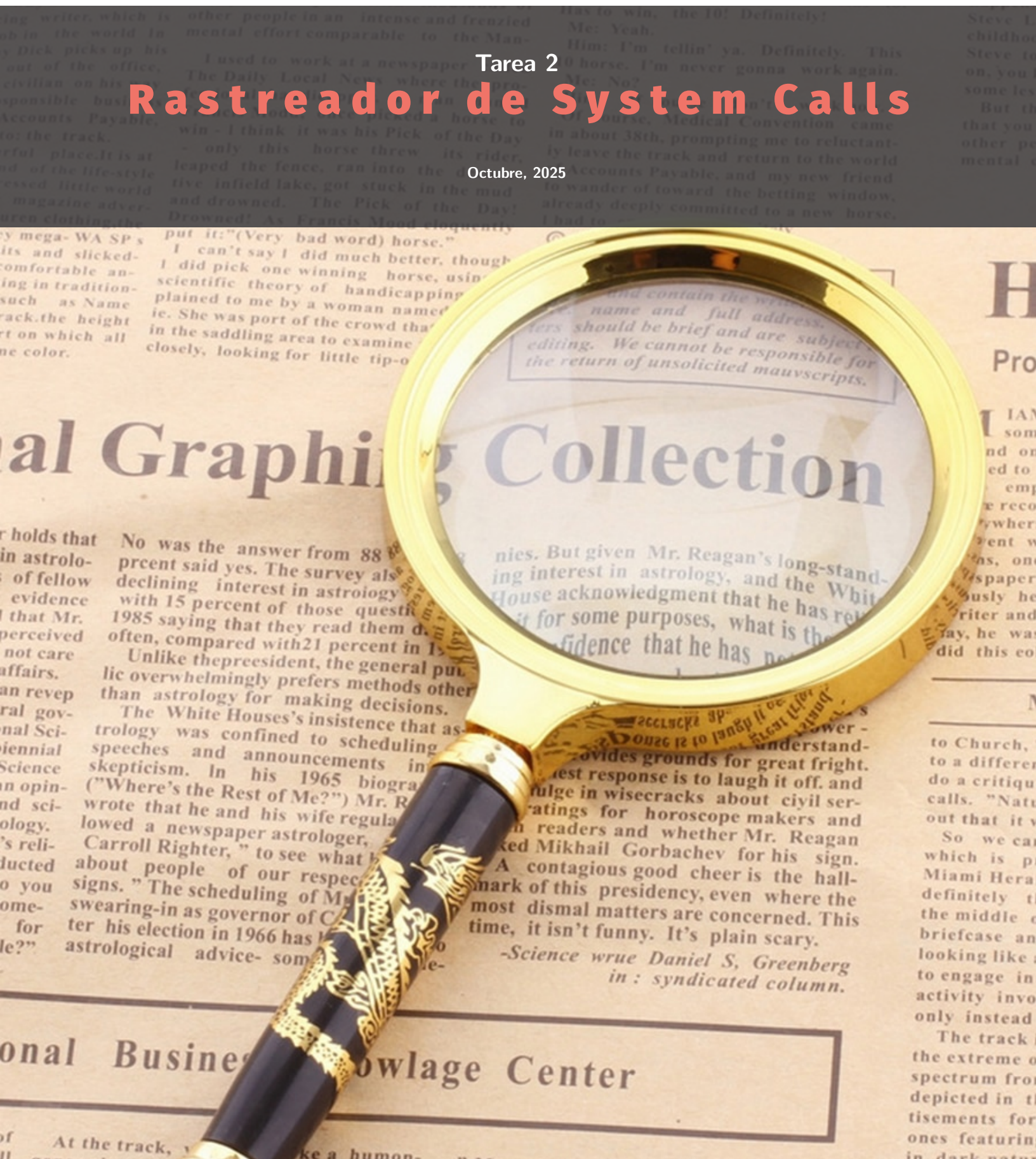


# Rastreador de System Calls

Octubre, 2025



## Objetivo

El propósito de esta tarea individual es darle familiaridad con el ambiente de POSIX y aplicaciones de Sistemas Operativos.

## Datos Generales

- **Fecha de Entrega:**  
En la plataforma.
- **Fecha de Revisión:**  
.
- **Lenguaje:**  
Rust para GNU/Linux
- **Recurso Humano:**  
Individual
- **Valor de la asignación:** 5 %

## Profesor

Kevin Moraga  
kmoragas@ic-itcr.ac.cr  
Escuela de Computación

## Introducción

Las llamadas al sistema comúnmente usan una instrucción especial del CPU que causa que el procesador transfiera el control a un código privilegiado (generalmente es el kernel) previamente especificado. Esto permite al código privilegiado especificar donde va a ser conectado, así como el estado del procesador. Cuando una llamada al sistema es invocada, la ejecución del "programa que invoca" es interrumpida y sus datos son guardados, normalmente en su PCB (Bloque de Control de Proceso del inglés Process Control Block), para poder continuar ejecutándose luego. El procesador entonces comienza a ejecutar las instrucciones de código de bajo nivel en modo privilegiado, para realizar la tarea requerida. Cuando esta finaliza, se retorna al proceso original y continúa su ejecución. El retorno al proceso original no obligatoriamente es inmediato, depende del tiempo de ejecución de la llamada al sistema y del algoritmo de planificación de CPU.

# Requerimientos

Su programa tendrá la misión de poner a ejecutar a otro programa (digámosle *Prog*), pasarle los argumentos seleccionados por el usuario y rastrear todos los system calls utilizados por *Prog*.

En todo caso, al final de la ejecución de *Prog*, rastreador siempre desplegará en la salida estándar una tabla acumulativa que muestre todos los System Calls utilizados por *Prog*, así como el número de veces que fue utilizado cada uno.

La sintaxis de ejecución desde línea de comando es:

```
rastreador [opciones rastreador] Prog [opciones de Prog]
```

- Las [opciones rastreador] podrían no venir del todo o aparecer en cualquier orden o combinación válida.
- Las [opciones de Prog] no serán analizadas ni consideradas por rastreador, sino que simplemente serán pasadas a Prog al iniciar su ejecución.

Las opciones válidas para rastreador son:

- -v desplegará un mensaje cada vez que detecte un System Call de Prog . Se debe desplegar la mayor cantidad posible de detalles respecto a cada System Call.
- -V será idéntico a la opción -v , pero hará una pausa hasta que el usuario presione cualquier tecla para continuar la ejecución de *Prog*.

# Aspectos Administrativos

## Entregables

- Código fuente del programa que cumpla los requerimientos funcionales y técnicos.
- Binario del programa, compilado para una arquitectura x86.
- Fuente de la documentación en Markdown ó Latex.
- PDF con la documentación.

## Evaluación

- Opción -v: 10 %
- Opción -V: 20 %
- Ejecución de Prog: 20 %
- Análisis de Syscalls: 30 %
- Documentación: 20 %

## Documentación

Las siguientes son las instrucciones para la documentación. NO LA IMPRIMA. Además la documentación se debe de realizar utilizando Markdown ó Latex.

1. **Introducción:** Debe presentar el problema, utilizando una redacción propia, discutido y redactado en el kick-off de la asignación.
2. **Ambiente de desarrollo:** Indicar las herramientas usadas para implementar la tarea.
3. **Estructuras de datos usadas y funciones:** Se debe describir las principales funciones y estructuras utilizadas en la elaboración de esta asignación.
4. **Instrucciones para ejecutar el programa:** Presentar las consultas concretas usadas para correr el programa para el problema planteado en el enunciado de la tarea y para los casos planteados al final de esta documentación.
5. **Actividades realizadas por estudiante:** Este es un resumen de las bitácoras de cada estudiante ( estilo timesheet) en términos del tiempo invertido para una actividad específica que impactó directamente el desarrollo del trabajo, de manera breve (no más de una línea) se describe lo que se realizó, la cantidad de horas invertidas y la fecha en la que se realizó. Se deben sumar las horas invertidas por cada estudiante, sean concientes a la hora de realizar esto el profesor determinará si los reportes están acordes al producto entregado.
6. **Autoevaluación:** Indicar el estado final en que quedó el programa, problemas encontrados y limitaciones adicionales. Adicionalmente debe de incluir el reporte de commits de git. Por otro lado, también debe incluir una calificación con la rúbrica de la sección "Evaluación".
7. **Lecciones Aprendidas** de la asignación: Orientados a un estudiante que curse el presente curso en un futuro.
8. **Bibliografía** utilizada en la elaboración de la presente asignación.
9. Es necesario documentar el código fuente.



## Aspectos Adicionales

Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones:

1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
2. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
3. El código debe ser desarrollado en el lenguaje especificado en los Datos Generales, en caso contrario se obtendrá una nota de 0.
4. Si no se siguen las reglas del formato del envío a través de Google Classroom se obtendrá una nota de 0.
5. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
6. Cada grupo tendrá como máximo 20 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
7. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final de la presente asignación.
8. Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberá avisar al menos 2 días antes de la revisión al profesor para coordinar el préstamo de estos.
9. Durante la revisión podrán participar asistentes, otros profesores y el coordinador del área.
10. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.