**INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS**

**AVANZADAS**

**Asignatura:** Sistemas Operativos en Tiempo Real

**Profesor: M**aza Casas Lamberto

**Grupo:** 3MV9

**Estándares POSIX**

**Alumno: T**éllez Pérez Luis José

Contenido

[Resumen 2](#_Toc9888506)

[Estándares POSIX 2](#_Toc9888507)

[2 EXTENSIONES DE TIEMPO REAL 4](#_Toc9888508)

[2.1 Planificación de Procesos de Tiempo Real 4](#_Toc9888509)

[2.2 Inhibición de la Memoria Virtual 5](#_Toc9888510)

[2.3 Sincronización de Procesos 5](#_Toc9888511)

[CONCLUSIONES 6](#_Toc9888512)

# Resumen

El estándar POSIX define una interfase portable para aplicaciones basadas en el popular sistema operativo UNIX2. El principal objetivo de este estándar es la portabilidad de las aplicaciones a nivel de código fuente, mediante la unificación de las diferentes versiones del UNIX. Una parte importante del POSIX aborda las necesidades de las aplicaciones de tiempo real. La portabilidad de estas aplicaciones es hoy en día prácticamente imposible debido a la gran cantidad de sistemas operativos y núcleos de tiempo real existentes. Aunque el UNIX no ha sido hasta ahora un sistema operativo para sistemas de tiempo real, es posible adaptarlo a estos sistemas si se le añaden los servicios necesarios, y se eliminan también aquellas funciones que dificultan implementaciones pequeñas y eficientes. En este artículo3 se comentarán las extensiones de tiempo real del POSIX y cómo estas extensiones permiten abordar las necesidades de aplicaciones con requerimientos de tiempo real.

# Estándares POSIX

**POSIX** significa **P**ortable**O**perating **S**ystem **I**nterface. Consiste en una familia de estándares especificadas por la IEEE con el objetivo de facilitar la interoperabilidad de sistemas operativos. Además, POSIX establece las reglas para la portabilidad de programas. Por ejemplo, cuando se desarrolla software que cumple con los estándares POSIX existe una gran probabilidad de que se podrá utilizar en sistemas operativos del tipo Unix. Si se ignoran tales reglas, es muy posible que el programa o librería funcione bien en un sistema dado pero que no lo haga en otro.

El POSIX es un grupo de estándares en evolución. Cada uno de los estándares que lo componen cubre diferentes aspectos de los sistemas operativos. Algunos de ellos ya han sido aprobados, mientras que otros están aún en fase de desarrollo. Los estándares POSIX se pueden agrupar en tres categorías diferentes:

|  |  |
| --- | --- |
| POSIX.1 | Interfaces del sistema (estándar básico) a ,b |
| POSIX.2 | Shell y utilidadesa |
| POSIX.3 | Métodos para medir la conformidad con POSIXa |
| 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 |

1. Estándares Base: Definen interfases del sistema relacionadas con diferentes aspectos del sistema operativo. El estándar especifica la sintaxis y la semántica de estos servicios del sistema operativo, de modo que los programas de aplicación puedan invocarlos directamente. El estándar no especifica cómo se implementan estos servicios; de este modo, los implementadores de sistemas pueden elegir la implementación que crean más conveniente—y así competir entre ellos—, siempre que cumplan la especificación de la interfase. Todos los estándares base desarrollados hasta el momento lo han sido para lenguaje C. En el momento de escribir este artículo está abierto el debate sobre si los estándares base deben desarrollarse de forma independiente del lenguaje, y luego especificar interfases concretas para los diferentes lenguajes de programación. La Tabla I y la Tabla II muestran los estándares base que están siendo desarrollados por los grupos de trabajo del POSIX.

|  |  |
| --- | --- |
| P1224 | Servicios de mensajería electrónica (X.400) |
| P1224.1 | Interfaz para portabilidad de aplicaciones X.400 |
| P1238 | Interfaz de comunicaciones OSI |
| P1238.1 | Interfaz OSI de transferencia de ficheros |
| P1201.1 | Interfaz gráfica a usuario (ventanas) |

1. Interfaz en diferentes lenguajes de programación: Son estándares secundarios que traducen a un lenguaje de programación concreto los estándares base. Los lenguajes utilizados hasta el momento son Ada, Fortran 77, y Fortran 90, además del lenguaje C, en el que se han especificado hasta el momento los estándares base. La Tabla III muestra las interfases POSIX que están actualmente en desarrollo para diferentes lenguajes de programación.

|  |  |
| --- | --- |
| POSIX.5 | Interfaz Ada |
| POSIX.9 | Interfaz Fortran 77 |
| POSIX.19 | Interfaz Fortran 90 |
| POSIX.20 | Interfaz Ada para las extensiones de tiempo real |

1. Entorno de Sistemas Abiertos. Estos estándares incluyen una guía al entorno POSIX y los perfiles de entornos de aplicación. Un perfil de aplicación es una lista de los estándares POSIX, con especificación de las opciones y parámetros necesarios, que se requieren para un cierto entorno de aplicación. El objetivo principal de los perfiles de aplicación es conseguir un conjunto pequeño de clases de implementaciones de sistemas operativos bien definidas y que sean apropiadas para entornos particulares de aplicaciones. La Tabla IV muestra la lista de estándares que están siendo desarrollados en este grupo.

|  |  |
| --- | --- |
| POSIX.0 | Guía al entorno POSIX de sistemas abiertos |
| POSIX.10 | Perfil de entorno de aplicaciones de supercomputación |
| POSIX.11 | Perfil de entorno de aplicaciones de procesado de transacciones |
| POSIX.13 | Perfiles de entornos de aplicaciones de tiempo real |
| POSIX.14 | Perfil de entorno de aplicaciones multiprocesadoras |
| POSIX.18 | Perfil de entorno de aplicación de plataforma POSIX |

# 2 EXTENSIONES DE TIEMPO REAL

Esta sección muestra algunas de las funciones más importantes del POSIX.4, que es la parte del POSIX que define interfases del sistema para soportar aplicaciones con requerimientos de tiempo real. El estándar POSIX.4 está muy cercano a su aprobación definitiva.

## 2.1 Planificación de Procesos de Tiempo Real

El estándar base POSIX.1 define un modelo con actividades concurrentes denominadas procesos, pero no especifica ninguna política de planificación ni ningún concepto de prioridad. Para que las aplicaciones de tiempo real puedan ser portables, es preciso especificar políticas de planificación que permitan obtener tiempos de respuesta predecibles. El POSIX.4 define tres políticas de planificación; cada proceso, a través de un atributo de planificación, puede elegir la que desee:

* SCHED\_FIFO: Es una política de planificación expulsora basada en prioridades estáticas, en la que los procesos con la misma prioridad se atienden en el orden de llegada (cola FIFO). Está política tendrá al menos 32 niveles de prioridad.
* SCHED\_RR: Esta política es muy similar a SCHED\_FIFO, pero emplea un método de rodaja temporal (round-robin) para planificar procesos de la misma prioridad. También tiene 32 niveles de prioridad como mínimo.
* SCHED\_OTHER: Es una política de planificación definida por la implementación.

La planificación expulsora de prioridad estática es una estrategia de prioridad utilizada con mucha frecuencia para sistemas de tiempo real. Es muy sencilla, y permite alcanzar altos niveles de utilización del sistema si se realiza la asignación de prioridades de acuerdo con los métodos del ritmo monotónico (rate monotonic) o plazo monotónico (deadline monotonic). Con las políticas de planificación especificadas en el estándar, junto a las funciones asociadas que permiten modificar y leer las políticas y prioridades de cada proceso, es posible planificar aplicaciones de tiempo real en sistemas operativos POSIX.

## 2.2 Inhibición de la Memoria Virtual

Aunque el estándar POSIX.1 no requiere que las implementaciones suministren mecanismos de memoria virtual, es práctica común en los sistemas UNIX el proporcionarlos. La memoria virtual presenta grandes ventajas para aplicaciones que no son de tiempo real, pero introduce una gran incertidumbre en la respuesta temporal. Con objeto de acotar los tiempos de acceso a memoria—y por tanto la respuesta temporal de la aplicación—el POSIX.4 define funciones para bloquear en memoria física o bien todo el espacio de direccionamiento de un proceso, o bien rangos seleccionados de ese espacio. Estas funciones deberán de ser utilizadas por los procesos con requerimientos temporales estrictos, así como por aquellos procesos con los que se sincronicen. De esta forma, se pueden conseguir tiempos de respuesta predecibles.

## 2.3 Sincronización de Procesos

El POSIX.4 define funciones para permitir la sincronización de procesos a través de semáforos contadores. Estos semáforos se identifican por un nombre que pertenece a un espacio de nombres definido por la implementación. Este espacio de nombres puede coincidir o no con el espacio de nombres de ficheros, por lo que no se hace necesaria la existencia del sistema de ficheros para utilizar los semáforos. 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. Desafortunadamente, los semáforos contadores especificados en POSIX.4 no evitan el fenómeno conocido por inversión de prioridad no acotada. La inversión de prioridad ocurre cuando un proceso de prioridad alta tiene que esperar a que un proceso de prioridad baja termine de utilizar un determinado recurso que tiene reservado. Utilizando los protocolos de sincronización adecuados, se puede conseguir que la inversión de prioridad esté acotada por la duración de secciones críticas, es decir, de las secciones de código durante las cuales el proceso reserva un recurso para su uso exclusivo. Sin embargo, con los semáforos convencionales, puede aparecer inversión de prioridad no acotada; esto significa que el retraso experimentado por tareas de prioridad alta no está acotado por la duración de secciones críticas, sino que depende del tiempo de ejecución total de tareas de prioridad más baja. Esta situación puede ocurrir cuando una tarea de alta prioridad está esperando a que una tarea de prioridad baja libere un semáforo que controla el acceso a un recurso compartido, y la tarea de prioridad baja es expulsada—del uso de la CPU— por una tarea de prioridad intermedia Los largos retrasos que se experimentan en estos casos son normalmente inaceptables para las tareas con requerimientos de tiempo real estricto. Si se utilizan protocolos de sincronización apropiados, la cantidad de inversión de prioridad puede quedar acotada a la duración de secciones críticas, que es normalmente muy pequeña.

# CONCLUSIONES

El POSIX es un estándar de sistema operativo en evolución, que se prevé que será ampliamente utilizado en los próximos años. Una importante parte de este estándar está pensada para proporcionar la portabilidad de las aplicaciones con requerimientos de tiempo real. Junto a las interfases de servicios del sistema, se estandarizan también perfiles de entornos de aplicaciones que permitirán a los implementadores desarrollar sistemas operativos POSIX de tiempo real para una gran variedad de plataformas, desde los sistemas empotrados pequeños hasta los sistemas de tiempo real grandes. El estándar define interfases en diferentes lenguajes de programación. En particular, las interfases de tiempo real están siendo definidas para C y Ada, que son los lenguajes estándar de programación más importantes para los sistemas prácticos de tiempo real.