Universidad De San Carlos De Guatemala
Facultad De Ingeniería
Escuela De Ciencias Y Sistemas
Sistemas Operativos 2 Sección A
Ing. Edgar René Ornelis Hoil
Aux. Derek Esquivel Díaz
Vacaciones Junio 2024

Proyecto

Manejo de Memoria

Objetivos

- Entender cómo funcionan el manejo de memoria en Linux.
- Comprender como funcionan las solicitudes de memoria de los procesos.
- Realizar reportes sobre la utilización de memoria.

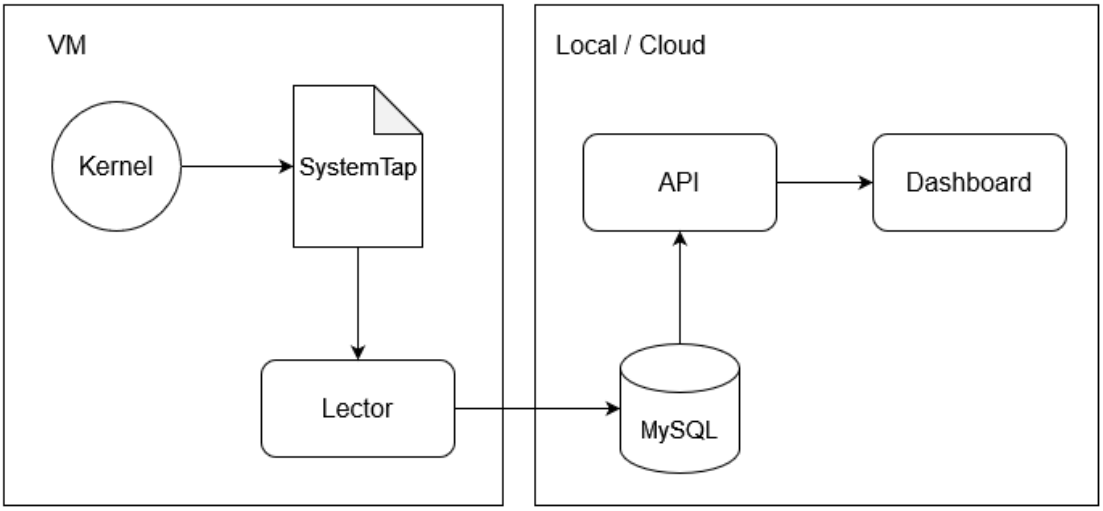
Descripción

La gestión de la memoria es un aspecto crítico del rendimiento del sistema, ya que garantiza la utilización eficiente de los recursos y mantiene la estabilidad bajo diferentes cargas de trabajo. Al monitorear exhaustivamente la memoria, se puede obtener información valiosa sobre el comportamiento de su sistema, identificar posibles cuellos de botella y optimizar la asignación de recursos.

En este proyecto se creara una aplicación capaz de monitorear el uso de memoria de cada proceso abierto en Linux, detectando las solicitudes de memoria que estos realizan al sistema operativo.

Especificaciones

Arquitectura



Script SystemTap

El script de SystemTap se encargará de capturar las solicitudes de memoria de todos los procesos activos. Estas solicitudes son realizadas por medio de estas llamadas al sistema:

- void *mmap(void *addr[.length]*, size_t *length*, int *prot*, int *flags*, int *fd*, off_t *offset*);
- int munmap(void *addr[.length]*, size_t *length*);

mmap() crea una nueva asignación en el espacio de direcciones virtuales del proceso de llamada. La dirección inicial para la nueva asignación se especifica en *addr*. El argumento de *length* especifica la longitud del mapeo (que debe ser mayor que 0).

La llamada al sistema munmap() elimina las asignaciones para el rango de direcciones especificado.

La captura de estas llamadas se realizaran por medio de un script de SystemTap, que es una herramienta para la instrumentación dinámica para sistemas operativos basados en el Linux por el cual los administradores del sistema pueden extraer, filtrar y resumir datos para permitir el diagnóstico de problemas complejos de rendimiento o funcionales.

Cada vez que mmap() o munmap() sean llamadas por algún proceso el script deberá de escribir obtener los datos necesarios para poder realizar un análisis del uso de memoria (PID, longitud del segmento de memoria, fecha de la solicitud, etc.)

Lector

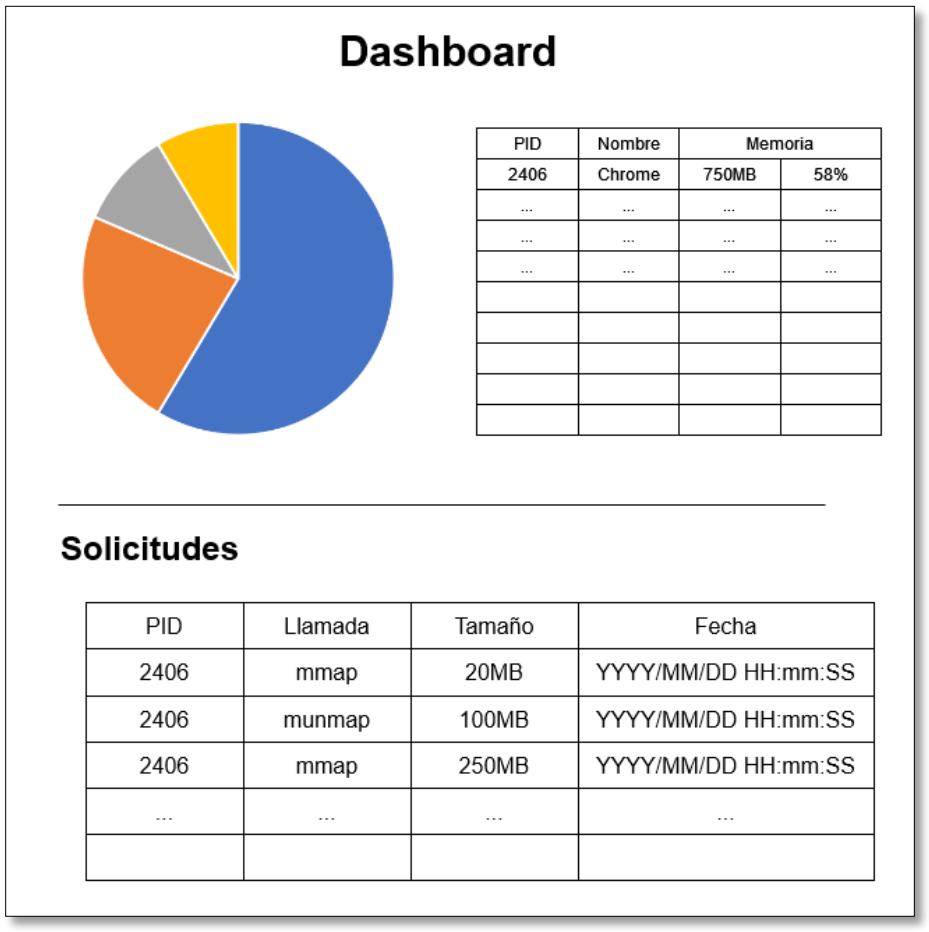
Este será un programa en C estará leyendo el output del script de SystemTap. Cada vez que el módulo de kernel escriba una nueva solicitud, este programa deberá de ser capaz de leerlo y subir esta llamada a una base de datos de MySQL.

Se deberá de almacenar como mínimo la siguiente información:

- PID del proceso.
- Nombre del proceso.
- Llamada.
- Tamaño del segmento de memoria solicitado o liberado.
- Fecha y hora de la solicitud.

Dashboard

Este será una aplicación web donde el usuario podrá visualizar información del uso de memoria.



En este dashboard el usuario deberá de ser capaz de ver una tabla donde se podrá leer:

- PID del proceso.
- Nombre del proceso.
- Cantidad de memoria que posee el proceso (mmap - munmap, debido a que el programa no se corre en startup puede que el resultado sea negativo, por lo que se debera mostrar 0).
- Porcentaje al que equivale la memoria del proceso al total de memoria virtual del sistema.

Así mismo se agregará una gráfica de pie donde se muestre que porcentaje de toda la memoria solicitada fue realizada por cada proceso. Aquí deberán de mostrar un máximo de 10 procesos, en caso de haber más serán agrupados en una sección de “otros”.

Por último, se debe incluir una tabla donde se listen todas las llamadas solicitudes realizadas.

El dashboard deberá ser actualizado en tiempo real, por lo que se recomienda la utilización de sockets para la comunicación con la api.

Observaciones

- La práctica se realizará en parejas.
- El lenguaje de programación a utilizar para el proceso lector será C.
- El lenguaje de programación a utilizar en la API y en el Dashboard quedan a elección del estudiante.
- La base de datos a utilizar será MySQL.
- El script de SystemTap y el proceso lector deberán de estar instalados en una VM de Linux Mint. La API y el Dashboard podrán estar instalados de manera local o en la nube (a elección del estudiante).
- Cualquier copia parcial o total será reportada a la Escuela de Ciencias y Sistemas para que proceda como indica el reglamento.

Entregables

- Código fuente
- Manual técnico con explicación de lo realizado en la proyecto. Este debe ser un archivo .md (Markdown) y será colocado como el README de la carpeta.

Forma de entrega

- Utilizar un repositorio de GitHub compartido entre los dos integrantes con el nombre: **SO2_GRUPO<<no. grupo>>..**
- Agregar al auxiliar al repositorio: **Desquivel501**
- Ambos estudiantes realizaran la entrega por medio de UEDI en el apartado correspondiente, donde el estudiante subirá un archivo de texto con el link al repositorio.

La entrega se debe realizar el 26 de junio de 2024 antes de las 23:59.