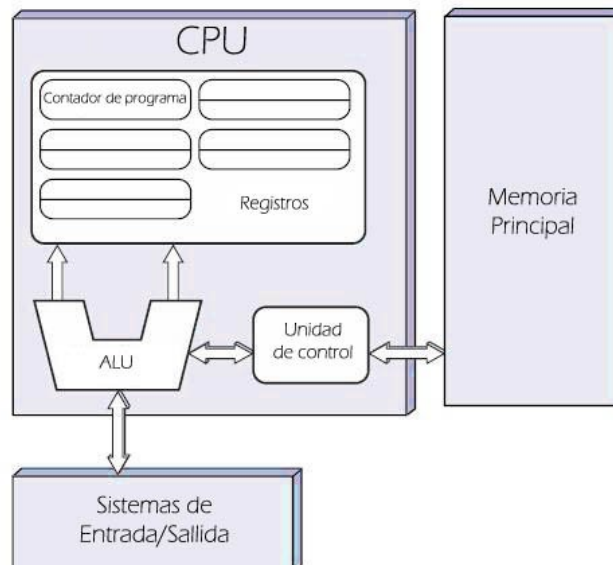
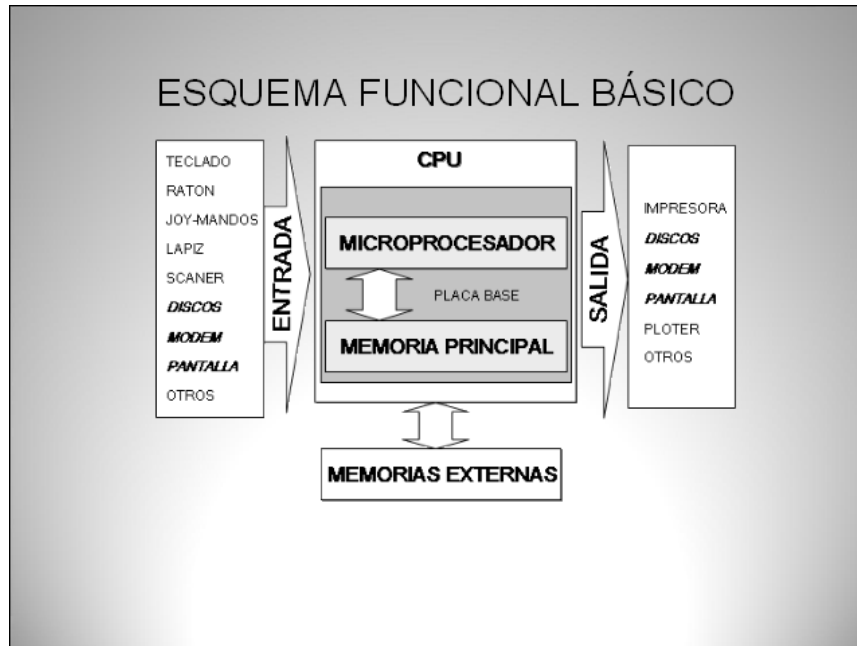


CUESTIONARIO DE REPASO

Daniel Alconchel Vázquez

1. Dibujar un esquema básico de los componentes de un computador y sus interconexiones: CPU, memoria y dispositivos.



2. Diferencia entre interrupciones y excepciones. ¿Cómo se gestionan?

Por una parte, una **interrupción** es una señal que envía un dispositivo de E/S a la CPU para indicar que la operación de la que se estaba ocupando ha terminado. Por otra parte, una **excepción** es una situación de error detectada por la CPU mientras ejecutaba una instrucción, que requiere tratamiento por parte del SO.

- **Tratamiento de las interrupciones:** Una interrupción se trata en todo caso, después de terminar la ejecución de la instrucción en curso. El tratamiento depende de cuál sea el dispositivo de E/S que ha causado la interrupción, ante la cual debe poder identificar el dispositivo que la ha causado.
- **Tratamiento de las excepciones:** Cuando la CPU intenta ejecutar una instrucción incorrectamente construida, la unidad de control lanza una excepción para permitir al SO ejecutar el tratamiento adecuado. Al contrario que en una interrupción, la instrucción en curso es abortada. Las excepciones al igual que las interrupciones deben estar identificadas.

3. En una arquitectura de modo dual de funcionamiento ¿cómo obtenemos un servicio del SO?

En la arquitectura de modo dual, las CPUs tienen al menos dos modos de funcionamiento:

- **Modo usuario:** El intento de ejecución de una excepción privilegiada en este modo produce una excepción.
- **Modo kernel o sistema:** Se puede ejecutar cualquier instrucción.

Para obtener un servicio del SO se realiza una llamada al sistema. Las llamadas al sistema se implementan a través de una trampa ("interrupción de software"). Esta llamada transfiere el control al modo kernel, donde el SO verifica si los parámetros son correctos y legales y, posteriormente, si así es, ejecuta la solicitud y devuelve el control a la siguiente instrucción. En caso contrario, devuelve un error.

4. Diferencia entre entradas/salidas programadas y e/s gestionadas por interrupción.

Por un lado, las **E/S programadas** se sincronizan a través de un programa, por lo que son más simples de implementar, pero presentan inconvenientes como la pérdida de tiempo (tiempo en el que el computador no realiza trabajo útil en el bucle de espera), impide la realización de tareas periódicas y presenta dificultades al atender a varios periféricos.

Por otro lado, las **E/S gestionadas por interrupción** permiten evitar el inconveniente de la pérdida de tiempo. Básicamente una interrupción viene determinada por la ocurrencia de una señal externa que provoca la bifurcación a una dirección específica de memoria, interrumpiendo momentáneamente la ejecución del programa. A partir de esa dirección se encuentra la rutina de tratamiento que se encarga de realizar la operación de E/S propiamente dicha, devolviendo después el control al punto interrumpido del programa.

5. ¿Cómo implementa un SO el concepto (abstracción) proceso?

Para implementar el modelo de procesos, el S.O. mantiene una tabla (un arreglo de estructura), llamada tabla de procesos, con una entrada por proceso. Cada entrada contiene información acerca del estado del proceso, su contador de programa, apuntador a la pila, asignación de memoria, estado de sus archivos abiertos, información contable y de calendarización entre otras cosas.

6. ¿Qué es la imagen de un proceso?

Estructura que contiene toda la información necesaria para que un proceso se ejecute y poniendo a disposición los recursos necesarios del ordenador.

7. ¿A qué estado pasa un proceso que solicita un servicio del SO por el que posiblemente debe esperar? ¿cómo alcanza dicho estado?

Pasa al estado de **preparado**, donde el proceso no está ejecutándose, pero es candidato a pasara estado activo. Es el planificador el que decide que proceso de la lista de procesos preparados pasa a estado activo.

El proceso alcanza dicho estado tras realizar una llamada al sistema.

8. ¿A qué estado(s) pasa el proceso que está actualmente ejecutandose cuando se produce una interrupción?

Puede pasar a estado **bloqueado**, donde el proceso está pendiente de un evento externo que le ha hecho bloquear, tales como una operación de lectura/escritura, la espera de finalización de un proceso hijo, una señal o una operación sobre un semáforo. El dispositivo/hecho externo "avisa" al S.O. cuando ha terminado la acción que realizaba mediante una INTERRUPCIÓN, dejando el S.O. lo que está haciendo para atender a esta última. Tras esto, el S.O. comprueba cuales son los procesos que fueron bloqueados por ese evento externo, cambiándolos al estado de preparado.

También puede que se encuentre en estado de **terminado**, donde el proceso realiza una llamada al sistema solicitando su terminación.

9. Los sistemas operativos actuales se construyen para que los procesos cooperen en la multiprogramación. ¿Cómo se hace para que cooperen?

La multiprogramación una técnica que intenta incrementar la utilización del CPU tratando de tener siempre algo para que el CPU lo ejecute. Esta idea funciona de la siguiente manera: El sistema operativo selecciona uno de los trabajos y empieza su ejecución. Eventualmente, el trabajo tiene que esperar por algo, tal como una lectura a disco, un comando que debe ser tecleado, etc. En un sistema no multiprogramado el CPU debe esperar si hacer nada. En un sistema con multiprogramación, el sistema operativo simplemente seleccionará otro trabajo y lo ejecutará.

Cuando ese trabajo necesite esperar, el CPU será asignado a otro trabajo y de esta forma continuará. Eventualmente el primer trabajo habrá terminado su espera y obtendrá el CPU nuevamente. De esta manera, si siempre existe algún trabajo para ejecutar, el CPU nunca estará ocioso.