TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
CE 4303 — Principios de Sistemas Operativos

Proyecto 2
Driver para histograma

Fecha de asignación:	21 de Octubre, 2025	Fecha de entrega:	21 de Noviembre, 2025
Grupos:	4 personas máximo	Profesor:	Jason Leitón Jiménez

1. Objetivo

Implementar las diferentes capas necesarias para que una aplicación pueda interactuar con hardware a través del sistema operativo, así como fomentar el diseño y creatividad de los estudiantes en la integración del software y hardware.

Implementar un algoritmo distribuidor de cargas de procesamiento en un sistema distribuido para optimizar el tiempo de respuesta, utilizando OpenMPI con información cifrada.

2. Atributos a evaluar

- Aprendizaje continuo. Se requiere que el estudiante valore las estrategias y el conocimiento adquirido para alcanzar el objetivo.
- Trabajo en equipo. Se requiere que el estudiante sea capaz de adaptarse y tabajar en grupo de acuerdo a roles establecidos.

3. Motivación

Con el desarrollo de este proyecto, se implementará un prototipo hardware, el cual deberá de tener su respectivo **device driver (módulo de driver)** y todas las capas de software que este requiere para que puedan interactuar el hardware y SO por medio del mismo. Se desarrollarán y reforzarán técnicas de integración del hardware y software por medio del device driver, además se detallará el proceso de comunicación necesario del sistema operativo hacia el hardware del computador para que la interacción sea adecuada. También se enfocará en el procesamiento distribuido para equilibrar cargas dentro de un sistema con varios nodos.

4. Descripción

El proyecto relaciona 4 áreas de los sistemas operativos: drivers, hardware, redes y procesamiento distribuido.

En general, el proyecto cuenta con las capas que se muestran en la figura 1, las cuales son de suma importancia por los insumos que necesitan y proporcionan para efectuar de buena manera el procesamiento.

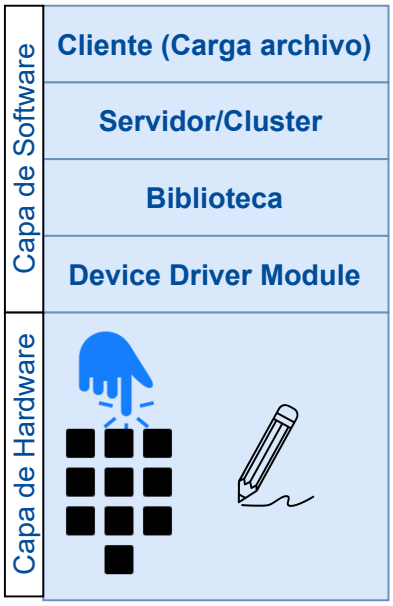

Figura 1: Capas generales del proyecto

4.1. Dispositivo físico

Se deberá crear un prototipo de hardware, utilizando cualquier sistema embebido con GPIO. Como parte del diseño se deberá elegir alguna interfaz para interactuar con el sistema (GPIO, RJ45, VGA, o cualquier otro), el cual debe contar con una justificación del porqué de dicha elección (Se aconseja que sea GPIO).

El dispositivo será el encargado de representar el histograma de los pixeles de la imagen filtrada. El objetivo es visualizar de alguna manera, el histograma correspondiente al resultado del procesamiento distribuido. La manera en que se deba de representar quedará a total libertad y creatividad de cada grupo, dentro de las opciones más comunes suelen ser, pantallas, leds, matrices de leds, vga, audio, entre otros.

4.2. Driver

Se deberá desarrollar un device driver en el lenguaje de programación C y utilizando algún ambiente de Linux. Será el encargado de proveer a las capas superiores las primitivas necesarias para la interacción con el dispositivo físico. La interacción con el dispositivo físico será mediante

el device driver (encargado de que el sistema operativo lo reconozca), en otro caso la nota será 0 en los rubros correspondientes. Es importante que considere que el device driver necesita ser un módulo del kernel de Linux (verificar con lsmod que se esté ejecutando). Se debe prestar atención a NO utilizar ningún driver del sistema para la interfaz que se utilice. Es importante resaltar que cada grupo debe saber defender el código del driver, en caso contrario, la nota será cero. No se permite únicamente utilizar un driver comercial, si no que debe estar justo a la medida para que lo se vaya a implementar.

4.3. Biblioteca

Se deberá crear una biblioteca (se espera un .a con los métodos específicos), la cual es la única que interactuará con el driver desarrollado en el punto anterior. La función de este módulo es proporcionar un conjunto de funciones consumibles al usuario para que pueda interactuar con el hardware a través del device driver. Se debe generar el .a. Esta biblioteca será la encargada de proporcionar funciones como Read y Write, Move, LedOn, entre otras.

4.4. Servidor-Clúster

Será el encargado de controlar el flujo de información, y es quien le brinda órdenes al hardware utilizando la biblioteca. El clúster estará compuesto por 4 nodos, 3 computadoras esclavas y el nodo maestro (sistema embebido). Este último será el encargado de distribuir los datos y recolectar los resultados, así como controlar el hardware.

El master será el encargado de orquestar el procesamiento distribuido entre los 3 nodos, además debe de recibir el resultado final proveniente de los nodos de procesamiento. El máster recibirá una imagen de cualquier tamaño en RGB y deberá repartirla de manera equitativa por los nodos conectados. Cada nodo deberá de recibir la parte de imagen que corresponde en escala de grises y aplicará el filtro llamado Sobel, el cual consiste en una máscara de 3x3 definida en un archivo de configuración en el máster (También debe de distribuirse a los nodos).

Una vez que los esclavos terminen de aplicar el filtro (las 3 computadores) enviarán el resultado al máster (embebido) para que posteriormente pueda unir todos los resultados y generar el histograma tanto en una imagen como en el hardware para que comprobar la forma.

Cabe destacar que cada nodo esclavo debe guardar en el file system la porción de imagen procesada que le correspondió ejecutar. Durante el procesamiento se debe verificar, utilizando topo o el monitor del sistema, el uso de todos los núcleos de los esclavos mientras se está aplicando el filtro.

4.5. Nodo master (embebido)

Será el encargado de tomar el nombre por consola de la imagen a procesar y distribuir los datos a los otros nodos. Posteriormente recolectará los resultados y será el único encargado de

interactuar con el hardware. Además, debe de tomar métricas de tiempo de latencia en la red, tiempo de procesamiento de cada nodo, cantidad de datos transferidos y una métrica propuesta por cada grupo.

4.6. Requerimientos técnicos

- Este proyecto se debe realizar en el lenguaje de programación C o RUST y como framework para la parte distribuida openMPI.
- Debe ser implementado en Linux, sin máquinas virtuales.
- No se permite soluciones "alambradas".
- Se debe prestar especial atención a los errores de acceso a memoria o utilización de recursos. Es inaceptable el error *segmentation fault*.
- En cada nodo se debe mostrar el monitor del sistema para ver el consumo de recursos.

5. Documentación- Estilo IEEE-Trans

- Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto. Para el atributo de **aprendizaje continuo** debe responder las siguientes preguntas:
  - ¿Cuales son las necesidades actuales de aprendizaje para enfrentar el proyecto?
  - ¿Cuáles son las tecnologías que se pueden utilizar para el desarrollo?
  - ¿Cuáles acciones se implementó para el desarrollo del proyecto (organización de tiempo, búsqueda de información, repaso de contenidos, entre otros)?
  - Evalúe de forma crítica la eficiencia de las acciones implementadas en el contexto tecnológico.

  Para el atributo de **Trabajo individual** y en equipo se debe especificar 7 puntos (Se debe colocar pregunta y respuesta), los cuales son los siguientes:
  - Indicar las estrategias para el trabajo individual y en equipo de forma equitativa e inclusiva en las etapas del proyecto (planificación, ejecución y evaluación).
  - Indicar la planificación del trabajo mediante la identificación de roles, metas y reglas.
  - Indicar cuales acciones promueven la colaboración entre los miembros del equipo durante el desarrollo del proyecto.
  - Indicar cómo se ejecutan las estrategias planificadas para el logro de los objetivos.

- Indicar la evaluación para la el desempeño del trabajo individual y en equipo
- Indicar la evaluación para las estrategias utilizadas de equidad e inclusión.
- Indicar la evaluación para las acciones de colaboración entre los miembros del equipo

6. Entregables

- Código fuente con documentación interna.
- Documentación.
- Archivos necesarios para ejecutar el programa.

7. Evaluación

- Driver 20 %
- Hardware 15 %
- Métricas 10 %
- Máster 10 %
- Biblioteca 10 %
- Clúster 25 %
- Documentación 10 %

8. Fecha de entrega

- 21 de Noviembre. 14:00 por tecdigital.

9. Puntos extras (10 puntos)

Cada grupo que lo implemente en windows, sin embargo, los demás requerimientos deben de seguir cumpliendose, por ejemplo, que sea desarrollado en C, RUST o ensamblador, utilizando alguna de las interfaces deadas y utilizando algún framework como OpenMPI para la distribución de cargas de procesamiento.

10. Otros aspectos administrativos

- Para la revisión del proyecto se debe de entregar tanto la documentación como la implementación del software.
- No se reciben trabajos después de la hora indicada.
- En la revisión del proyecto pueden estar presentes el coordinador y asistente.
- Es responsabilidad del estudiante proveer los medios para poder revisar la funcionalidad del software, por ejemplo, si no se realiza la interfaz, se debe de proporcionar otro medio para la verificación, de lo contrario la nota será cero en los rubros correspondientes a la funcionalidad faltante.