



# Sistemas Operativos

## Tema 1

### Arquitectura de computadores y Sistemas Operativos



# Arquitectura y S.O.

- La computadora es una máquina destinada a procesar datos mediante una serie de instrucciones máquina
- Los componentes básicos de un computador son:
  - **Procesador:**
    - Controla el funcionamiento del computador
    - Realiza el proceso de datos
    - Cuando solo hay uno se suele llamar CPU (Central Processing Unit)
  - **Memoria Principal:**
    - Almacena datos. Almacena programas
    - Habitualmente es volátil.
    - Se llama también memoria real o memoria física
    - Almacena el “programa” en ejecución en código máquina y sus resultados
  - **Módulos de E/S:**
    - Transfieren datos entre computador y su entorno externo.
    - El entorno serían los dispositivos a los que se puede conectar
    - Dispositivos de memoria secundaria (discos), equipos de comunicación, terminales

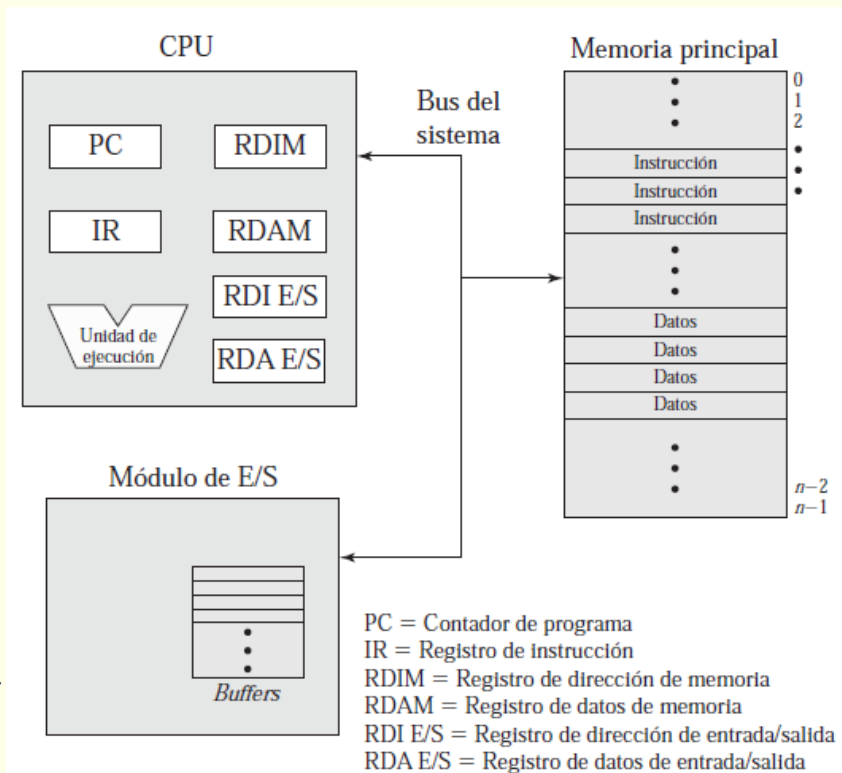


Figura 1.1. Componentes de un computador: visión al más alto nivel.

Se denomina programa máquina al conjunto de instrucciones máquina que tiene por objeto que la computadora realice una determinada función.

# Arquitectura y S.O.

## ➤ Registros del procesador

- Visibles para el usuario
  - Permiten reducir el acceso a memoria principal
  - Se acceden desde el código máquina vía instrucciones
  - Algunos lenguajes como C permiten sugerir al compilador qué variables almacenar en registros. (Aceleran el código)  
(*register*)
- De control y estado
  - Usados por el procesador para controlar su operación
  - Usado por rutinas privilegiadas del sistema operativo para controlar la ejecución de programas

La clasificación en estas categorías no está nítidamente definida, depende de la arquitectura. Por ejemplo contador de programa a veces visible.

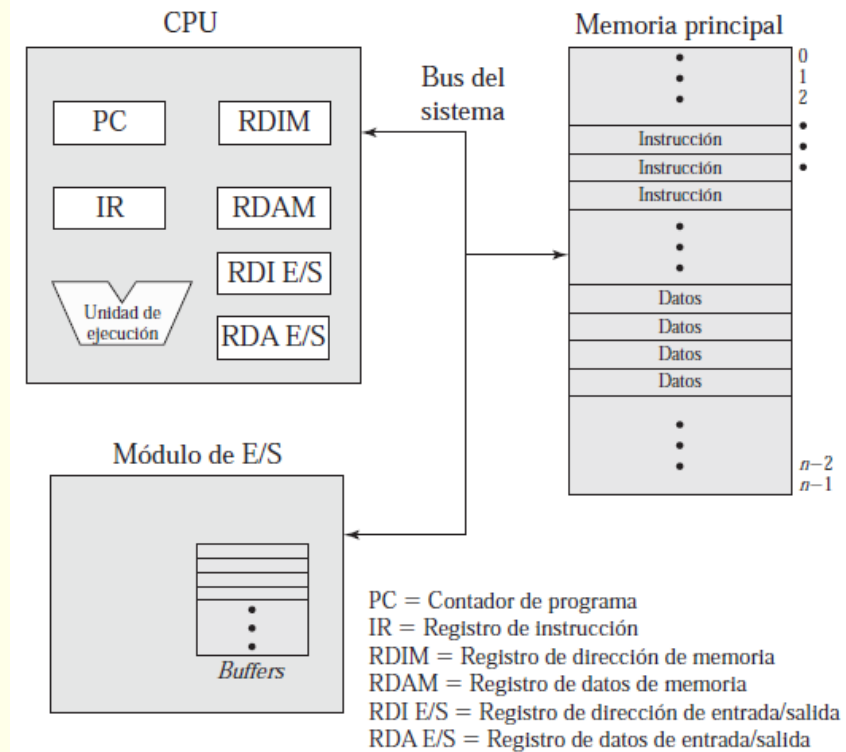
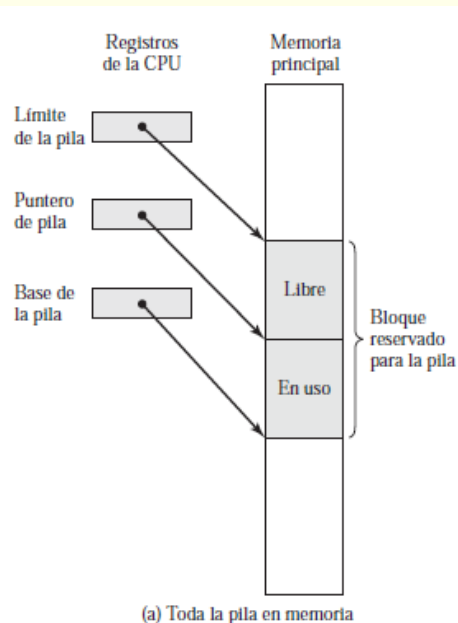


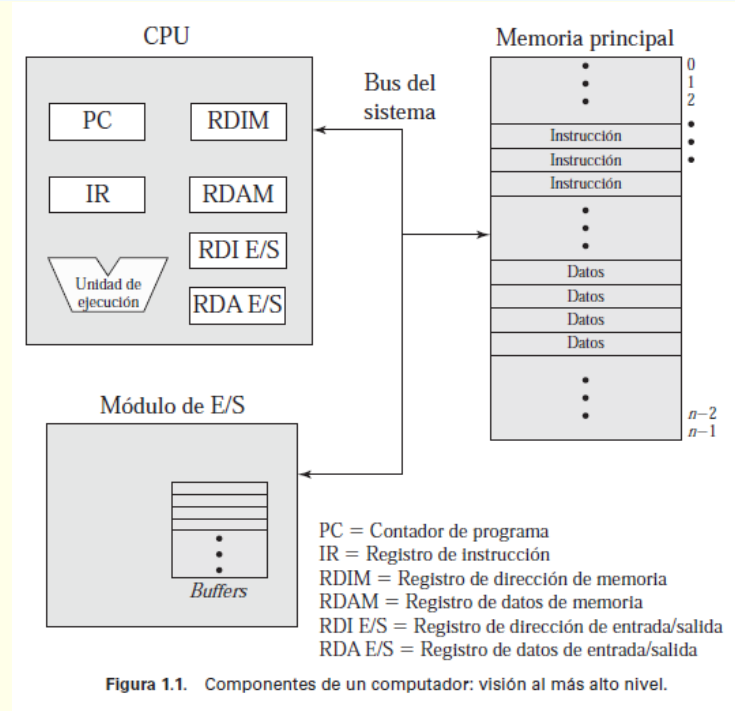
Figura 1.1. Componentes de un computador: visión al más alto nivel.

# Arquitectura y S.O.

- Registros Visibles (datos, dirección y códigos de condición)
  - Registros de datos
    - Diversos registros de propósito general normalmente accesibles por cualquier instrucción.
    - Puede haberlos dedicados (coma flotantes, específicos op. enteras)
  - Registros de dirección:
    - Contienen direcciones de datos o instrucciones
    - La dirección puede ser efectiva (la real) o virtual (sirve para calcular la efectiva)



- Registro índice: Para el direccionamiento indexado. Se suma un índice a un valor base para obtener la dirección
- Puntero de segmento: Una referencia a memoria segmentada almacena en él el comienzo del segmento. Puede haber varios, uno para el S.O. y otro para la aplicación en ejecución.
- Puntero de pila: Apunta a la dirección cima de la pila. Permite usar instrucciones sin dirección, cogen el dato de la pila. Operaciones específicas *push* y *pop*.



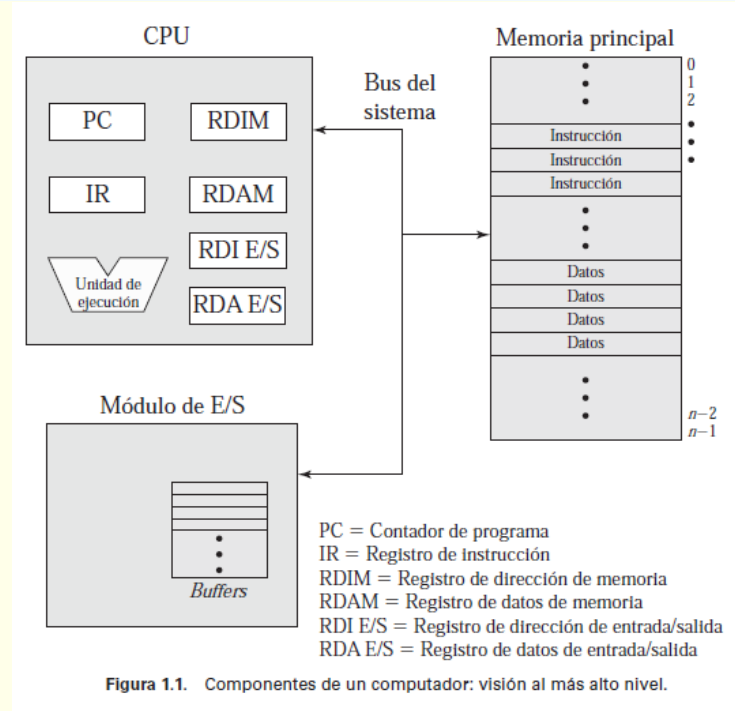
La pila se almacena en memoria principal  
Es un conjunto secuencial de posiciones  
Se acceden en modo LIFO (Last In First Out)  
Se accede siempre a la cima de la pila

# Arquitectura y S.O.

## ➤ Registros de control y estado

Diferente terminología entre máquinas.

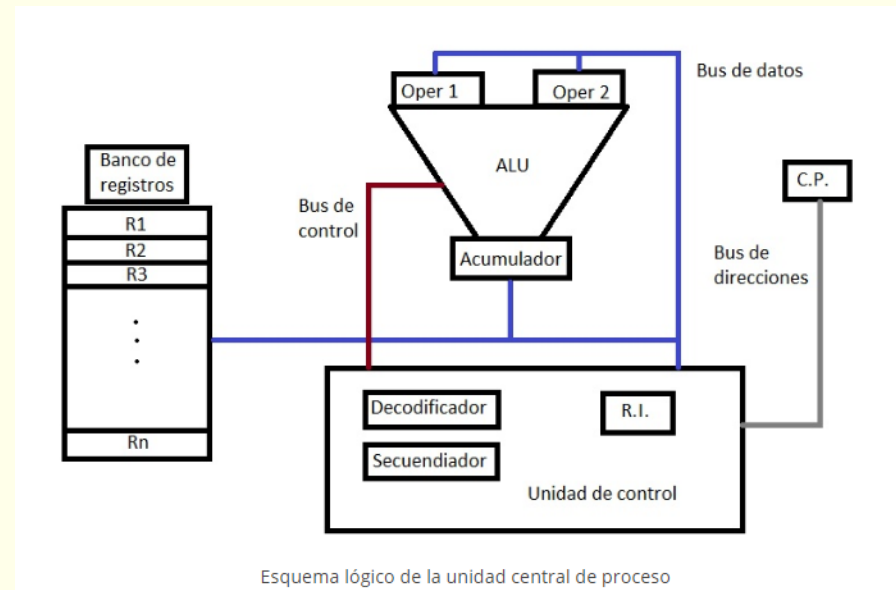
- RDIM: Registro de dirección de memoria:  
Contiene la dirección de memoria de la siguiente lectura o escritura.
- RDAM: Registro de datos de memoria:  
Contiene los datos que se van a escribir en la memoria.  
Contiene los datos que se han leído de memoria
- RDI/ES: Registro de dirección de entrada salida:  
Contiene la dirección de memoria asignada al dispositivo E/S
- RDA/ES: Registro de datos de E/S  
Contiene los datos que se van a escribir en E/S  
Contiene los datos que se han leído de E/S
- PC: Contador de programa  
Contiene la dirección de la próxima instrucción máquina que se leerá de memoria.
- RI: Registro de instrucción  
Contiene la última instrucción máquina leída
- PSW: Program Status Word  
Un registro o conjunto de registros  
Contiene información de estado de la CPU, indicadores, bit para habilitar/deshabilitar interrupciones, bit modo usuario/superusuario  
Los indicadores normalmente los edita el hardware como resultado de las operaciones: positivo/negativo, cero, desbordamiento, etc..  
Los indicadores pueden ser leídos por las instrucciones máquina
- Punteros de pila: Base, cima, límite.
- Registros de control de E/S: dedicados a operaciones de E/S



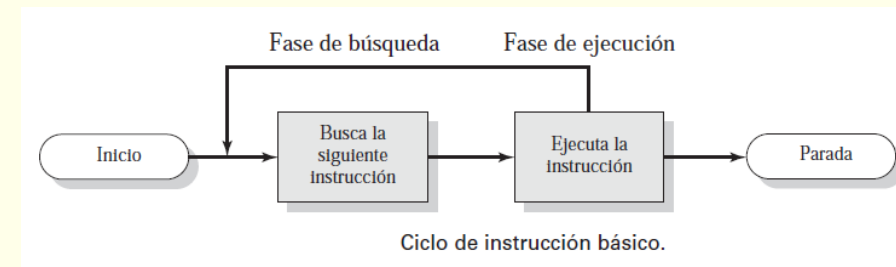
En máquinas con múltiples interrupciones hay varios registros de interrupción con un puntero a la rutina de tratamiento.

# Arquitectura y S.O.

- La unidad aritmético-lógica (ALU) se encarga de realizar una serie de operaciones aritméticas y lógicas sobre uno o dos operandos, controlada por la unidad de control vía el bus de control
- La unidad de control se encarga de:
  - Leer de memoria las instrucciones máquina que forman el programa a ejecutar
  - Interpretar cada instrucción
  - Leer los datos de memoria referenciados por cada instrucción
  - Ejecutar cada instrucción
  - Almacenar el resultado de cada instrucción.
- Ejecución de instrucciones
  - Se ejecutan en el ciclo de instrucción
  - El ciclo de instrucción es atómico
  - El programa se puede interrumpir entre instrucciones.
  - El PC se incrementa automáticamente tras cada instrucción
  - La instrucción se carga en el R.I. (Registro de instrucción)
  - El procesador interpreta la instrucción (bits del R.I.)
  - El procesador la ejecuta
  - Algunas instrucciones tienen más fases, más direcciones, (Juego de instrucciones más completo)
  - Instrucciones, categorías: Procesador-memoria, Procesador-E/S, Procesamiento de Datos, Control



Esquema lógico de la unidad central de proceso

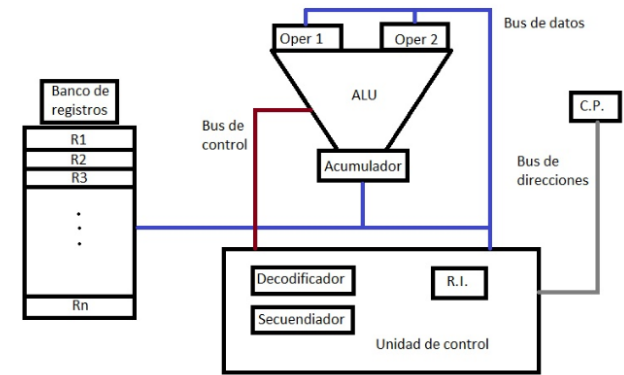


Ciclo de instrucción básico.

# Arquitectura y S.O.

## Ejemplo ordenador de 16 bits

- El procesador contiene un único registro de datos (AC) Acumulador
- Tanto instrucciones como datos tiene un ancho de 16 bits.
- La memoria está organizada en palabras de 16 bits (2bytes)
- El formato de instrucción proporciona:
  - 4 bits para código de operación  $\Rightarrow 2^4=16$  opcodes
  - Los opcodes se representan con un dígito hexadecimal
  - 12 bits para la dirección.
  - Con 12 bits de dirección se pueden direccionar  $1^{12}=4\ 096$  palabras =4K palabras



Esquema lógico de la unidad central de proceso



(a) Formato de Instrucción



(b) Formato de un entero

Contador de programa (PC) = Dirección de la Instrucción  
Registro de instrucción (IR) = Instrucción que se está ejecutando  
Acumulador (AC) = Almacenamiento temporal

(c) Registros internos de la CPU

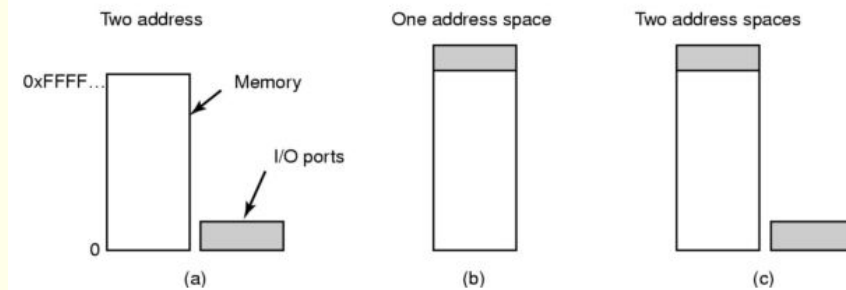
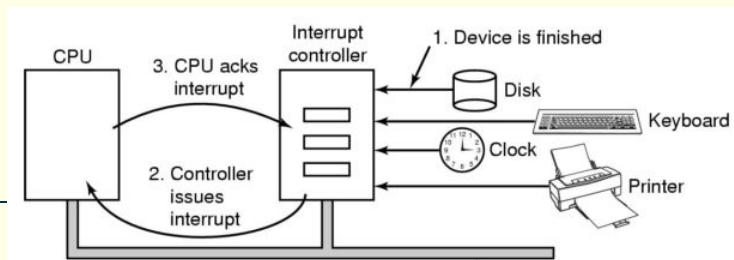
0001 = Carga AC desde la memoria  
0010 = Almacena AC en memoria  
0101 = Suma a AC de la memoria

(d) Lista parcial de códigos-de-op

# Arquitectura y S.O.

## SISTEMA DE ENTRADA/SALIDA

- El procesador puede acceder directamente a leer/escribir de un módulo de E/S
  - El módulo y sus dispositivos puede estar mapeado en memoria
  - Identifica el dispositivo fijando el RDI/ES
  - Las instrucciones (dirección) hacen referencia a la memoria del módulo
  - El módulo puede tener permiso para acceder a la memoria principal.
  - Se emiten ordenes (CPU o Módulo E/S) de acceso a memoria (lectura/escritura) que son atendidas por el DMA (Direct Memory Access)
- Tipos de operaciones de E/S
  - Programada
  - Controlada por Interrupciones
  - Mediante DMA



- a) Espacios separados para Puertos de E/S y memoria
- b) Espacio único para memoria y Puertos de E/S
- c) Modelo mixto

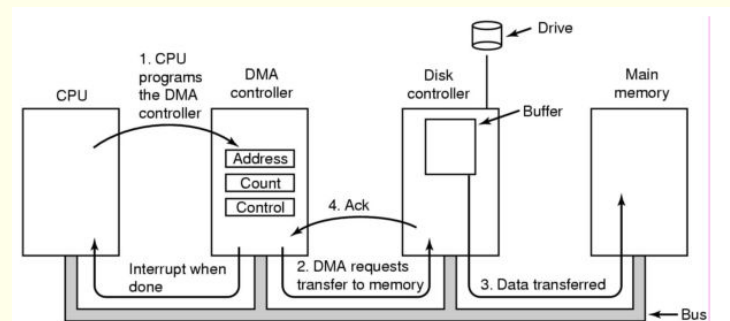
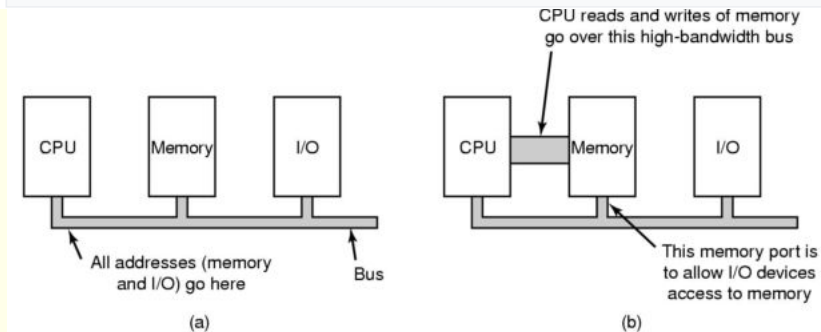
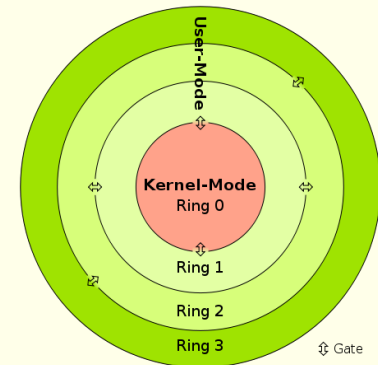


Gráfico que muestra la operativa relacionada con una transferencia de E/S controlada por DMA



# Arquitectura y S.O.

- El modelo de programación de una computadora se caracteriza por:
  - Los elementos de almacenamiento
  - El juego de instrucciones
  - La secuencia de funcionamiento
- La mayoría de las computadoras presentan dos niveles de ejecución:
  - Nivel de usuario (una o varias capas):
    - La computadora solo ejecuta un subconjunto de las instrucciones máquina quedando prohibidas el resto.
  - Nivel de núcleo:
    - La computadora ejecuta todas sus instrucciones sin ninguna restricción.
- Los tres mecanismos que permiten romper la secuencia de ejecución de una computadora:
  - Las instrucciones de salto.
  - Las interrupciones internas o externas
  - La instrucción TRAP.



# Arquitectura y S.O.

## INTERRUPCIONES

- Mecanismo por el que se puede interrumpir la secuencia de ejecución del procesador.
- Permiten mejorar la utilización del procesador
  - Sincronizan dispositivos (más lentos que) con el procesador
- Clases de interrupciones
  - De programa: Desbordamiento, división por cero, acceso ilegal (instrucción o memoria), Trap
  - Por temporizador: Generada por el temporizador del procesador. Permite al S.O. tomar el control tras tiempo cedido al proceso
  - De E/S: Generada por el controlador de E/S para indicar: fin de operación, condición de error, etc.
  - Por fallo del Hardware: Fallo en un dispositivo, error de paridad, acceso a memoria, etc.
- El procesador continúa mientras no se le interrumpa
- El ciclo de aceptación de una interrupción incluye:
  - Salvar algunos registros de la computadora.
  - Pasar al procesador a modo núcleo.
  - Cargar un nuevo valor en el contador de programa para pasar a ejecutar otro programa (Manejador de interrupciones)

Computador a 1GHz  
 $10^9$  instrucciones/segundo  
Disco duro a 7200 rpm  
rotación de media pista 4ms  
4 millones de veces más lento

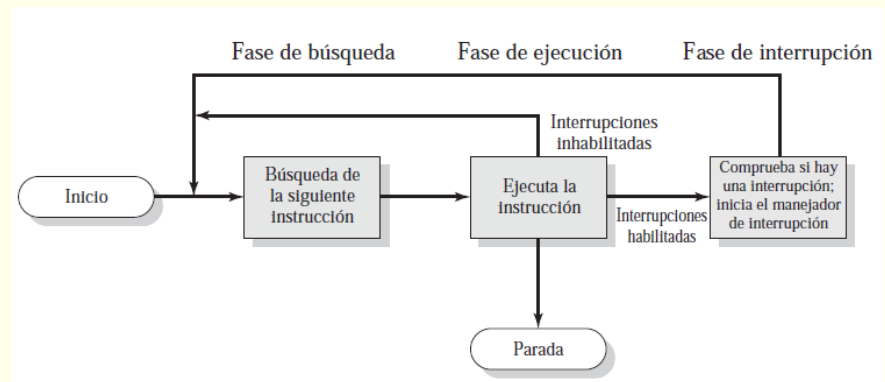


Figura 1.7. Ciclo de instrucción con interrupciones.

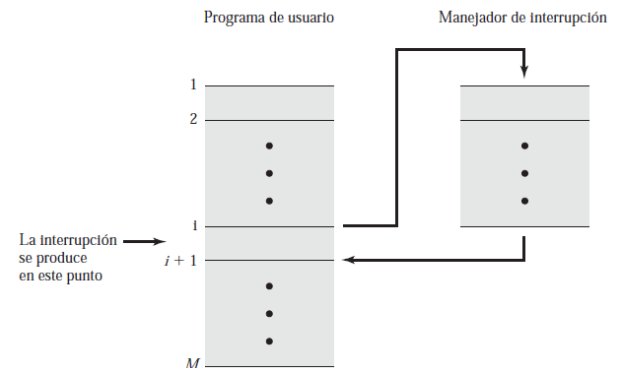


Figura 1.6. Transferencia de control mediante interrupciones.

# Arquitectura y S.O.

- Memoria => ¿Capacidad? ¿Velocidad? ¿Coste? => compromiso
- Solución: La memoria se estructura como una jerarquía que utiliza:
  - Memorias permanentes de alta capacidad y baja velocidad (discos) para almacenamiento permanente de la información
  - Memorias de semiconductores de tamaño reducido y alta velocidad para almacenar la información que se está utilizando en un momento determinado.
- Los parámetros característicos de una jerarquía de memoria son:
  - La tasa de aciertos: Define la probabilidad de encontrar en un nivel de la jerarquía la información referenciada
  - El tiempo medio de acceso efectivo: Mide el tiempo medio de acceso a la información

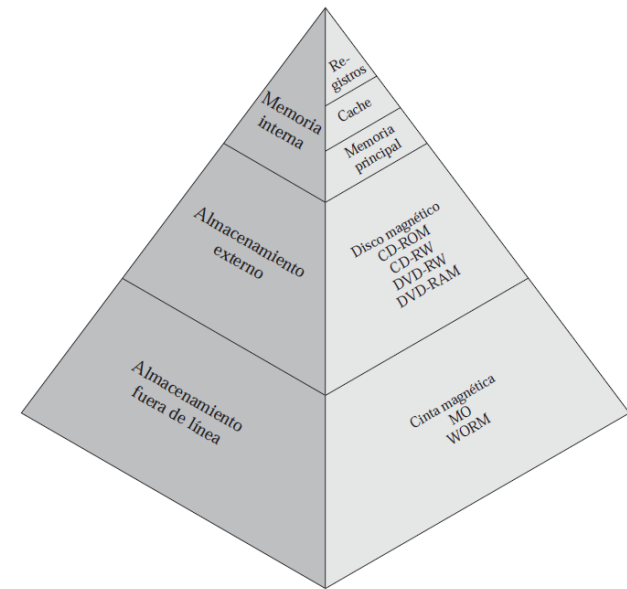


Figura 1.14. La jerarquía de memoria.

# Arquitectura y S.O.

## Memoria Cache

### ➤ Motivación

- En cada ciclo de instrucción el procesador accede a memoria una o más veces
- Idealmente el procesador no debería esperar al dato (lectura/escritura), memoria igual de rápida que los registros de CPU
- Aprovechando el principio de proximidad se utiliza una memoria mucho más rápida (pero más pequeña) entre procesador y memoria.

- La cache contiene copia de parte de la memoria principal
- El procesador comprueba primero si el byte a leer está en la cache
- Si no está se lee de memoria un bloque y se lleva a la cache
- Memoria principal de  $2^n$  palabras en bloques de longitud fija de  $K$  palabras
- Hay  $M = 2^n/K$  bloques en memoria
- La cache tiene  $C$  huecos (líneas) de  $K$  palabras cada uno.
- El número de líneas de cache es mucho menor ( $C \ll K$ )
- Cada línea tiene el identificador de bloque almacenado en ella
- Existe un algoritmo de reemplazo de línea. LRU (ayuda del hardware)
- Política de escritura de la línea escrita. En el acto, al reemplazo.

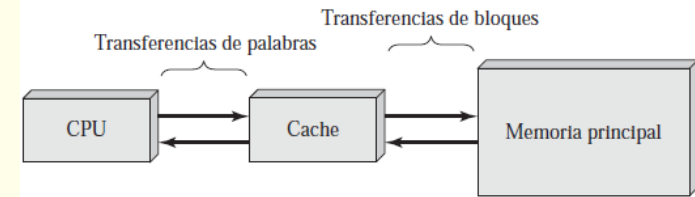


Figura 1.16. Cache y memoria principal.

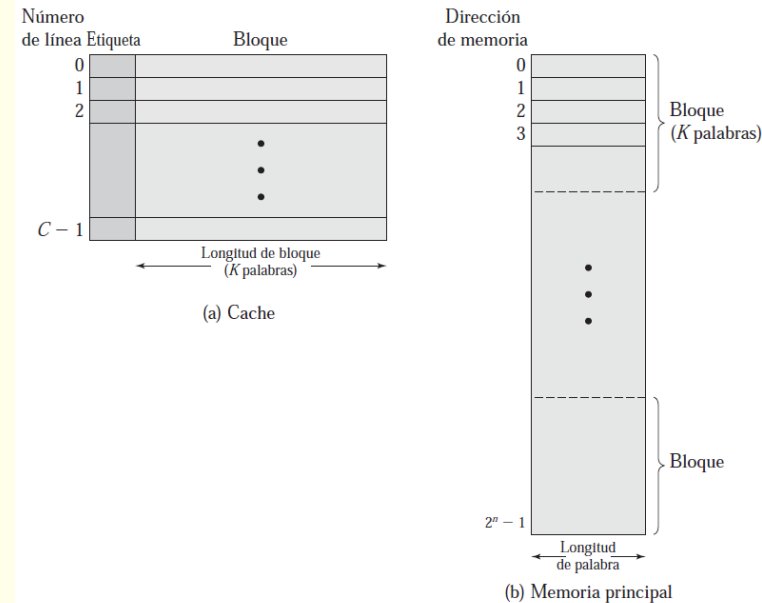
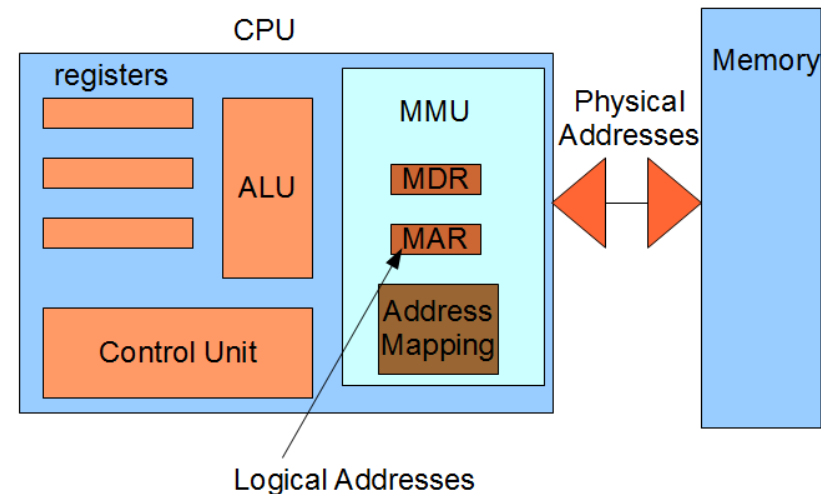
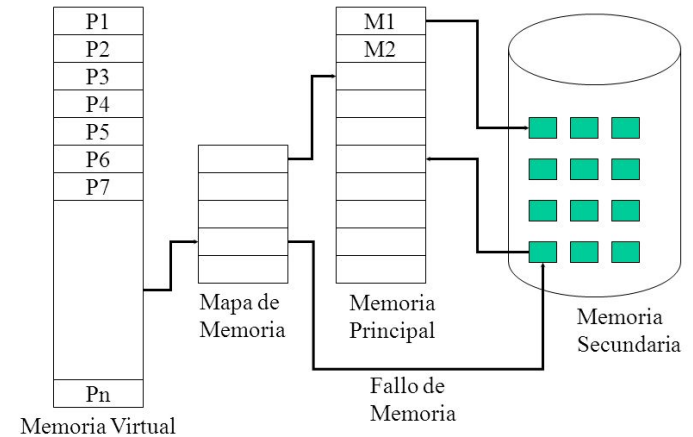


Figura 1.17. Estructura de cache/memoria principal.

# Arquitectura y S.O.

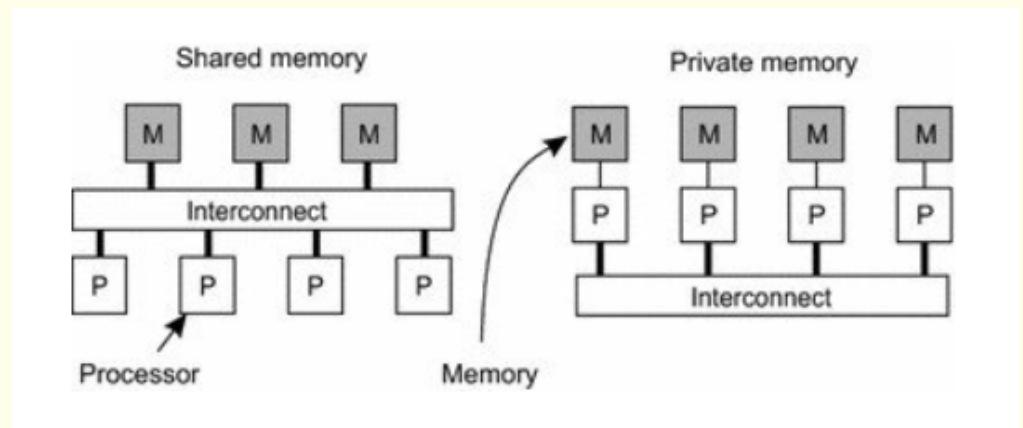
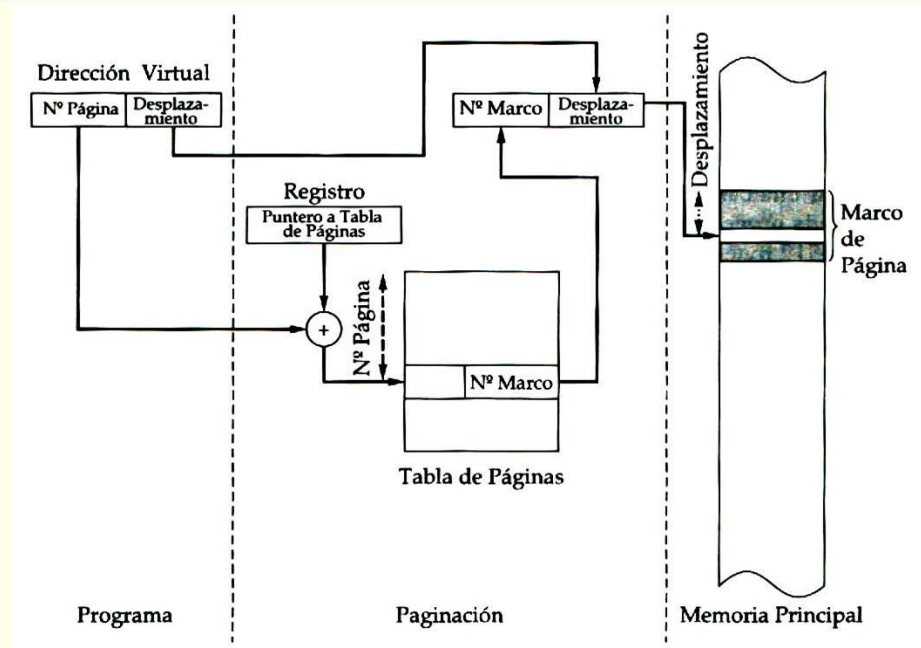
- Utilizando memoria virtual, las direcciones generadas por el procesador se refieren a un espacio de direcciones virtual que puede residir en memoria principal o en disco. La memoria virtual exige una gestión automática de la parte de la jerarquía de memoria formada por la memoria principal y el disco.
- La MMU (Memory Management Unit) se encarga de traducir las direcciones virtuales a direcciones físicas
- Un esquema de memoria virtual muy empleado es el de paginación. En él, el espacio de direcciones se divide en páginas y la memoria principal en marcos de página. En este esquema la MMU debe obtener el marco de página donde se encuentra la página referenciada.
- Registros de la MMU accedidos por la unidad de control.
  - MDR: Memory Data Register.  
Dato a enviar a o recibido de memoria.
  - MAR: Memory Address Register.  
Dirección lógica a la que se quiere acceder (el programa)
  - Utilizando un mapa de memoria y un esquema de direccionamiento la MMU traduce la dirección lógica en dirección física.

## Gestión de Memoria Virtual



# Arquitectura y S.O.

- La tabla de páginas es una estructura de información que almacena la información de donde residen las páginas de un programa en ejecución.
- Las páginas pueden residir en marcos de página de la memoria principal o en disco.
- Un multiprocesador es una máquina formada por un conjunto de procesadores que comparten una memoria principal común.
- Una multicomputadora es una máquina compuesta por varios nodos, con cada nodo compuesto por un procesador, su memoria principal y elementos de E/S.
  - Los distintos procesadores no comparten acceso a una memoria común y la comunicación entre ellos se realiza a través de una red de interconexión de alta velocidad.



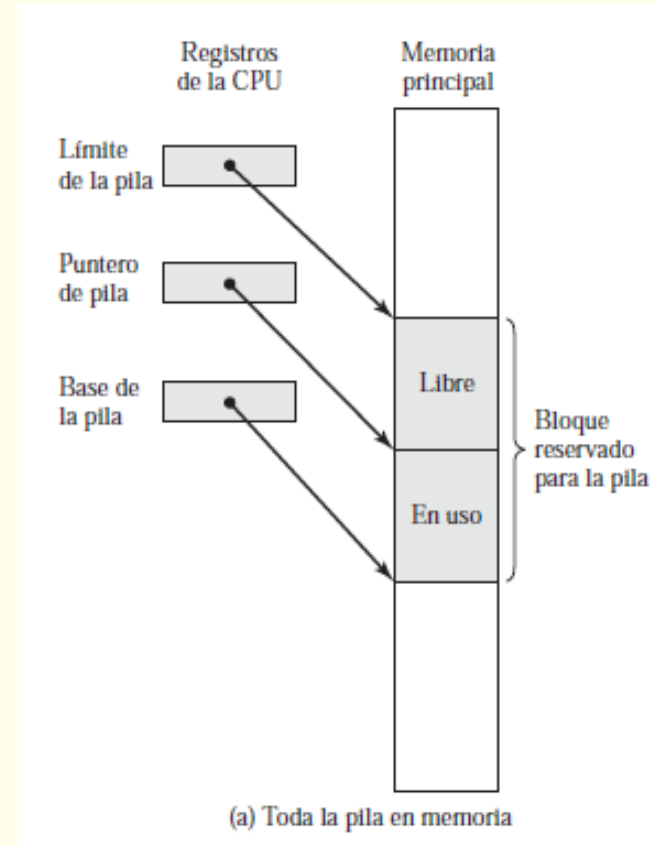
# Arquitectura y S.O.

## LA PILA

- Para controlar la ejecución de llamadas a procedimientos y retornos de los mismos

### Implementación

- Conjunto ordenado de elementos.
- Sólo se puede acceder a uno. Elemento en la **cima** de la pila.
- Acceso en orden LIFO (Last-In First-Out)
- Longitud de la pila es variable.
- La pila crece en orden inverso de memoria
- Se requieren tres registros en el procesador
  - Puntero de pila: Dirección a la cima de la pila.  
Si *push* se decrementa, si *pop* se incrementa.
  - Base de pila: Primera posición de una pila vacía.  
*Pop* sobre el da error.
  - Límite de pila: Dirección de primera dirección del bloque reservado a la pila.  
Si *push* da error.



# Arquitectura y S.O.

## LLAMADAS Y RETORNOS DE PROCEDIMIENTOS

- El procesador realiza una llamada a función.
  - Se almacena (apila, push) la dirección de retorno en la pila
- Se produce un retorno
  - Se utiliza la cima de la pila se extrae (pop) de esa dirección

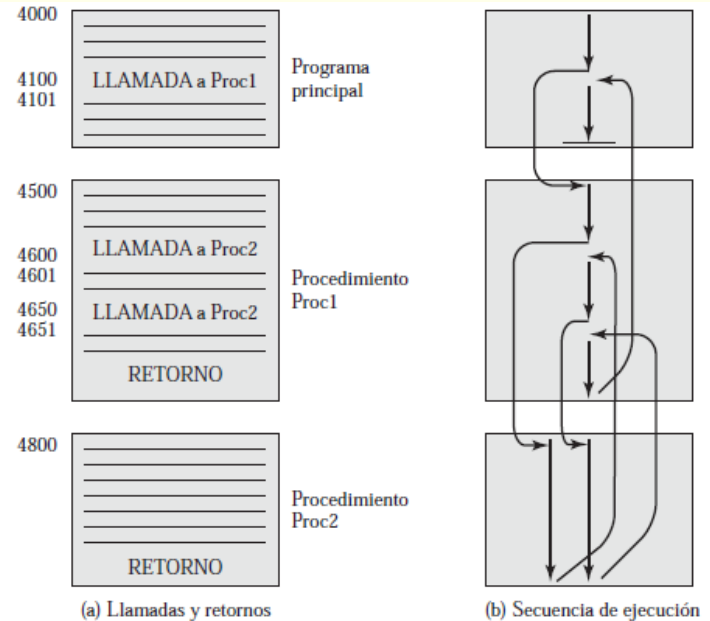
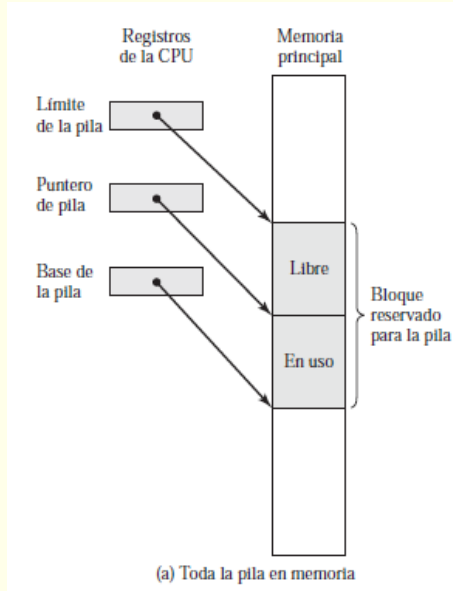


Figura 1.26. Procedimientos anidados.

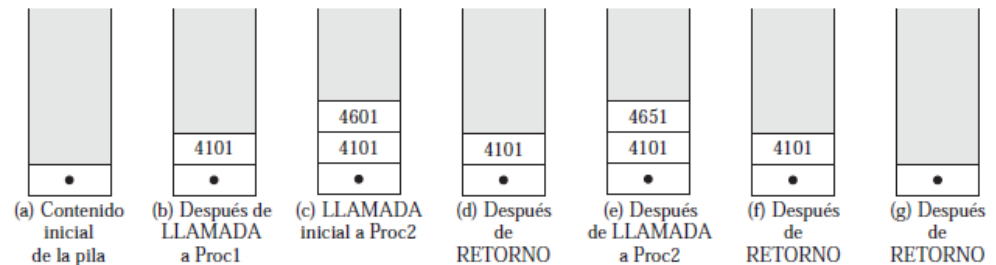


Figura 1.27. Uso de la pila para implementar los procedimientos anidados de la Figura 1.26.



# Arquitectura y S.O.

## LLAMADAS Y RETORNOS DE PROCEDIMIENTOS

- Cuando se pasan parámetros
  - Se apilan también los parámetros en orden
  - El procedimiento llamado los puede desapilar para acceder a ellos
  - Y apilara si procede los valores de retorno
- El conjunto de parámetros con la dirección de retorno se llama marco de pila
- Un procedimiento P declara variables locales  $x1$  y  $x2$
- Un procedimiento Q llamado por P declara variables  $y1$  e  $y2$   
( $y1$  es parámetro,  $y2$  var local)
- El primer elemento almacenado es un puntero al marco anterior  
(permite paso variable de parámetros)
- Se almacena la dirección de retorno de este marco (de esta llamada)
- Se reserva en la pila para las variables locales
- Se llama a Q(8) 8 es el valor que se pasan.  
(Aún con parámetros opcionales se reserva igualmente)

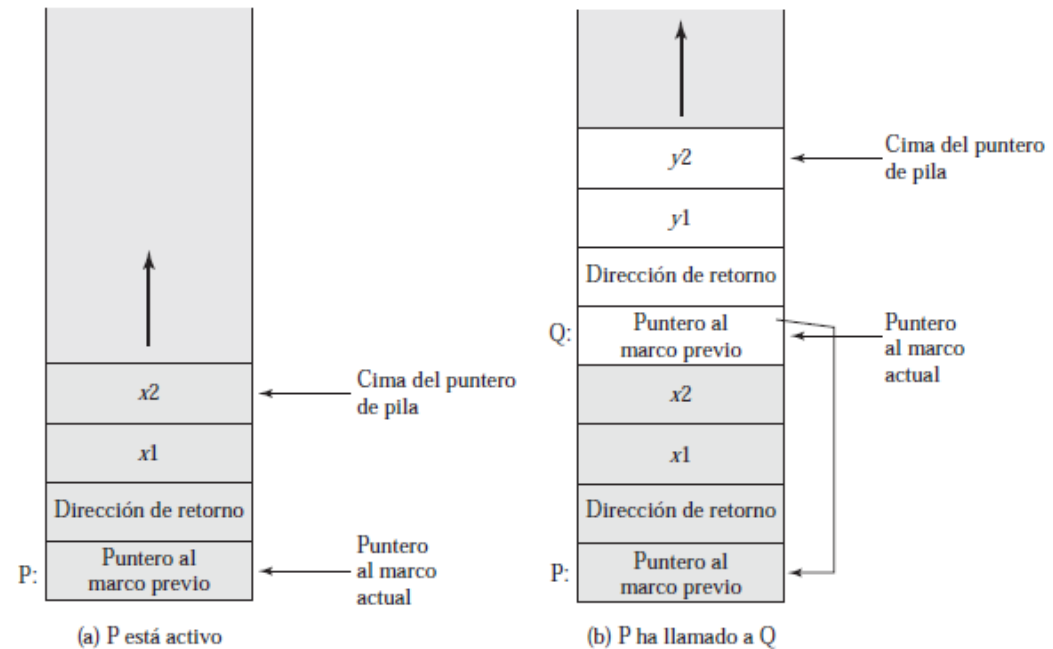


Figura 1.28. Crecimiento del marco de pila utilizando los procedimientos de ejemplo P y Q.

- Se crea un nuevo marco para Q
  - Puntero al previo, Dirección de retorno
  - Variables locales,
  - Como recibe parámetros, la local se inicializan con el valor del parámetro pasado ( $y1=8$ )
  - La otra variable es simplemente para uso local del procedimiento.
- Las variables locales se apilan para permitir la reentrada de procedimientos