UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

SISTEMAS OPERATIVOS 1 SECCIÓN A

ING. SERGIO MENDEZ

AUX. JOSÉ DANIEL VELÁSQUEZ OROZCO

ESCUELA DE VACACIONES - JUNIO 2024

PROYECTO 1

Plataforma de Monitoreo de Procesos

Objetivos

- Conocer el Kernel de Linux mediante módulos de C.
- Hacer uso de programación asíncrona con Rust.
- Comprender el funcionamiento de los contenedores usando Docker.

Introducción

Este proyecto tiene como objetivo principal implementar un sistema de monitoreo de recursos del sistema y gestión de procesos empleando varias tecnologías y lenguajes de programación. El sistema resultante permitirá obtener información clave sobre el rendimiento del computador, procesos en ejecución y su administración a través de una interfaz amigable.

VM a Monitorear

Parte del proyecto se deberá implementar en una máquina virtual de Ubuntu Server 22.04 sin GUI, virtualizada en el hipervisor KVM.

Módulos de Kernel

Módulo de RAM

Este módulo deberá de estar alojado en un archivo ubicado en el directorio /proc.

Características:

- Importar la librería **<linux/mm.h>**
- La información que se mostrará en el módulo debe ser obtenida por medio de los struct de información del sistema operativo y no por otro medio.
- El nombre del módulo será: **ram_so1_jun2024**

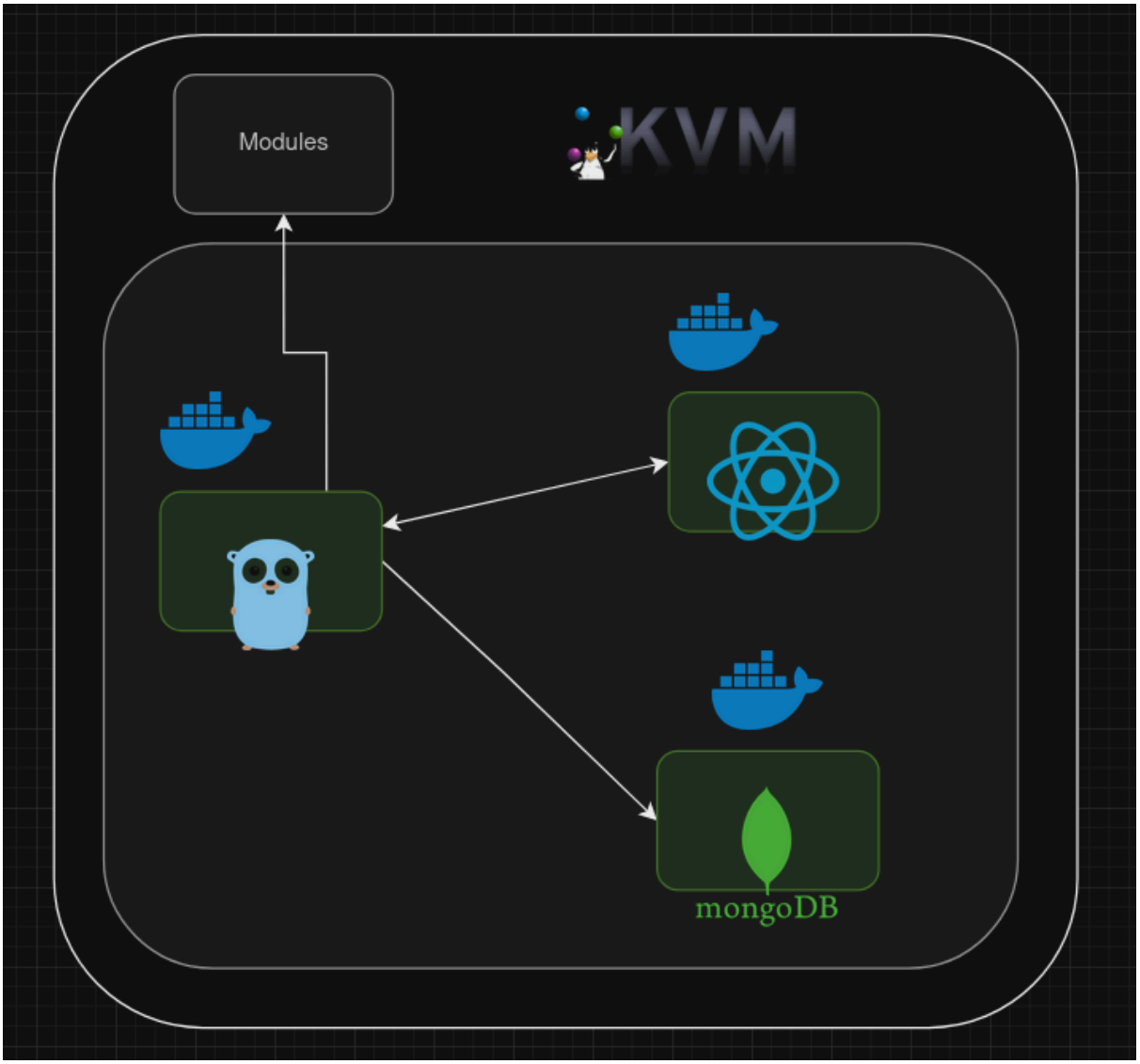
Módulo de CPU

Este módulo deberá de estar alojado en un archivo ubicado en el directorio /proc.

Características:

- Importar la librería **<linux/sched.h>**
- La información que se mostrará en el módulo debe ser obtenida por medio de los struct de información del sistema operativo y no por otro medio.
- El nombre del módulo será: **cpu_so1_1s2024**

Arquitectura



Plataforma de Monitoreo

Frontend

Se debe de crear una Aplicación Web desarrollada en el framework de su elección que se divide en:

Monitoreo en Tiempo Real

Se debe mostrar:

- Gráfica en Tiempo Real del porcentaje de utilización de la memoria RAM (obtenidos mediante el módulo de kernel instalado).

- Gráfica en Tiempo Real del porcentaje de utilización del CPU. (Obtenido mediante el comando mpstat)

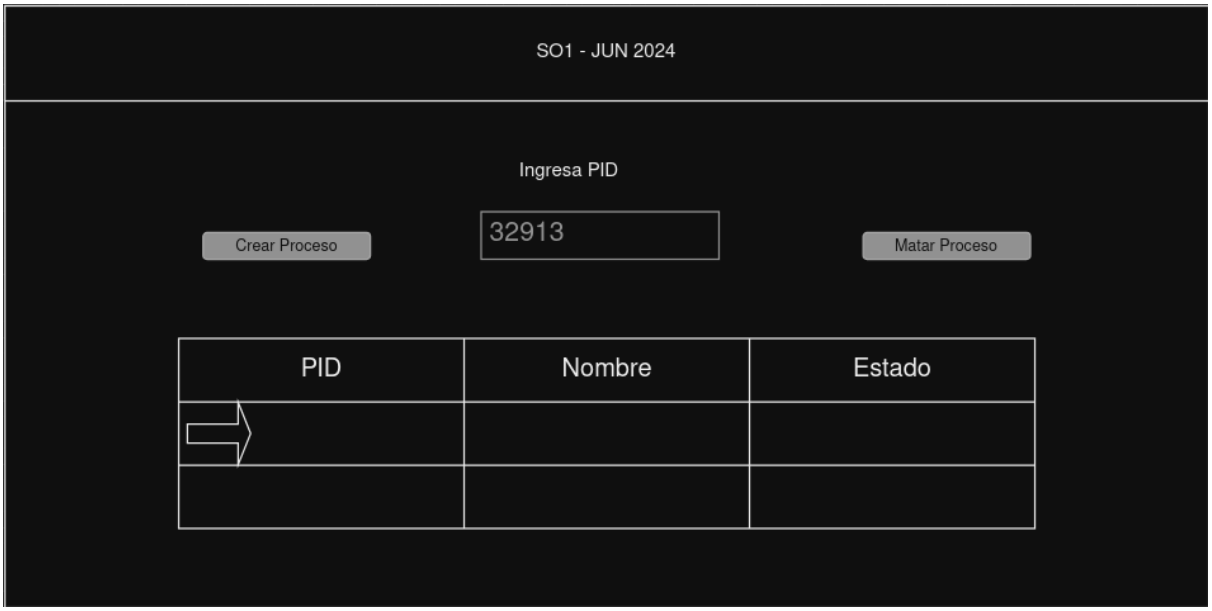
Tabla de Procesos

En esta página de debe de Agregar:

- Una tabla que desglosa los procesos y sus hijos (si existen).
- Un botón para crear un proceso sleep infinity en el kernel del sistema operativo.
- Un botón para eliminar dicho proceso mediante su PID.

Nota:

- Los procesos deben de ser obtenidos de la información que brinda el módulo de CPU.



API GOLANG

- Se encargará de realizar los llamados a los módulos ubicados en la carpeta proc y almacenar los datos en la BD de MongoDB.
- Realizará el envío de datos hacia el monitoreo de gráficas en tiempo real.

- Realizará el envío de datos necesarios para la tabla de procesos en tiempo real.
- Tiene como tarea realizar las operaciones de creación y eliminación del proceso sleep infinity.

Base de datos

Se debe implementar una base de datos MongoDB por medio de un contenedor de Docker, esto para guardar los datos de los procesos padre, procesos hijos, información de memoria ram y cpu. La base de datos debe de tener persistencia por lo que se requiere un Volumen de Docker para evitar que los datos se pierdan cada vez que el contenedor se reinicia.

Docker Compose

Para facilitar el despliegue de los contenedores de la Plataforma de Monitoreo se utilizará Docker Compose el cual permite gestionar múltiples contenedores en un solo bloque lo cual facilita la administración.

Docker Hub

Se debe utilizar Docker Hub para alojar las imágenes de los contenedores utilizados.

Pruebas de Stress

Para verificar el funcionamiento correcto de las gráficas y obtención de datos se estará usando el módulo de Linux llamado Stress.

Requisitos Mínimos

- Utilización de Proxies.
- Desplegar frontend en el puerto 80.
- Documentación
- KVM con Ubuntu Server sin GUI

Restricciones

- El proyecto se realizará en parejas.
- La obtención de la información se hará a través de los módulos de Kernel en C a excepción del porcentaje de CPU.
- La API se realizará con Golang.
- El Frontend queda a elección del estudiante.
- La máquina virtual deberá ser de Ubuntu Server 22.04.
- Las imágenes de los contenedores deben de ser publicadas de Docker Hub.

Github

- El código fuente debe de ser gestionado por un repositorio privado de Github
- Crea una Carpeta en el repositorio llamado: Proyecto1
- Agregar al auxiliar al repositorio de GitHub: **DannyLaine158**

Calificación

- Al momento de la calificación se verificará la última versión publicada en el repositorio de GitHub
- Cualquier copia parcial o total tendrán nota de 0 puntos y serán reportadas al catedrático y a la Escuela de Ciencias y Sistemas.
- Si el proyecto se envía después de la fecha límite, se aplicará una penalización del 25% en la puntuación asignada cada 12 horas después de la fecha límite. Esto significa que, por cada período de 12 horas de retraso, se reducirá un 25% de los puntos totales posibles. La penalización continuará acumulándose hasta que el trabajo alcance un retraso de 48 horas (2 días). Después de este punto, si el trabajo se envía, la puntuación asignada será de 0 puntos.

Entregables

- La entrega se debe realizar antes de su horario de calificación del 16 de junio de 2024.
- La forma de entrega es mediante UEDI subiendo el enlace del repositorio.
- Manual técnico con explicación de todos los componentes utilizados en el proyecto por ejemplo web UI, database, tabla de procesos, módulos, etc. Este manual debe ser un archivo .md (Markdown), por lo que deberá estar en el README del repositorio.