

Relatoría Técnica: Simulador de Planificación de CPU - Sistema SIGET

1. Contexto Arquitectónico y Bases del Diseño

El simulador SIGET fundamenta su diseño en el contraste entre arquitecturas de hardware. Modela el **cuello de botella de Von Neumann**, donde la memoria unificada genera un atasco en el bus compartido —como una "autopista de un solo carril"—, lo que provoca latencia y riesgos de **inyección de código** al mezclar datos e instrucciones.

Para mitigar esto, integra la **arquitectura Harvard**, que utiliza buses y memorias independientes para instrucciones y datos. Esta segmentación elimina la competencia física al permitir accesos concurrentes, garantizando la **seguridad y predictibilidad** necesarias para el procesamiento en tiempo real.

2. Objetivos del Diseño: El Balance de Carga

El sistema SIGET optimiza la estabilidad y velocidad mediante dos pilares:

- **Latencia para emergencias:** Inspirado en la eficiencia de la **arquitectura Harvard**, prioriza señales críticas para eliminar la latencia de bus y el **overhead**, garantizando una respuesta inmediata de tiempo real.
- **Eficiencia rutinaria:** Maximiza el **rendimiento (throughput)** mediante una **planificación inteligente** de procesos, asegurando que el flujo masivo de datos no bloquee la capacidad de respuesta y manteniendo la estabilidad operativa.

3. Decisiones de Diseño: Atributos y Ciclo de Vida del Proceso

En el sistema SIGET, un **proceso** es una instancia asignada al procesador para su ejecución. Para funcionar, requiere tres recursos clave: **tiempo de procesamiento, memoria y acceso a archivos o dispositivos**. Basándose en su **prioridad**, el sistema gestiona el avance del proceso a través de sus **cinco estados**: Nuevo, Listo, En ejecución, Bloqueado y Terminado.

Estado	Función dentro del simulador
Nuevo	Fase de creación donde el proceso es admitido por el sistema operativo y se preparan sus estructuras de control.
Listo (Ready)	El proceso ha sido cargado en memoria principal y se encuentra en la cola de espera para que el planificador le asigne un turno en la CPU.
En ejecución (Running)	El proceso posee el control total del procesador y sus instrucciones están siendo procesadas activamente por la ALU y la unidad de control.
Bloqueado (Waiting)	El proceso suspende su ejecución voluntaria o involuntariamente a la espera de un evento externo, una señal de E/S o la disponibilidad de un recurso.
Terminado (Terminated)	La ejecución ha finalizado exitosamente o por error, lo que dispara la liberación de la memoria y los descriptores de archivos asignados.

4. Análisis de Algoritmos de Planificación Implementados

El sistema SIGET implementa dos estrategias de planificación fundamentales para gestionar la coexistencia de aplicaciones de manera eficiente:

1. Transición Exitosa de Estados: El simulador se ejecutó correctamente las etapas fundamentales de un proceso:

- **Nuevo:** Los tres procesos (**Rutina_Vial**, **EMERGENCIA**, **Semaforo**) fueron creados e instanciados.
- **Listo (Ready):** El planificador admitió los procesos a la cola de espera para ser asignados al procesador.
- **En ejecución (Running):** Se observa la asignación de la "ráfaga de CPU" a cada tarea.
- **Terminado (Terminated):** Todos los procesos finalizaron su ejecución con éxito.

Nota: En esta ejecución específica no se activó el estado **Bloqueado (Waiting)**, lo que indica que no hubo esperas por recursos externos o dispositivos de E/S en este ciclo.

2. Evidencia de Planificación y Multiprogramación: El orden de salida demuestra conceptos críticos de los sistemas operativos:

- **Prioridad:** El proceso **EMERGENCIA_Ambulancia** pasó a ejecución y terminó primero, cumpliendo con el objetivo de dar **respuesta inmediata a eventos críticos**.
- **Multiprogramación y Round-robin:** Los procesos de **Semaforo** y **Rutina_Vial** alternaron sus estados (de *En ejecución* a *Listo* y viceversa). Esto confirma que el sistema está permitiendo que **varias actividades comparten el procesador**, lo cual es la base de un sistema multitarea eficiente.

3. Análisis del Comando `ps aux` y el mensaje `Done`

El mensaje `+ Done python3 simulador_siget_rsc.py` es la confirmación definitiva de que el sistema operativo ha llevado el proceso completo al estado **Terminado**.

En cuanto al resultado de tu búsqueda con `grep`:

- **PID 118765:** Es el identificador único que el sistema asignó temporalmente al comando de búsqueda (`grep`), no a tu script.
- **Estado S+:** Indica que el comando `grep` está en un estado de **espera o dormido (Sleeping)** mientras termina de procesar la salida de texto.
- **Ausencia del script:** El hecho de que no aparezca una línea para `python3 simulador_siget_rsc.py` confirma que el proceso ya no ocupa **tiempo de procesamiento ni memoria**, pues ha finalizado su ciclo de vida.

5. Conclusiones y Optimización del Rendimiento

La **planificación de procesos** es vital para evitar el colapso del sistema y garantizar la estabilidad al organizar recursos como el tiempo y la memoria. Mientras que en sistemas antiguos solo existía el **pseudoparalelismo**, el diseño de SIGET permite alcanzar el **paralelismo real** mediante el uso de **multihilo**. Esto permite aprovechar los procesadores modernos para ejecutar tareas simultáneamente, logrando un **máximo rendimiento y eficiencia** en el hardware.