

## TEMA 1. Introducción

- Definición y objetivos del sistema operativo
- Historia de los S.O
- Hardware de la computadora
- Tipos de S.O
- Conceptos básicos
  - Llamadas al sistema
  - Procesos
  - Ficheros
  - Intérprete de comandos
- Componentes y Servicios de un S.O
- Estructuras de S.O

1

## Definición y objetivos del S.O

- Conjunto de programas que se encargan de:
  - Ocultar toda la complejidad del hardware al programador
    - Presenta un conjunto de funciones más fácil de programar que el hardware.
    - ej: LEER BLOQUE DE FICHERO
    - Presenta al usuario una máquina virtual
  - Administrar los recursos de la máquina
    - Asigna los recursos (procesadores, memorias, dispositivos de entrada/salida) de forma ordenada
    - Lleva la cuenta de quién emplea los recursos, contabiliza su utilización y decide la concesión a diferentes programas o usuarios en caso de conflicto



2

## Historia de los S.O

- Historia de la arquitectura de computadores
  - La arquitectura de los computadores ha ido evolucionando a lo largo de la historia.
  - Se divide la historia en distintas etapas llamadas generaciones.

## Generación 0: 1642-1945

- Tecnología:
  - Computadores mecánicos o electromecánicos con muchas limitaciones.
- Personas destacadas:
  - [Blaise Pascal](#) construyó en 1642 una máquina calculadora para sumar y restar.
  - [Charles Babbage](#) construyó en 1834 una máquina de propósito general (almacén, taller y sección de E/S). Contrató a Ada para la programación de la máquina.
  - Aiken construyó la Mark I en 1944, inspirado en los estudios de Babbage.

## 1ª Generación: 1945-1955

- Tecnología:
  - Válvula electrónica de vacío.
- Modelos:
  - ENIAC (1946): 18.000 válvulas, 30 toneladas, 1400 m2, 100 Kw, 5.000 sumas por segundo.
  - EDSAC (1949): primer ordenador con programa almacenado.
  - UNIVAC: primer ordenador comercial.
- Personas destacadas:
  - Jonh Von Neumann establece un modelo de la estructura de un ordenador (memoria, U.A.L., U. de control y U. de E/S). Crea la idea de computador con programa almacenado.

5

## 1ª Generación

- Modo de funcionamiento:
  - Se programa en lenguaje máquina, propio de cada máquina y muy complicado.
  - Se desconocen los leng. de programación.
  - No existe S.O.
  - Se realiza el programa cableado, se solicita hora para la máquina, se inserta el panel de conexiones en el computador para ejecutar el programa.
  - Se resolvían cálculos numéricos.
  - A principios de los 50 se mejoró el procedimiento con las tarjetas perforadas.

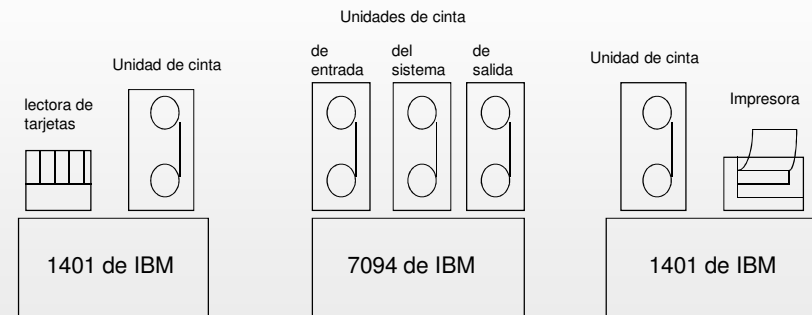
6

## 2ª Generación: 1955-1965

- **Tecnología:**
  - Transistor (Bardeen-Brattain, 1947). Ventajas: menor espacio, menor consumo, más barato y mayor fiabilidad. Esto hace disminuir el precio y tamaño de los computadores.
- **Modelos:**
  - PDP-1 de DIGITAL
- **Modo de funcionamiento:**
  - Lenguajes de alto nivel : FORTRAN, COBOL, ALGOL, PL/1. Se escribe el programa en papel, se perfora en tarjetas, se lleva al operador, se recoge el listado de impresora.
  - Sistema de procesamiento por lotes (con S.O.) - batch

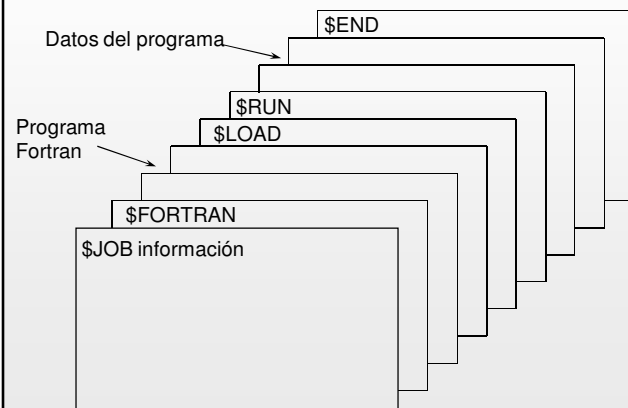
7

## Sistema de procesamiento por lotes



8

## Ejemplo de procesamiento por lotes



9

## 3ª Generación: 1965-1980

- **Tecnología:**
  - Circuitos integrados SSI (hasta 100 componentes) y MSI (100-3000)
- **Modelos:**
  - IBM sistema 360 y PDP-8 (DIGITAL)
- **Modo de funcionamiento:**
  - Lenguajes de alto nivel BASIC y PASCAL
  - S.O con multiprogramación:
    - División de la memoria.
    - Procedimientos de spooling (operación simultánea de periféricos conectados en línea).
    - Tiempo compartido.

10

## 4ª Generación: 1980-1990

- Tecnología:
  - Se integra la UCP en un sólo chip: el microprocesador.
  - Circuitos integrados LSI (3000-30000) y VLSI (más de 30000)
- Modelos:
  - IBM PC (1981), IBM PC XT (1982), IBM PC AT (1984), IBM PS/2 (1987), VAX (DIGITAL, 1980), CRAY X-MP (1983)
- Modo de funcionamiento:
  - Software fácil de usar.
  - Sistemas operativos MS-DOS, UNIX.
  - Sistemas operativos de red y sistemas operativos distribuidos.

11

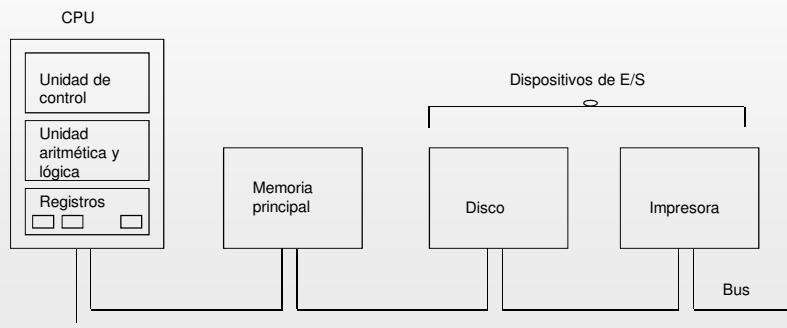
## 5ª Generación: 1990 en adelante

- Tecnología:
  - Circuitos con más de un millón de componentes.
  - Nuevas arquitecturas: paralelismo.
  - Tecnología óptica.
- Modelos:
  - CONNECTION MACHINE, máquina masivamente paralela.
- Modo de funcionamiento:
  - Inteligencia artificial y sistemas expertos.

12

## Hardware de la computadora


- Procesador, memoria, dispositivos de E/S



## Tipos de Sistemas Operativos

- De computadores centrales (mainframe)
- De servidores
- De multiprocesadores
- De computadoras personales
- De computadoras de bolsillo
- De sistemas integrados (embedded)
- De nodos sensores
- En tiempo real (hard / soft)
- De tarjetas inteligentes

## Conceptos básicos

- Llamadas al sistema 
- Procesos
- Ficheros
- Intérprete de comandos

15

## Llamadas al sistema

- Los programas de usuario se comunican con el S.O y requieren sus servicios mediante **llamadas al sistema**
- A cada llamada al sistema le corresponde un procedimiento que el programa de usuario puede llamar
- El procedimiento se encarga de iniciar la llamada al sistema (mediante un TRAP)
- Son distintas en cada sistema
- ej: `count = read (file, buffer, nbytes);`

16



# Portions of Generated Motorola 68000 Assembler Code

Addr Instruction

# code for main

```
58:  mov    &0x1b6, (%sp)    # move 0666 onto stack
5e:  mov    &0x204, -(%sp)    # move stack ptr
                        # and move variable "name" onto stack
64:  jsr     0x7a              # call C library for creat
```

# library code for creat

```
7a:  movq   &0x8, %d0         # move data value 8 into data register 0
7c:  trap   &0x0              # operating system trap
7e:  bcc    &0x6 <86>         # branch to addr 86 if carry bit clear
80:  jmp     0x13c             # jump to addr 13c
86:  rts                    # return from subroutine
```

# library code for errors in system call

```
13c:  mov    %d0, &0x20e       # move data reg 0 to location 20e (errno)
142:  movq   &-0x1, %d0        # move constant -1 into data register 0
144:  mova   %d0, %a0
146:  rts                    # return from subroutine
```

```
char name[] = "file";
main()
{
```

```
    int fd;
    fd = creat(name, 0666);
}
```

"The design of the UNIX Operating System"  
Maurice J. Bach  
Llamada al sistema Creat  
Y porción de código ensamblador


17

```
char name[] = "file";
main()
{
    int fd;
    fd = creat(name, 0666);
}
```

"The design of the UNIX Operating System"  
Maurice J. Bach  
Llamada al sistema Creat  
Y porción de código ensamblador

17

## Procesos (I)

- Programa en ejecución:
  - Código ejecutable del programa
  - Datos
  - Pila del programa
  - Contador de programa
  - Puntero a la pila y otros registros
  - Toda la información necesaria para ejecutar el programa
- Tabla de procesos
- Un proceso consta de: 
  - Una imagen de memoria (Espacio de direcciones del proceso)
  - Una entrada en la tabla de procesos

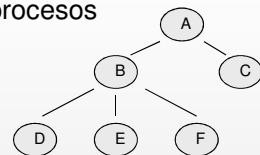
18

Los procesos tienen una estructura en árbol. En la imagen el proceso A es el proceso padre del proceso B y C. El proceso B es el proceso hijo del proceso A. El Shell del sistema operativo, que es donde introducimos las ordenes al sistema operativo, crea un proceso hijo que es quien verdaderamente ejecuta el proceso. Señales: son el equivalente en software de las interrupciones hardware y su tratamiento es similar. Cuando se produce una señal, el proceso que se está ejecutando se interrumpe, se almacena la información para continuar después con el proceso y se comienza a ejecutar un manejador de la señal. Una vez que finaliza el procesamiento de la señal, el proceso anterior sigue por donde iba.

UID => identificador de usuario. Cuando un proceso arranca lo hace con el UID del usuario que lo ha arrancado.

## Procesos (y II)

- Estructura en árbol de los procesos



- Señales

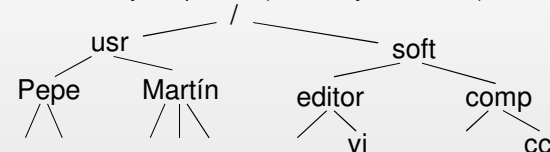
- Equivalente software a las interrupciones hardware
  - El proceso suspende temporalmente lo que estaba haciendo, salva todos sus registros.
  - Comienza a ejecutar un procedimiento especial de tratamiento de la señal.
  - Sigue por donde lo dejó.

- uid

19

## Ficheros (I)

- Para almacenar información
- Se agrupan en directorios
- Sistema de ficheros:
  - Estructura jerarquizada (ficheros y directorios)



- Ruta absoluta y ruta relativa (al directorio de trabajo)
- Mecanismo de protección

20

FICHEROS: sirven para almacenar información. Se agrupan en directorios para tener una estructura jerárquica. La estructura siempre empieza por el directorio raíz (/). Tenemos rutas absolutas (empiezan en el directorio raíz) y rutas relativas (al directorio de trabajo en el que se está en ese momento).

. => directorio actual

.. => directorio padre.

Mecanismos de protección: cada sistema tiene su propio sistema. En unix se utilizan 9 bits en los que se almacenan los permisos. Hay tres para usuario, tres para grupo y tres para acceso público, y se indica si se tiene permisos de escritura, lectura y/o ejecución.

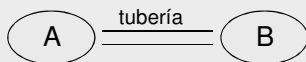
Hay varios tipos de ficheros: de bloques y de caracteres. Se usan para representar dispositivos de bloques o caracteres.

Cuando un proceso arranca tiene una tabla con tres entradas: stdin, stdout y stderr. El descriptor de fichero es el índice de la tabla de ficheros abiertos. El manejador es la entrada en la tabla y el descriptor el numerito.

Tuberías (pipes): sirven para comunicar procesos. Todo lo que escriba el proceso A en la tubería puede ser leído por el proceso B. Desde el punto de vista del proceso la tubería se va a manejar como si fuera un fichero. (Se verá en prácticas).

## Ficheros (y II)

- Ficheros especiales
  - De bloques: para representar dispositivos de bloques
  - De caracteres: para representar dispositivos de caracteres
- Tabla de ficheros abiertos del proceso
  - Descriptores de ficheros
    - 0: entrada estándar (stdin)
    - 1: salida estándar (stdout)
    - 2: salida estándar de errores (stderr)
- Tuberías (pipes)



21

## Intérprete de comandos (shell)

- No es parte del S.O
- Constituye la interfaz entre el S.O y el usuario
- Un shell se arranca por cada usuario que entra en el sistema
- Por cada orden, el shell crea un proceso hijo
- Procesos en background : &
- Redirección de E/S y pipes:
  - `date > fichero`
  - `sort < f1 > f2`
  - `cat f1 f2 f3 | sort > f4`



22

SHELL: constituye una interfaz entre el S.O y el usuario. El código que se ejecuta forma parte del núcleo del sistema operativo, pero no se puede utilizar directamente, se necesita el intérprete de ordenes, que es lo que realiza las llamadas al sistema para utilizar los recursos del sistema operativo.

Por cada usuario del sistema se crea un SHELL. Ahí es donde se aceptan las órdenes. Por cada orden el SHELL crea un proceso hijo, que es el que se encarga de realizar las operaciones. Cuando termina, el SHELL vuelve a mostrar el prompt para solicitar nuevas órdenes.

FOREGROUND: cuando el SHELL espera a que el proceso termine.

BACKGROUND: el proceso se ejecuta pero el SHELL sigue pidiendo nuevas órdenes. En bash se activa utilizando &.

El SHELL nos permite además redirigir la entrada salida (ver ejemplos).

Un sistema operativo es muy grande, por lo que se divide en módulos. Cada módulo tiene su función y su interfaz para comunicarse.

Lo más importante:

- Administrador de procesos.
- Administrador de memoria principal.
- Administrador de entrada salida.
- Administrador de archivos.

Son los bloques que se irán viendo durante el curso.

## Componentes y servicios de un sistema operativo

- Dividido en módulos, cada uno con una función y una interfaz concreta
- Los programas de usuario se comunican con el S.O y requieren sus servicios mediante llamadas al sistema
- Los servicios del S.O son los suministrados por los distintos componentes

23

## Componentes y servicios de un sistema operativo

- Administrador de procesos
  - Crear y eliminar procesos
  - Suspende y reanuda la ejecución de los procesos
  - Proporcionar mecanismos:
    - de sincronización
    - para la comunicación
    - para el manejo de bloqueos
- Administrador de la memoria principal
  - Controlar las zonas de memoria utilizadas y quién las utiliza
  - Decidir qué procesos se cargarán en memoria si queda espacio disponible
  - Asignar y recuperar espacio

24

Administrador de procesos: ver diapositiva.

manejo de bloqueos: un bloqueo se produce cuando un proceso espera a otro y el otro espera al primero. Algunos S.O gestionan los bloqueos pero otros no.

Administrador de memoria principal: ver diapositiva. asignar y recuperar espacio (malloc y calloc en C).

Administrador del sistema de entrada salida: lo conocido como drivers de dispositivos. Presentan una interfaz general con los manejadores. Cada hardware es diferente, pero para las capas superiores es como si fuera todo igual, es una interfaz común.

Administrador de archivos: gestiona el espacio en disco (qué esta ocupado y qué esta libre). (ver diapositiva).

## Componentes y servicios de un sistema operativo

- Administrador del sistema de E/S
  - Manejadores para dispositivos hardware específico
  - Presentar una interfaz general con los manejadores
- Administrador de archivos
  - Gestión del espacio en disco
  - Gestión de ficheros (crear, borrar y L/E)
  - Gestión de directorios (crear, borrar y L/E)
  - Correspondencia entre archivos y almacenamiento secundario
  - Gestión de copias de seguridad

25

## Componentes y servicios de un sistema operativo

- Sistema de protección
  - Controlar el acceso a los recursos
    - No permitir que usuarios no autorizados accedan a los recursos
- Sistema de comunicación
  - Gestionar los accesos a la red
  - Conexión a través de la red de los distintos procesadores
  - Acceso a los recursos compartidos

26

Sistema de protección: controlan el acceso a los recursos para que solo puedan acceder los usuarios autorizados.

Sistema de comunicación: gestión del acceso a la red.

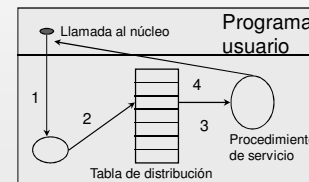
## Estructuras de Sistemas Operativos

- Sistemas monolíticos
- Sistemas en niveles
- Máquinas virtuales
- Modelo cliente-servidor

27

## Sistemas Monolíticos

- Organización más común
- Conjunto de procedimientos
- Modo de funcionamiento



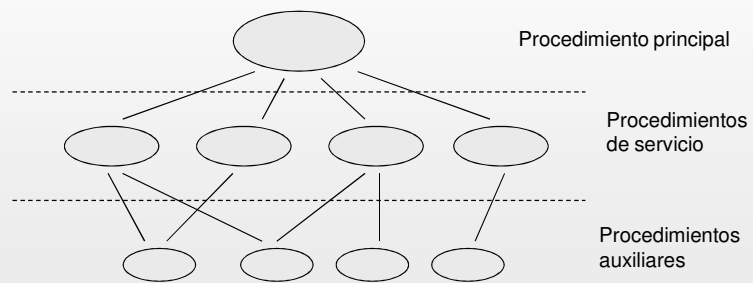
- (1)
  - Se dejan los parámetros de la llamada en lugares previamente definidos (ej: registros, pila)
  - Se ejecuta la inst. TRAP (llamada al núcleo o llamada al supervisor)
    - Cambia la máquina de modo usuario a modo privilegiado
    - Transfiere el control al S.O.
- (2)
  - El S.O. determina el nº del servicio requerido, examinando los parámetros
- (3)
  - El S.O. lee de la posición k un puntero al procedimiento que implementa la llamada y lo llama
- (4)
  - Devuelve el control al programa de usuario

28

Sistemas Monolíticos: es la organización de S.O más común. El S.O es un conjunto de procedimientos que se llaman entre sí según va siendo necesario.

## Sistemas monolíticos

- Arquitectura básica



29

## Sistemas en Niveles

- Jerarquía de niveles
- S.O. The (Dijkstra 1968)

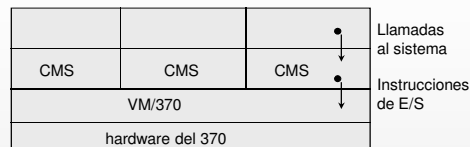
5	El operador
4	Programas de usuario
3	Gestión de Entrada/Salida
2	Comunicación operador-procesos
1	Gestión de la memoria principal y secundaria
0	asignación del procesador y multiprogramación

30



## Máquinas virtuales

- Se separan:
  - Multiprogramación
  - Máquina virtual
- VM (Virtual Machine)
  - Multiprogramación
  - Se ejecuta directamente sobre el hardware
  - Ofrece varias máquinas virtuales
- CMS (conversational Monitor System)
  - Cada máquina virtual puede tener cualquier S.O
  - Tiene instrucciones normales para operaciones de E/S
- No es sencillo de simular



31

## Virtualización

- Redescubrimiento de las máquinas virtuales
- Las máquinas virtuales actuales ejecutan una gran variedad de S.O en la misma máquina y sobre muchos S.O diferentes
  - Ej: Linux en Windows o Windows en Linux
- Ej: VirtualBox o VMWare
- Ventajas de la virtualización:
  - Ahorro energético
  - Ejecutar software de distintos S.O
  - Prueba de software
  - Instalación del software

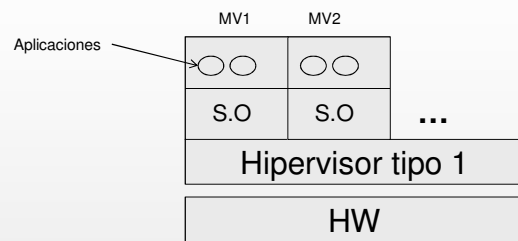
32



Hipervisor tipo 1: vmware lo vende, no es el gratuito. HW tiene que estar preparado para avisar al hipervisor tipo 1 de las interrupciones que van para los distintos sistemas operativos virtuales. No hay S.O host. S.O esta acostumbrado a tener privilegios sobre el hardware pero está en medio el hipervisor tipo 1, y eso lo tiene que soportar el hardware.

Hipervisor tipo 2: el normal y corriente de vmware.

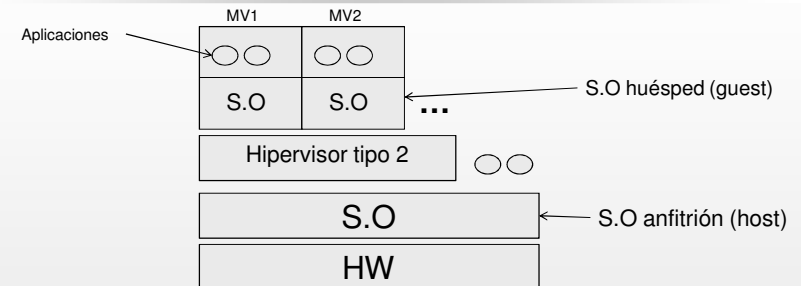
### Esquemas de virtualización: Hipervisor tipo 1



- El S.O se ejecuta como proceso de usuario
- El Hw tiene que estar preparado

33

### Esquemas de virtualización: Hipervisor tipo 2



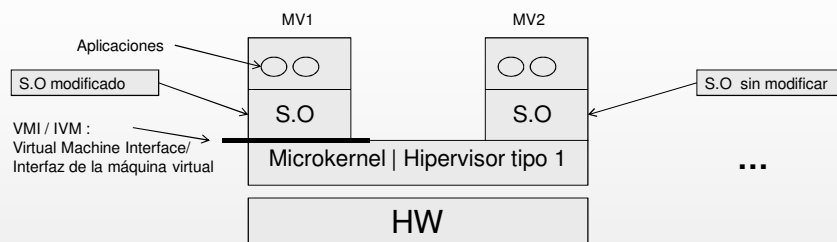
- El hipervisor traduce las instrucciones del S.O huésped a llamadas al hipervisor de tipo 2
- El hipervisor de tipo 2 hace las llamadas al S.O anfitrión

34

Paravirtualización: las instrucciones del so huésped son pasadas a un sistema operativo diseñado expresamente para esto.

JVM: el compilador de Java pasa el código a bytecode, que son instrucciones sobre la máquina virtual de Java. Después un intérprete las convierte a instrucciones de la máquina real.

## Esquemas de virtualización: Paravirtualización

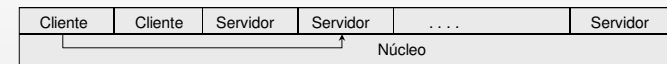


- El S.O. huésped se modifica para que llame a funciones de la MV en vez de ejecutar instrucciones privilegiadas
  - El S.O. huésped tiene que ser de código abierto
- Es más eficiente: No hay que producir la interrupción que atrapa el hipervisor.

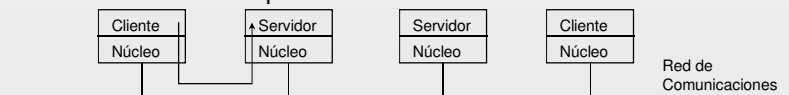
35

## Modelo Cliente-Servidor

- Núcleo mínimo:
  - Gestión de las comunicaciones entre clientes y servidores
  - Acceso directo al hardware



- Servidor:
  - Proceso que espera peticiones de los clientes
- Es sencilla la adaptación a sistemas distribuidos



36

Núcleo mínimo => microkernel. Se encarga de las labores básicas del sistema operativo: manejar peticiones, comunicaciones entre procesos, etc. Encima del núcleo mínimo tenemos clientes y servidores.

## Funcionamiento del modelo C/S

- El proceso servidor comienza
- Se va a dormir en espera de una petición
- El proceso cliente comienza
- El cliente envía una petición
- El servidor procesa la petición
- El servidor se va de nuevo a dormir
- Ej de servicios que proporcionan los servidores:
  - Hora del día
  - Imprimir un fichero
  - Leer o escribir en su sistema de ficheros
  - Ejecutar una orden para el cliente en el sistema del servidor
  - Dibujar algo en la pantalla servida por el servidor

37

## Tipos de servidores

- Iterativos
  - El servidor maneja la petición
  - La petición del cliente puede ser atendida por el servidor en un tiempo conocido y corto
  - Ej: Servidor de la hora
- Concurrentes
  - El servidor invoca a otro proceso que da el servicio mientras el servidor duerme esperando otro cliente
  - Arranca un nuevo proceso por cada cliente que llega
  - No se sabe el tiempo que llevará servir la petición, ya que la cantidad de tiempo requerido, depende de la petición.
  - Ej: Servidores de fichero: el tiempo requerido para leer o escribir en fichero depende de la cantidad de información a L/E.

38