

POSIX y sistemas en tiempo real

Sistemas en tiempo real

Un sistema de tiempo real se define como: “un sistema que debe responder ante estímulos generados por el entorno dentro de un periodo de tiempo finito especificado”.

Aspectos fundamentales

- Son sistemas que interactúan con el entorno (mundo físico), adquiriendo estímulos y estados del entorno y generando una acción sobre dicho entorno.
- Son sistemas en los que el tiempo de respuesta es un aspecto crítico.
- Todo sistema de tiempo real debe ser suficientemente rápido para cumplir los requerimientos, pero el que un sistema sea rápido no implica que éste sea de tiempo real.
- La rapidez de un sistema de tiempo real está fijada por la naturaleza de los estímulos con los que tiene que tratar, la cual puede variar de unas aplicaciones a otras.
- Tiempo real significa trabajar con aspectos del sistema como distribución física de los componentes, manejo de estímulos impredecibles, fallos en los componentes, incertidumbre acerca del entorno, y el efecto de estos aspectos sobre el rendimiento y robustez del sistema.

Entorno

- Entorno **activo** que comprende los operadores que intervienen en el comportamiento del sistema modificando el modo de operación.
- Entorno **pasivo** que comprende a los usuarios que están solamente interesados en el servicio proporcionado y al ambiente exterior que no tiene nada que ver con la existencia del sistema.

Tolerancia a fallos

Los sistemas pueden fallar, es decir exhibir un comportamiento para el cual no fueron diseñados.

- Ya que los fallos son inevitables, un sistema de tiempo real debe ser robusto frente a ellos, comportamiento denominado ‘tolerancia a fallos’.

Tipos de sistemas

- Sistemas con parada segura (“*fail-safe*”): en caso de fallo colocan al sistema en un estado seguro (sistemas críticos).
- Sistemas con degradación aceptable (“*fail-soft*”): presentan una pérdida parcial de funcionalidad en caso de fallo, continúan funcionando a pesar de ello.

Arquitectura

- Controlada por eventos o interrupciones (“*event-driven*” o “*interrupt-driven*”).
 - Contempla la ejecución de un componente o tarea basándose en la aparición de una interrupción o señal generada por un evento externo.
 - Son mucho más eficaces para responder ante eventos externos no regulares.
- Controlada por tiempo (“*time-driven*”).
 - Opera de acuerdo con los ciclos del reloj o relojes del sistema.
 - Se utiliza cuando se precisa la ejecución de tareas periódicas o mediante temporizadores.
 - Son más sencillos y robustos (interfaces entre subsistemas muy controlados y con un funcionamiento independiente basado en relojes locales).

POSIX

El término POSIX corresponde a las iniciales de interfaz de sistema operativo portable (*Portable Operating System Interface*).

¿Qué es POSIX?

- Es un estándar orientado a facilitar la creación de aplicaciones confiables y portables. La mayoría de las versiones populares de UNIX (Linux, Mac OS X) cumplen este estándar en gran medida.
- Es un estándar que especifica la interfaz entre el sistema y el usuario. Intenta estandarizar las interfaces de los SO (Sistemas Operativos) para permitir que las aplicaciones funcionen en distintas plataformas.

Una parte importante del POSIX aborda las necesidades de las aplicaciones de tiempo real.

Objetivo

El principal objetivo de este estándar es la portabilidad de las aplicaciones a nivel de código fuente, es decir, que sea posible portar una aplicación de un computador a otro sin más que recompilar su código.

Se compone

De distintas normas y estándares llamadas “miembros”:

- POSIX.1 Interfaces del sistema (estándar básico). (a,b)
- POSIX.2 Shell y utilidades. (a)
- POSIX.3 Métodos para medir la conformidad con POSIX. (a)
- POSIX.4 Extensiones de tiempo real.
- POSIX.4a Extensión de *threads*, o múltiples flujos de control.
- POSIX.4b Extensiones adicionales de tiempo real.
- POSIX.6 Extensiones de seguridad.
- POSIX.7 Administración del sistema.
- POSIX.8 Acceso a ficheros transparente a la red.
- POSIX.12 Interfaces de red, independientes del protocolo.
- POSIX.15 Extensiones de colas *batch*.
- POSIX.17 Servicios de directorios.

a → Estándares IEEE ya aprobados.

b → Estándar ISO/IEC ya aprobado.

Estandarización de los SO en tiempo real

Debido a la necesidad de conseguir la portabilidad de las aplicaciones de tiempo real, se desarrollan estándares para añadir al POSIX básico (o UNIX) los servicios del sistema operativo necesarios para poder desarrollar aplicaciones de tiempo real.

Estas aplicaciones se caracterizan porque el funcionamiento correcto no sólo depende de los resultados del cálculo, sino también del instante en el que se generan estos resultados.

Para garantizar que los cálculos se realicen en los instantes requeridos, es preciso que el sistema de tiempo real tenga un comportamiento temporal predecible, y para ello, los servicios del sistema operativo tienen que ser capaces de proporcionar el nivel de servicio con un tiempo de respuesta acotado.

Sincronización de procesos

Semáforos

El POSIX.4 define funciones para permitir la sincronización de procesos a través de semáforos contadores. El semáforo contador es un mecanismo de sincronización muy común, que permite el acceso mutuamente exclusivo a recursos compartidos, la señalización y espera entre procesos, y otros tipos de sincronización. Uno de los usos más comunes de los semáforos es permitir que diferentes procesos puedan compartir datos; esto se consigue en POSIX.4 utilizando objetos de memoria compartida, junto con los semáforos.

Memoria compartida

Los procesos POSIX.1 tienen espacios de direccionamiento que son independientes entre sí. Sin embargo, muchas aplicaciones de tiempo real necesitan compartir grandes cantidades de datos de una manera eficiente. Esto se puede conseguir si los procesos son capaces de compartir regiones de memoria física. Con este propósito, el estándar POSIX.4 define los objetos de memoria compartida, que son regiones de memoria que pueden ser mapeadas en el espacio de direcciones de un proceso. Cuando dos o más procesos mapean el mismo objeto de memoria entonces comparten la región de memoria asociada. Si los objetos de datos que se colocan en memoria compartida requieren un acceso mutuamente exclusivo por parte de distintos procesos, se pueden utilizar semáforos para efectuar estos accesos.

Señales

El mecanismo de señales definido en el POSIX.1 permite notificar eventos que ocurren en el sistema. El estándar POSIX.4 extiende la interfaz de señales para:

- Las señales de tiempo real se guardan en colas, por lo que los eventos no se pierden.

- Las señales de tiempo real pendientes de procesado se extraen de la cola en orden de prioridad, usando el número de señal como prioridad. Esto permite diseñar aplicaciones con tiempos de respuesta más rápidos ante eventos urgentes.
- Las señales de tiempo real contienen un campo adicional de información, que la aplicación puede utilizar para intercambiar datos entre el generador de la señal y el módulo que la procesa. Por ejemplo, este campo puede ser utilizado para identificar la fuente y la causa de la señal.
- El rango de señales disponibles para la aplicación ha sido expandido.

Cola de mensajes

Se especifica un mecanismo sencillo de colas de mensajes para la comunicación entre procesos. Los mensajes tienen asociado un campo de prioridad, y se extraen de las colas en orden de prioridad. Esta facilidad permite minimizar la cantidad total de inversión de prioridad en el sistema. La recepción y la transmisión de mensajes puede hacerse tanto de forma bloqueante—si es necesario el proceso se suspende hasta que llegue un mensaje o haya espacio en la cola—como no bloqueante. La transmisión y la recepción no están sincronizadas, es decir, el transmisor no necesita esperar a que el receptor haya recibido el mensaje. Los tamaños máximos de mensajes y colas se pueden seleccionar durante la creación de la cola, lo que permite incrementar la predictibilidad de las operaciones con colas de mensajes.

Threads

El estándar POSIX.4a define interfaces para soportar múltiples actividades concurrentes, denominadas *threads*, dentro de cada proceso POSIX. Los *threads* definidos en el POSIX.4a tienen un estado asociado más pequeño que el de un proceso. Todos los *threads* que pertenecen al mismo proceso comparten el mismo espacio de direccionamiento. Pueden ser implementados con tiempos de cambio de contexto y de creación y destrucción más bajos que los de los procesos. El POSIX.4a ha sido específicamente desarrollado para abordar las necesidades de los sistemas multiprocesadores de memoria compartida. Los *threads* no sólo están pensados para aplicaciones de tiempo real, sino que también pueden ser empleados para sistemas que, no siendo de tiempo real, requieren cambios de contexto eficientes y tiempos de creación/destrucción pequeños, como en aplicaciones de ventanas, software multiprocesador, etc.

Time Outs

Algunos de los servicios definidos en el POSIX.1, POSIX.4, y POSIX.4a pueden suspender al proceso que los invoca durante un período indefinido de tiempo, hasta que los recursos necesarios para completar el servicio estén disponibles. En sistemas de tiempo real estricto es importante limitar la cantidad máxima de tiempo que un proceso puede emplear esperando a que uno de estos servicios se complete. Esto permite detectar condiciones anormales, y por tanto incrementa la robustez del programa permitiendo implementaciones tolerantes a fallos. Los tiempos límite especifican la máxima cantidad de tiempo que el proceso puede estar suspendido en espera de la terminación de un servicio. Los servicios elegidos para tener tiempos límite han sido aquellos que todavía no tenían capacidad de especificar un tiempo límite, y cuyo uso se consideró más probable en los segmentos de código en los que la respuesta temporal es crítica:

- Esperar a que un semáforo se desbloquee.
- Esperar a la llegada de un mensaje a una cola de mensajes.
- Enviar un mensaje a una cola de mensajes.
- Esperar a que un mutex sea liberado.