

---

## Proyecto 2

### Evaluador de servidores con diferentes procesos

---

Fecha de asignación: 3 de Noviembre, 2020  
Grupos: 3 personas

Fecha de entrega: 26 de Noviembre, 2020  
Profesores: Jason Leitón, Luis Barboza

---

## 1. Objetivo

Evaluar el comportamiento de los diferentes tipos de procesos con las distintas maneras de procesar datos por medio de la arquitectura cliente-servidor.

## 2. Atributos a evaluar

- Aprendizaje continuo. Se requiere que el estudiante valore las estrategias y el conocimiento adquirido para alcanzar el objetivo.
- Herramientas de Ingeniería. Se requiere que el estudiante sea capaz de adaptar técnicas, recursos y herramientas modernas para la solución de problemas.

## 3. Motivación

El procesamiento de datos es un área que ha estado en pleno auge en los últimos años. Cada vez más se requiere paralelismo con el fin de ejecutar más procesos en menor cantidad de tiempo, así mismo, ese procesamiento usualmente es distribuido. Por este motivo es que los Ingenieros en Computadores deben de analizar las diferentes maneras de crear un cliente-servidor, con el fin de evaluar cual es la mejor opción para implementar, dependiendo de la aplicación para la cual se está realizando.

## 4. Descripción

Este proyecto consiste en implementar una arquitectura cliente-servidor de tres maneras distintas. La idea fundamental es crear un servidor cuya función sea recibir consultas y procesar lo que se desea, por parte de los clientes. Es importante recalcar que se deberá de implementar 3 servidores independientes, es decir, cada uno con su propio ejecutable, sin embargo, todos cumplirán la misma función, la cual consiste en recibir una imagen de cualquier tamaño (arbitrariamente grande) y aplicar el filtro Sobel (pueden utilizar una biblioteca u otro lenguaje para aplicar el filtro) a dicha imagen, una vez aplicado el filtro, guardará las primeras 100 imágenes

procesadas en un directorio elegido por cada grupo, este directorio debe ser distinto para cada servidor.

#### 4.1. Servidores

La maneras que se va implementar cada servidor corresponden a las siguientes 3:

1. **Secuencial:** Consiste en ejecutar cada consulta con un único proceso y por medio de FIFO.
2. **Heavy Process:** Consiste en procesar cada solicitud con un “proceso pesado”, utilizando fork. Al terminar de procesar la solicitud el proceso muere. El proceso principal crea cada proceso y se regresa inmediatamente a recibir más consultas.
3. **Pre Heavy Process:** Al iniciar el servidor se le debe pasar por línea de comandos la cantidad de procesos que debe crear con fork. La idea fundamental de este método es que los procesos pesados estén creados y el servidor asigne consultas a los que estén disponibles. Cuando los procesos terminan la solicitud entran en un estado de “reposo”, el cual indica que está a disposición del servidor para ejecutar otra consulta. Debe existir un método elegante para terminar o parar el servidor, esto implica terminar con los procesos creados. No se permite utilizar “busy waiting”.

#### 4.2. Cliente

Los clientes servirán para estresar a los servidores con solicitudes y con ello evaluar el rendimiento de cada uno de estos. La manera en que se van a ejecutar los clientes es con el siguiente patrón: **cliente** < *ip* > < *puerto* > < *imagen* > < *N – threads* > < *N – ciclos* >. Donde *ip* es la que posee el servidor, el puerto corresponde al puerto lógico del servidor, la imagen corresponde al archivo que se desea filtrar, *N-threads* corresponde al número de hilos que debe estar enviando solicitudes y *N-ciclos* corresponde a la cantidad de veces que se va a repetir el mismo proceso.

Una vez procesadas todas las solicitudes el cliente debe de proporcionar los datos para obtener una serie de estadísticas. Estos datos serán el insumo del visualizador.

#### 4.3. Visualizador

Será un módulo de software que mostrará las estadísticas del procesamiento de los tres servidores. Puede ser implementado en cualquier lenguaje. El visualizador debe de mostrar al menos tres gráficas: La primera debe ser la cantidad de solicitudes - Tiempo en ejecutar todas (se tiene que mostrar las curvas de los tres servidores). La segunda es cantidad de solicitudes- Tiempo promedio por solicitud (se debe mostrar la curva de los tres servidores). La tercera es por elección de cada grupo.

#### 4.4. Requerimientos técnicos

- Este proyecto se debe realizar en el lenguaje de programación C, excepto los apartados antes mencionados. Pueden utilizar las bibliotecas que sean necesarias.
- Debe ser implementado en Linux y se debe proporcionar un makefile.
- 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*.

### 5. Documentación- Estilo IEEE-Trans (máximo 5 páginas)

- Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- Ambiente de desarrollo: Todos los detalles de implementación y herramienta durante el desarrollo del proyecto.
- Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto (documento aparte que explique cómo se reforzó).
- Detalles del diseño del programa desarrollo, tanto del software como del hardware (en caso de que aplique): Diagramas UML (obligatorio), diagrama de arquitectura (obligatorio), diagrama de funcionalidades (obligatorio), imágenes, descripciones entre otros, todo lo que sea necesario para entender de una mejor manera el diseño y funcionamiento del proyecto.
- Instrucciones de cómo se utiliza el proyecto.
- Tabla de actividades por cada estudiante: bitácora con el total de horas trabajadas.
- Conclusiones
- Sugerencias y recomendaciones.
- Referencias

## 6. Entregables

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

## 7. Evaluación

- Servidor FIFO 15 %
- Servidor Heavy Process 15 %
- Servidor Pre Heavy Process 15 %
- Vizualizador 15 %
- Cliente 15 %
- Makefiles 5 %
- Documentación 20 %

## 8. Fecha de entrega

- 26 de Noviembre. 23:55 por tecdigital.

## 9. 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.