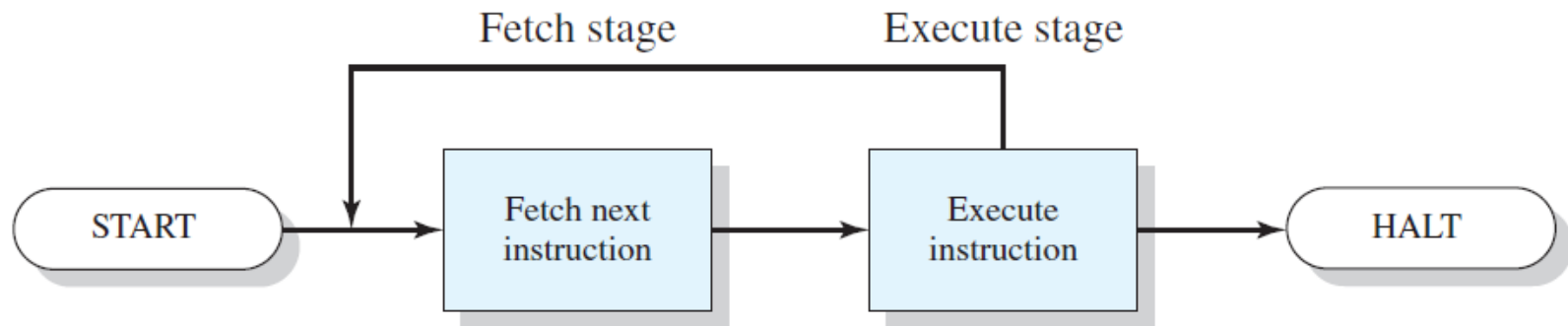


Conceptos básicos

- **Ordenador:** Máquina electrónica capaz de almacenar información y tratarla automáticamente mediante operaciones matemáticas y lógicas *controladas por programas informáticos*.
- **Programa:** conjunto de instrucciones a ejecutar en un ordenador para resolver algún problema.
- **Proceso:** resultado de la ejecución de un programa.

Conceptos básicos

- **Ciclo de ejecución del procesador:** bucle Fetch-Decode-Execute, que es lo único que sabe hacer un procesador.



- **Interrupción Hardware:** señal que interrumpe el ciclo normal de ejecución del procesador, haciéndolo saltar a otra instrucción.
- **Instrucción TRAP:** instrucción del procesador que cuando se ejecuta genera una interrupción.

Conceptos básicos

- **Modo usuario del procesador:** modo de ejecución donde el procesador tiene algunas restricciones (no puede usar todas el conjunto de instrucciones completo, no puede acceder a todo el espacio de direcciones, ...)
- **Modo núcleo del procesador:** modo de ejecución donde el procesador puede ejecutar el conjunto de instrucciones completo y puede acceder a cualquier dirección de memoria.

Partes de un ordenador

- **Procesador:** Calcula. Ejecuta. Almacena información en registros:
 - Unos de control (PC, IR, PSW, ...)
 - Otros accesibles por el programa.
- **Memoria principal:** almacena datos y programas.
- **Dispositivos E/S:** conectan el ordenador con el mundo.
- **Bus del sistema:** conecta las distintas partes.

Índice de contenidos

- 1. Concepto y funciones del Sistema Operativo**
2. Arranque y activación del Sistema
3. Tipos de Sistemas Operativos
4. Componentes de los Sistemas Operativos
5. Interfaces que ofrece el Sistema Operativo
6. Diseño de los Sistemas Operativos
7. Evolución de los Sistemas Operativos

1.- Concepto y funciones del SO

Podemos usar un ordenador sin Sistema Operativo?

¿Qué es un Sistema Operativo?

- ¿Qué hace?*
- ¿Cuál es su función?*
- ¿Qué proporciona?*



K1.8-

1.- Concepto y funciones del SO

*Una computadora moderna consta de uno o más procesadores, una memoria principal, discos, impresoras, un teclado, un ratón, una pantalla o monitor, interfaces de red y otros dispositivos de entrada/salida. En general es un sistema complejo. Si todos los programadores de aplicaciones tuvieran que comprender el funcionamiento de todas estas partes, no escribirían código alguno. Es más: el trabajo de administrar todos estos componentes y utilizarlos de manera óptima es una tarea muy desafiante. Por esta razón, las computadoras están equipadas con una capa de software llamada **sistema operativo**, cuyo trabajo es proporcionar a los programas de usuario un modelo de computadora mejor, más simple y pulcro, así como encargarse de la administración de todos los recursos antes mencionados.*

Andrew S. Tanenbaum

1.- Concepto y funciones del SO

- “Un sistema operativo es un programa cuyo objetivo es **simplificar el manejo** y la utilización de la computadora, **haciéndolo seguro y eficiente**” *Jesús Carretero*
- El sistema operativo es como el director. El **responsable de coordinar todos los componentes individuales del computador**, de forma que operan juntos siguiendo un solo plan. El sistema operativo asigna recursos del computador a los diversos programas, sincroniza sus actividades y proporciona los mecanismos adecuados para que los programas se ejecuten en perfecta armonía. *Gary Nutt*
- Un sistema operativo es un programa que **controla la ejecución de los programas** de aplicación y que **actúa como interfaz** entre las aplicaciones del usuario y el hardware de un computador. Puede considerarse que un sistema operativo tiene 3 objetivos: **Comodidad, eficiencia y capacidad de evolución**. *William Stallings*.

1.- Concepto y funciones del SO

¿Qué es un Sistema Operativo?

Es como....

Gestor de recursos

Cerebro

Director de orquesta

Da ordenes al HW

*Coordina todas las operaciones pedidas
optimizando resultados*

Máquina extendida

Dependiente

Intermediario

*Recibe peticiones del usuario (final o
programa) y las gestiona (enviando órdenes al
HW)*

Índice de contenidos

1. Concepto y funciones del Sistema Operativo
- 2. Arranque y activación del Sistema**
3. Tipos de Sistemas Operativos
4. Componentes de los Sistemas Operativos
5. Interfaces que ofrece el Sistema Operativo
6. Diseño de los Sistemas Operativos
7. Evolución de los Sistemas Operativos

2.- Arranque y activación del SO

- Cuando el ordenador está trabajando, ejecuta un programa *cargado en memoria principal*.
- Usamos el interfaz de usuario para ejecutar una aplicación. Esta interfaz es un programa.
- Pero cuando el ordenador está apagado, la memoria principal está vacía.

¿Cómo puede cargarse el SO en memoria principal?



2.- Arranque y activación del SO

Proceso de arranque (boot)

1. Cuando se enciende un ordenador, su diseño hace que la CPU salte a una determinada dirección de memoria, donde el fabricante ha dejado grabado un **programa** que ejecuta varias tareas:
 - Comprueba la configuración existente de hardware.
 - Verifica el estado actual del hardware.
 - Carga el Sistema Operativo.
 - Finalmente, el SO lanza algún tipo de interfaz de usuario, de manera que éste puede ejecutar sus programas.
2. Este proceso no es tan simple.

Mira este [vídeo](#) para una explicación detallada.

2.- Arranque y activación del SO

Activación del SO

Una vez que el usuario ha lanzado una aplicación, ¿cuándo retoma el control el SO?

- 1.
- 2.
- 3.
- 4.



2.- Arranque y activación del SO

Activación del SO

Una vez que el usuario ha lanzado una aplicación, ¿cuándo retoma el control el SO?

- 1. Cuando finaliza la aplicación.**
- 2. Cuando la aplicación realiza ciertas operaciones ilegales.**
- 3. Cuando la aplicación necesita que el sistema ejecute alguna operación especial en su nombre.**
- 4. Cuando algún dispositivo necesita atención.**

El mecanismo para implementar cada situación es único: ***Interrupciones.***

2.- Arranque y activación del SO

Activación del SO

Gestión de interrupciones

- Cada vez que aparece una interrupción, la CPU deja de hacer lo que estaba haciendo y ejecuta alguna rutina del SO.

1. Almacenar el estado del PC y el Registro de Estado.
2. Utilizar la tabla de *Vectores de Interrupciones* (o de *Descriptor de Interrupción*) para establecer el Nuevo valor del PC (dirección a la que saltar).
3. Pasar a modo Kernel.
4. **Ejecutar la rutina de tratamiento de la interrupción.**
5. Reanudar la ejecución del proceso interrumpido (o de otro).
 1. Restaurar los valores almacenados del PC y Registro de estado.
 2. Volver a modo usuario.

Función del Hardware

Función del SO

Repaso

Concepto de SO; Arranque y Activación del SO



K2

Índice de contenidos

1. Concepto y funciones del Sistema Operativo
2. Arranque y activación del Sistema
- 3. Tipos de Sistemas Operativos**
4. Componentes de los Sistemas Operativos
5. Interfaces que ofrece el Sistema Operativo
6. Diseño de los Sistemas Operativos
7. Evolución de los Sistemas Operativos

3.- Tipos de Sistemas Operativos

- **¿Qué tipos de Sistemas Operativos conoces?: Analiza el significado de los siguientes tipos:**
 - Monotarea / Multitarea
 - Monoproceso / Multiproceso
 - Monousuario / Multiusuario
 - Interactivo / Por lotes (batch)
 - Monoprocesador / Multiprocesador
 - Empotrados
 - De tiempo real
 - Móviles



3. Tipos de Sistemas Operativos

Virtualización

- La virtualización se introdujo en el sistema operativo VM de IBM en 1972. Este sistema operativo es capaz de crear “máquinas virtuales”, un programa que *emula* el comportamiento de un sistema real.
- Tras años en el olvido, la virtualización ahora se utiliza ampliamente en entornos de PCs y servidores.
 - Permite tener diferentes sistemas (incluso con diferentes SO) ejecutándose al mismo tiempo en la misma máquina.
 - Optimiza el uso de la CPU, dado que normalmente un sistema único es incapaz de utilizar toda la potencia del sistema.

3. Tipos de Sistemas Operativos

Virtualización

- El software que crea y gestiona máquinas virtuales se suele llamar “*hipervisor*”.
- Hay dos posibles implementaciones:
 - Hypervisor “Bare metal” hipervisor (type 1): el hipervisor se ejecuta directamente sobre el hardware, sin SO.
 - Hypervisores ejecutándose sobre el SO (hipervisores tipo 2): el hipervisor es un programa que se ejecuta sobre un sistema operativo instalado sobre el hardware.

3. Tipos de Sistemas Operativos Virtualización

- Algunos ejemplos:
 - Hipervisores tipo 1:
 - VMware ESX, ESXi
 - Microsoft Hyper-V
 - Citrix XEN Server
 - Oracle VM
 - Hipervisores tipo 2
 - VMware Server, Player, Workstation, ...
 - Microsoft Virtual PC
 - Citrix Xex Desktop
 - Oracle Virtual Box

Índice de contenidos

1. Concepto y funciones del Sistema Operativo
2. Arranque y activación del Sistema
3. Tipos de Sistemas Operativos
- 4. Componentes de los Sistemas Operativos**
5. Interfaces que ofrece el Sistema Operativo
6. Diseño de los Sistemas Operativos
7. Evolución de los Sistemas Operativos

4.- Componentes del SO

1. Desde el punto de vista funcional el SO contiene

- Sistema de gestión de procesos.
- Sistema de gestión de memoria.
- Sistema de gestión de dispositivos.
- Sistema de gestión de archivos.
- Seguridad y protección

¿Qué posibles servicios ofrecerá cada uno de estos sistemas?

.... los estudiaremos en cada uno de los temas de la asignatura



Índice de contenidos

1. Concepto y funciones del Sistema Operativo
2. Arranque y activación del Sistema
3. Tipos de Sistemas Operativos
4. Componentes de los Sistemas Operativos
- 5. Interfaces que ofrece el Sistema Operativo**
6. Diseño de los Sistemas Operativos
7. Evolución de los Sistemas Operativos

5.- Interfaces que ofrece el SO

1. Interfaz de programación de aplicaciones o API (*Application Program Interface*)

- Estándar POSIX (**P**ortable **O**perating **S**ystem; Interface; X Unix)
 - Se trata de una especificación, no de una implementación
- SUS (Single Unix Specification) está sustituyendo a Posix
- Interfaz Win32
 - No es un estándar genérico, es una implementación de Microsoft

2. Interfaz de usuario

- Línea de órdenes (CLI)
- Interfaces gráficas (GUI)
- Ficheros de órdenes o shell-scripts
 - Bourne de Unix,
 - cmd-line, Jscript, PowerShell... de Windows

5.- Interfaces que ofrece el SO APIs

```
void _tmain( int argc, TCHAR *argv[] )
{
    STARTUPINFO si;
    PROCESS_INFORMATION pi;

    ZeroMemory( &si, sizeof(si) );
    si.cb = sizeof(si);
    ZeroMemory( &pi, sizeof(pi) );

    if( argc != 2 )
    {
        printf("Usage: %s [cmdline]\n", argv[0]);
        return;
    }

    // Start the child process.
    if( !CreateProcess( NULL,    // No module name (use command line)
        argv[1],                // Command line
        NULL,                   // Process handle not inheritable
        NULL,                   // Thread handle not inheritable
        FALSE,                  // Set handle inheritance to FALSE
        0,                      // No creation flags
        NULL,                   // Use parent's environment block
        NULL,                   // Use parent's starting directory
        &si,                     // Pointer to STARTUPINFO structure
        &pi )                   // Pointer to PROCESS_INFORMATION structure
    )
    {
        printf( "CreateProcess failed (%d).\n", GetLastError() );
        return;
    }
}
```

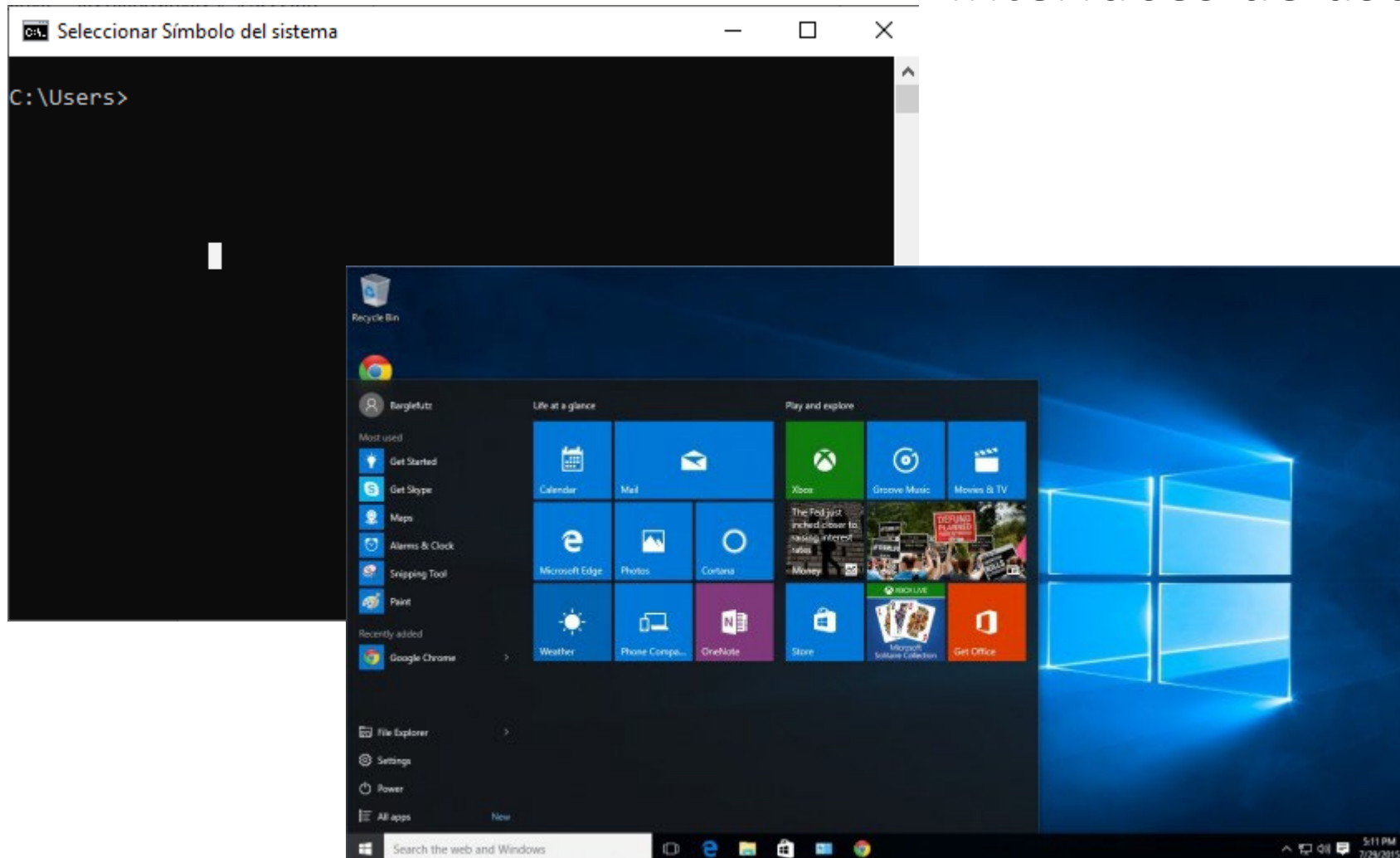
```
#include <stdio.h>    /* printf, stderr, fprintf */
#include <sys/types.h> /* pid_t */
#include <unistd.h>    /* _exit, fork */
#include <stdlib.h>    /* exit */
#include <errno.h>     /* errno */

int main(void)
{
    pid_t pid;

    /* Output from both the child and the parent process
     * will be written to the standard output,
     * as they both run at the same time.
     */
    pid = fork();
    if (pid == -1)
    {
        /* Error:
         * When fork() returns -1, an error happened
         * (for example, number of processes reached the limit).
         */
        fprintf(stderr, "can't fork, error %d\n", errno);
        exit(EXIT_FAILURE);
    }
}
```

5.- Interfaces que ofrece el SO

Interfaces de usuario



5.- Interfaces que ofrece el SO

Interfaces de usuario



Índice de contenidos

1. Concepto y funciones del Sistema Operativo
2. Arranque y activación del Sistema
3. Tipos de Sistemas Operativos
4. Componentes de los Sistemas Operativos
5. Interfaces que ofrece el Sistema Operativo
- 6. Diseño de los Sistemas Operativos**
7. Evolución de los Sistemas Operativos

6.- Diseño de los SO

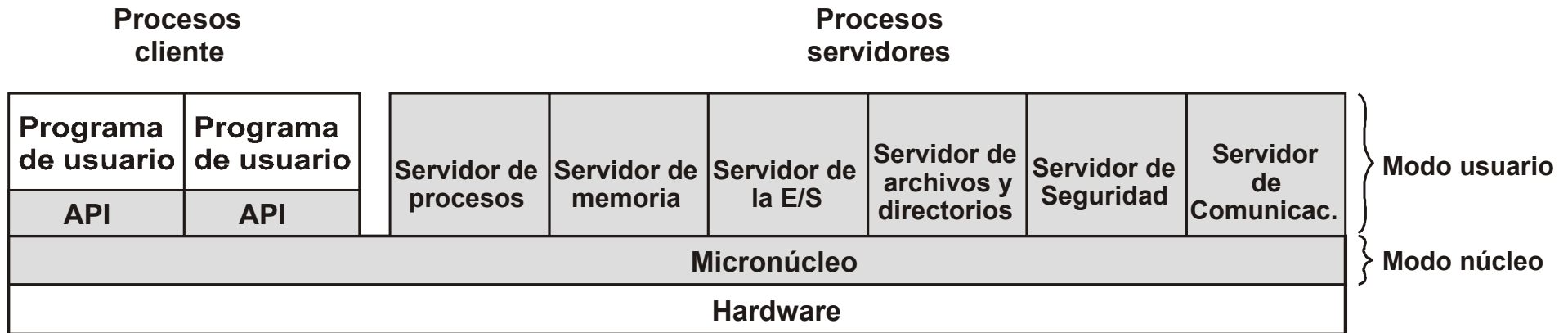
Arquitectura del sistema

- Sistemas Operativos Monolíticos
 - Funcionalidad en un único programa
 - Ejecución en modo núcleo
 - Difícil modificación
- Sistemas Operativos Estructurados
 - Por capas
 - Micronúcleos (Cliente/servidor)
- Sistemas Operativos Híbridos
 - Modular
 - Micronúcleo no puro
- Sistemas Operativos distribuidos
 - Puros
 - Middleware

6.- Diseño de los SO

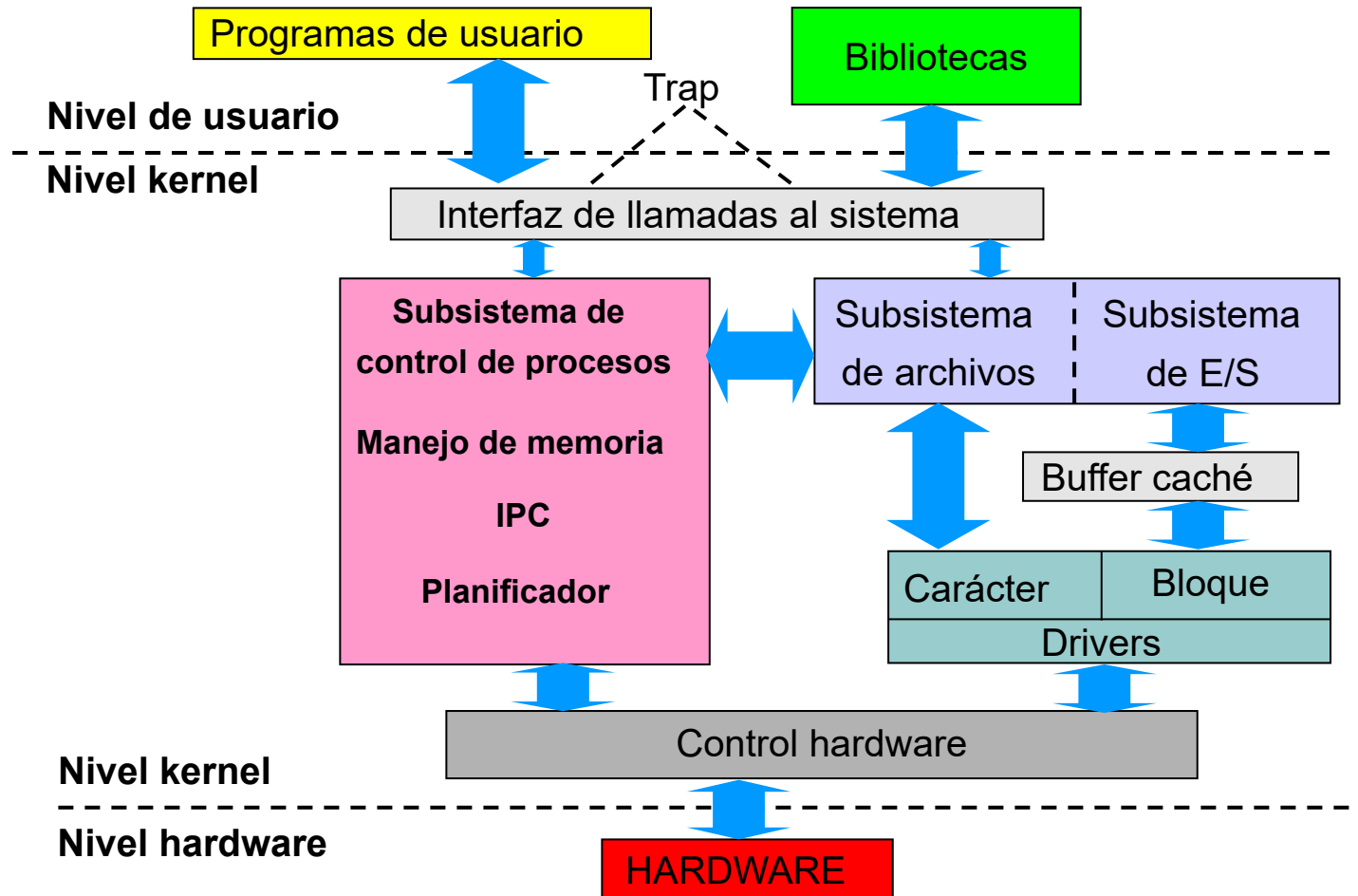
Arquitectura del sistema

- Arquitectura de micronúcleo puro (cliente servidor)



6.- Diseño de los SO

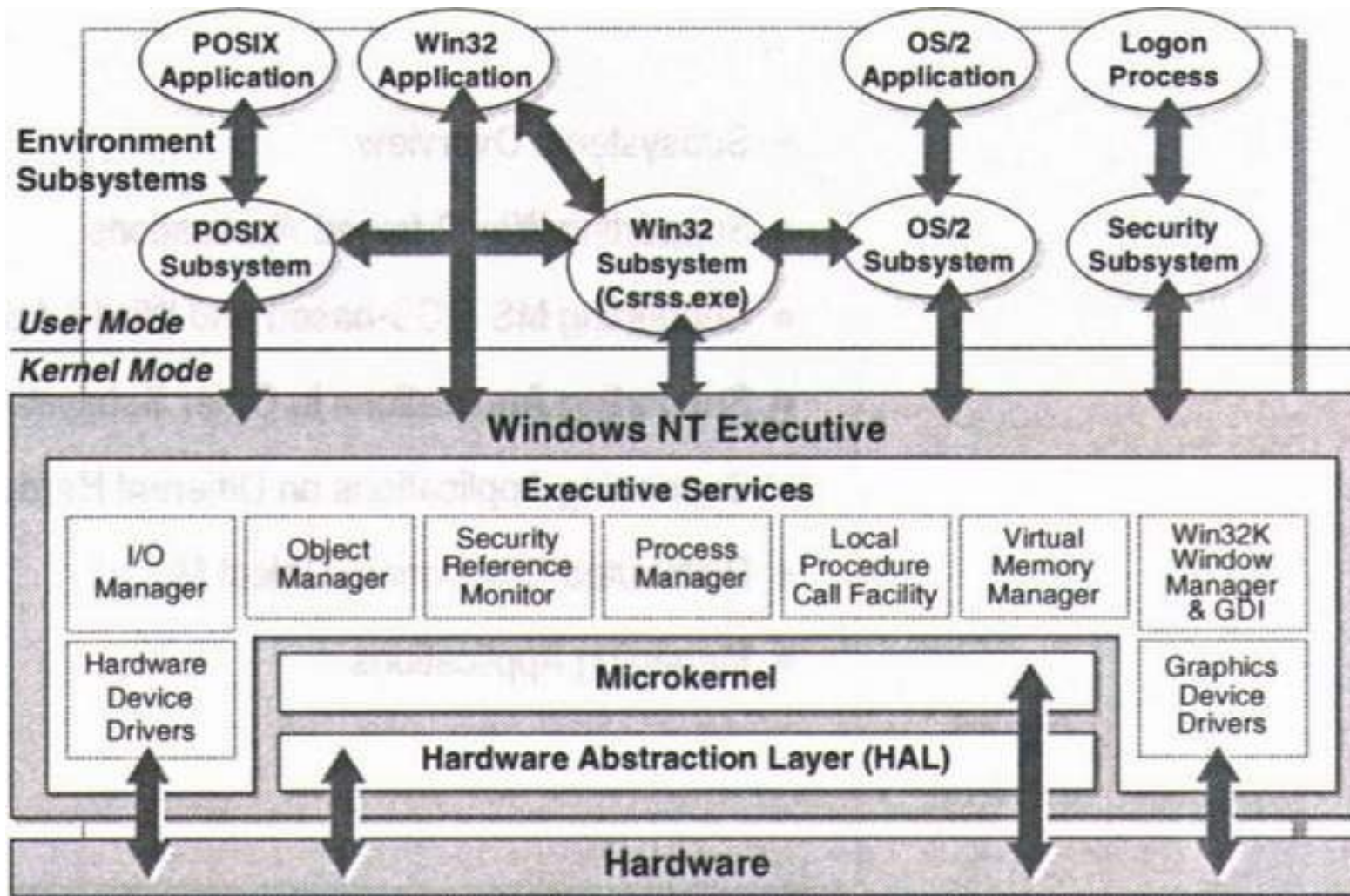
Diseño del SO Unix



©Sebastián
Sánchez
Prieto. Un.
Alcalá

6.- Diseño de los SO

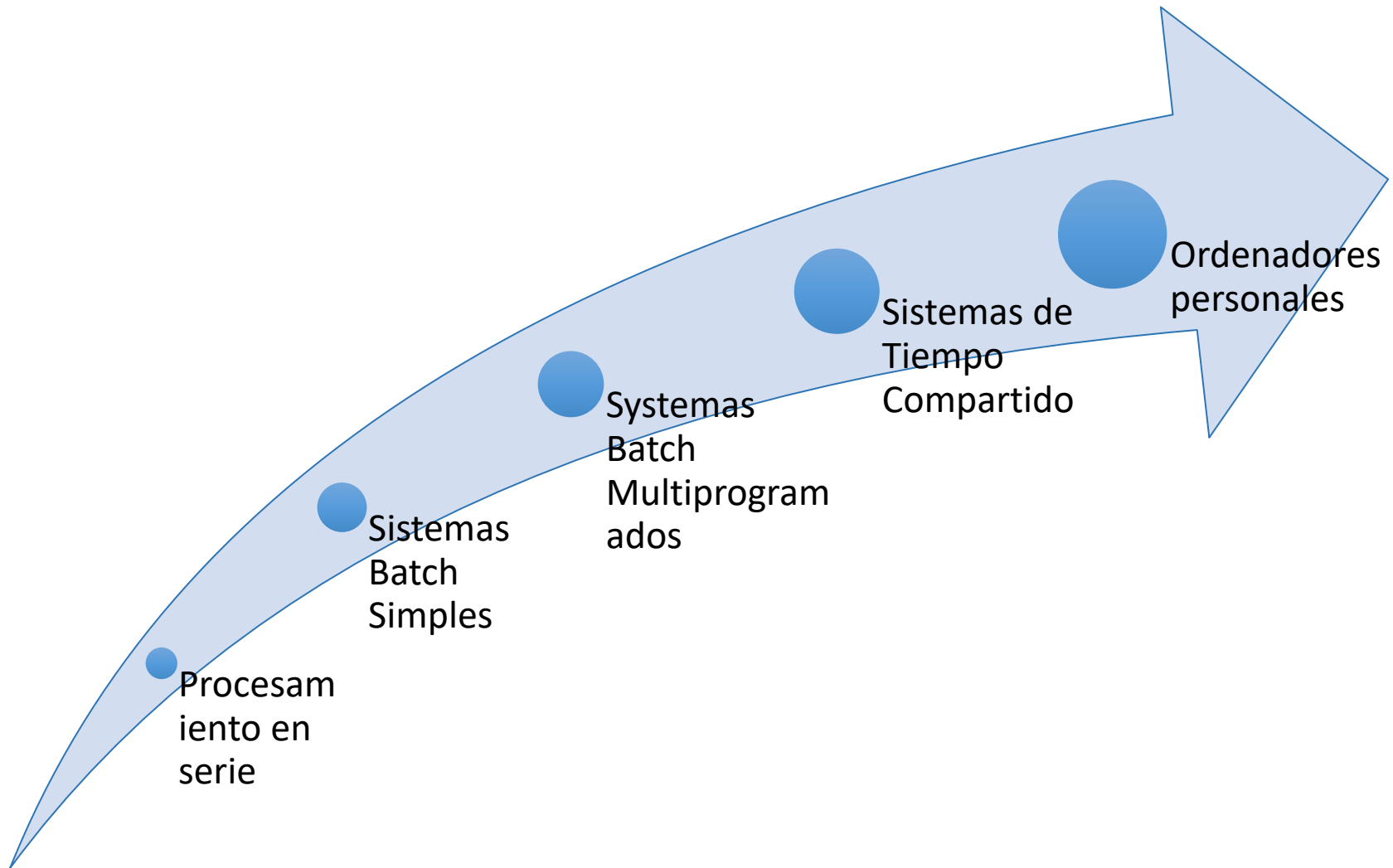
Diseño del SO Windows NT



Índice de contenidos

1. Concepto y funciones del Sistema Operativo
2. Arranque y activación del Sistema
3. Tipos de Sistemas Operativos
4. Componentes de los Sistemas Operativos
5. Interfaces que ofrece el Sistema Operativo
6. Diseño de los Sistemas Operativos
- 7. Evolución de los Sistemas Operativos**

7.- Evolución de los SO



7.- Evolución de los SO

Época	Características del Sistema Operativo	HW/ SO lenguajes
Años 40 Prehistoria	Inexistente. Procesamiento en serie	Válvulas Arq. Von Neumann EDVAC ENIAC (1943) Ensamblador
Años 50 1ª Generación	Monitor residente: Carga y ejecución de programas Procesamiento de trabajos por lotes Rutinas de e/s (manejadores de dispositivos) Recuperación de errores Lenguaje de control	Transistores IBSYS (IBM) FMS (IBM) FORTRAN, COBOL

7.- Evolución de los SO

Época	Características del Sistema Operativo	HW/ SO lenguajes
Años 60 2ª Generación	<p>Incrementar rendimiento</p> <ul style="list-style-type: none">• Multiprogramación (E/S por ADM)• Multiusuario interactivo: Tiempo compartido• Tiempo Real• Multiprocesador <p>Grandes y costosos</p> <p>Complejo lenguaje de control</p>	<p>Circuitos integrados</p> <p>PDP-8 Digital</p> <p>IBM 360</p> <p>CTSS (IBM 7090)</p> <p>OS/360(IBM 360)</p> <p>MULTICS – UNIX</p> <p>BASIC, ALGOL</p>

7.- Evolución de los SO

Época	Características del Sistema Operativo	HW SO lenguajes
Años 70	<p>SO de propósito general Difusión de multiusuario y tiempo compartido <u>Unix</u></p> <ul style="list-style-type: none"> •Laboratorios Bell ATT •Ken Thompson, Dennis Ritchie •Implementación en Lenguaje C (Dennis Ritchie) (1973) •Difusión a lab. y universidades (cod. Fuente) •Distintas distribuciones <ul style="list-style-type: none"> •Aparición de BSD (Uni. Berkeley) •Aparición de System V (Lab Bell) •Otros fabricantes (Sun, HP, IBM)... 	<p>PDP-7,11 Apple II Intel 8008 UNIX (Bell 1976) MVS (IBM) - Mainframes VM (IBM) Miniordenadores CP/M- PCs</p>

7.- Evolución de los SO

Época	Características del Sistema Operativo	HW/ SO/ lenguajes
Años 80	<p>Simplificación de los SO. Importancia usuario</p> <p>Gestión de redes</p> <p>Sistemas operativos de red</p> <p>Interfaces gráficas</p> <p>Diseño interno Orientado a Objetos</p> <p>Diferentes SO para diferentes procesadores de ordenadores personales</p> <p>Sistemas Operativos Distribuidos</p>	<p>UNIX</p> <p>MS-DOS</p> <p>Windows, Amiga</p> <p>Mc OS, OS/2</p> <p>Mach, Chorus,</p> <p>Amoeba</p>

7.- Evolución de los SO

Época	Características del Sistema Operativo	HW/ SO lenguajes
Años 90	<p>Sistemas Operativos Libres</p> <p>Sistemas Operativos de tiempo real</p> <p>Sistemas Operativos con procesamiento paralelo</p> <p>Capas de software intermedio (Middleware)</p> <p>Arquitectura cliente/servidor</p> <p>Estandarización de Interfaces</p> <p>Inclusión de múltiples interfaces de programación</p> <p>Seguridad, criptografía</p> <p>Sistemas Operativos para enseñanza</p>	<p>Linux (91) FreeBSD</p> <p>QNX</p> <p>CORBA, DCOM</p> <p>Windows XP, NT..</p> <p>POSIX</p> <p>MINIX, SOS,</p> <p>NACHOS</p>

7.- Evolución de los SO

Época	Características del Sistema Operativo	HW/ SO lenguajes
Presente y futuro	<p>Sistemas operativos empotrados: móviles, PDAs, tabletas...</p> <p>Paralelismo</p> <p>Computación distribuida</p> <p>Tolerancia de fallos</p> <p>Desarrollo de nuevas interfaces</p> <p>Personalización y usabilidad</p> <p>Sistemas Abiertos</p> <p>Diseños Orientados a Objetos</p> <p>Personalidades múltiples</p> <p>Arquitectura cliente servidor distribuida</p> <p>Incorporación de múltiples utilidades de seguridad y acceso remoto</p> <p>Mejora notable de la seguridad (esperemos)</p>	<p>Android, IOS</p> <p>Linux (RedHat Slackware SuSe...)</p> <p>Windows 8 Windows 10</p>