

Tarea 1: Filtrador y clasificador de imágenes como servicio.

1. Motivación.

La ejecución de aplicaciones o procesos en segundo plano provee algunas ventajas en el uso de las mismas, por ejemplo, se puede utilizar como un WebServer. En esta asignación se desarrollará un *Daemon Linux* utilizando SysVinit o Systemd. El Daemon implementará la funcionalidad de un servidor web cuya función principal será el procesamiento de imágenes.

2. Objetivo.

Implementar un servidor como Deamon utilizando el lenguaje de programación C, el cual se iniciará su función en el arranque del sistema operativo. El servidor utilizará el protocolo HTTP para la consultas y transferencia de archivos.

3. Información general

- Valor del proyecto: 5%
- Nombre del programa: Filtrador y clasificador de imágenes como servicio.
- Este proyecto se debe implementar en grupos de 4 estudiantes.
- Cualquier fraude detectado significará una nota de cero y su proceso respectivo.

4. Requerimientos funcionales.

Para la implementación de este proyecto será necesario considerar lo siguiente:

- ✓ El servidor será desarrollado en el lenguaje de programación C.
- ✓ Se utilizará conceptos relacionados con procesos e hilos (socket, bind, listen)
- ✓ Para el intercambio de información entre los clientes y el servidor se utilizará el protocolo HTTP (GET,POST, transferencia de archivos).

A. El servidor.

Será el módulo que realiza todo el procesamiento con los archivos. Los archivos (imágenes de cualquier formato) se procesarán de acuerdo con el tamaño de los mismos, es decir, se procesará primero los archivos más pequeños (analogía con el proceso más corto). El servidor se encargará de dos funciones principales, la primera corresponde a recibir una imagen y aplicarle un filtro de mediana y media, el cual consiste en mejorar el contraste de los colores de una imagen. La imagen se guardará en una ruta predeterminada que se define en el archivo de configuración.

La segunda función del servidor es clasificar imágenes de acuerdo con el color predominante en la misma, es decir, habrá tres directorios con los nombres de “verdes”, “rojas”, “azules”, en los cuales se almacenarán las imágenes con mayor cantidad de color, según corresponda.

El puerto que utilizará el servidor por defecto es 1717, sin embargo, este puede cambiar especificándolo en el archivo de configuración. Cada vez que inicie el servidor actualizará dicho valor.

El archivo de configuración estará en el directorio “/etc/server/config.conf”. También se deberá crear un archivo de registro, donde se almacene la actividad del servidor, es decir, que guarde el cliente que realizó la petición, el archivo analizado, estado de la ejecución y hora de la misma.

El siguiente es un ejemplo del archivo de configuración.

```
Puerto:1719
DirColores:/jason/carpeta1/carpeta2
DirHisto: /jason/dir1/dir2
DirLog: /carpeta1
```

El servidor debe iniciar cuando arranca el sistema (asegurándose de incorporar todas las dependencias) y la implementación del mismo quedará a diseño de los creadores, puede utilizar SysVinit or Systemd service, se debe justificar la elección. La distribución de Linux es libre, pero se debe de tomar en cuenta que el desarrollo puede variar de una versión y otra.

El nombre del servidor será ImageServer y deberá implementar las funciones de *start*, *stop*, *status*, *restart*. Las cuales consisten en iniciar, parar, reiniciar (debe cargar los datos de configuración) y ver el estado del servidor en cualquier momento desde la consola. Además se debe de implementar el Stop(SO): cuando el equipo se apaga el SO llama a stop de los servicios.

La siguiente imagen es un ejemplo de ejecución del servidor.

```
# service ImageServer stop
Stopping ImageServer... done

# /etc/init.d/ImageServer stop
Stopping ImageServer... done

# systemctl stop ImageServer

# service ImageServer restart
Restarting ImageServer... done

# /etc/init.d/ImageServer restart
Restarting ImageServer... done

# systemctl restart ImageServer

# service ImageServer start
Starting ImageServer... done

# /etc/init.d/ImageServer start
Starting ImageServer... done

# systemctl start ImageServer

# service ImageServer status
daemon: ImageServer is running (pid 17039)

# /etc/init.d/ImageServer status
daemon: ImageServer is running (pid 17039)

# systemctl status ImageServer
```

Es importante recalcar que el estudiante debe proveer los medios para la comprobación de estos comandos, es decir, listar los procesos en segundo plano actuales o alguna otra manera.

B. Los clientes.

Consiste en una aplicación móvil y una página web. La idea fundamental de los clientes es proveer una interfaz gráfica para que el usuario elija la imagen que desea analizar y con ello enviarla al servidor para que proceda con el filtrado de la misma. Esta etapa debe

ser lo más amigable con el usuario, debe considerar el módulo de configuración si es necesario (ip, puerto o algún otro dato).

C. La máquina virtual.

El servidor deberá estar en la nube corriendo en una máquina virtual. Se aconseja que puede ser Azure o cualquier otro equivalente. Lo anterior permite que cualquier integrante pueda manipular el servidor por medio de SSH.

5. Requerimientos técnicos.

- ✓ Este proyecto se debe realizar en el lenguaje de programación C, la interfaz gráfica no tiene restricción de lenguaje, sin embargo, se le debe de indicar al profesor con anterioridad.
- ✓ El programa debe ser implementado en Linux (gcc), debe realizar un *makefile*.
- ✓ De debe crear los *scripts* de inicialización del server, así como el eliminar el servicio.

6. Documentación.

- ✓ 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).
- ✓ Detalles del diseño del programa desarrollo, tanto del software como del hardware: Diagramas UML, imágenes, descripciones entre otros, todo lo que sea necesario para entender de una mejor manera el diseño y funcionamiento del proyecto.
- ✓ Instrucciones de como se utiliza el proyecto.
- ✓ Coevaluación.
- ✓ Tabla de actividades por cada estudiante: bitácora.
- ✓ Conclusiones
- ✓ Sugerencias y recomendaciones.
- ✓ Referencias

Tarea 1.

7. Entregables

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

8. Evaluación

- ✓ Filtro: 15%
- ✓ Clasificador de imágenes: 15%
- ✓ Server (como deamon): 25%
- ✓ Máquina virtual en la nube 5%
- ✓ Interfaz gráfica (cliente): 10%
- ✓ Coevaluación: 5% (definida por los estudiantes)
- ✓ Documentación: 20%

9. Fecha de entrega.

- ✓ 25 de Marzo 17:00 por tecdigital.

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.

11. Ejemplo del filtro.

