
Tarea 1

Contador de consonantes

Fecha de asignación: 11 de Agosto, 2022
Grupos: 3

Fecha de entrega: 25 de Agosto, 2022
Profesor: Jason Leitón Jiménez

1. Objetivo

Implementar un servicio que se comporte como de demonio, con el fin de proporcionar recursos a un cliente utilizando virtualización de recursos tanto en máquina virtual como en contenedor.

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

La virtualización de los recursos es una técnica que ha estado en pleno auge, debido a que cada vez más se requiere aplicaciones independientes de la arquitectura donde se desarrolla, además de que existen equipos más especializados con los cuales se puede ejecutar mejor las tareas computacionales. Esta técnica está estrechamente relacionadas con los SO porque implica administrar los elementos del hardware, por lo que es de suma importancia que los Ingenier@s en Computadores se familiaricen con los entornos de virtualización tanto a nivel de hardware como a nivel de empaquetamiento como lo son los contenedores.

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.

4. Descripción

La idea fundamental de la tarea es crear una arquitectura cliente-servidor, en la cual se realicen el envío, procesamiento y almacenamiento de un archivo de texto de cualquier tamaño.

4.1. Servidor

El servidor se ejecutará en una máquina virtual con sistemas operativo basado en Linux, y tiene como función recibir los archivos de manera secuencial, este módulo retornará la cantidad de consonantes que tiene el archivo así como las llamadas al sistemas utilizadas para realizar dicho cálculo (Se sugiere utilizar strace). El servidor deberá ser capaz de rastrear cada llamada al sistema además de la duración de la misma. La respuesta debe ser devuelta al cliente para ser mostrada de manera elegante.

Se deberá crear un archivo de configuración donde se le especifique al servidor el puerto donde debe recibir las consultas y la carpeta donde se deben de almacenar los archivos enviados. El servidor también contará con un archivo donde se registre todos los eventos (log).

El servidor debe iniciar cuando arranca el sistema, similar a un demonio, (asegurándose de incorporar todas las dependencias, especialmente las de red) y la implementación del mismo quedará a diseño de los creadores, puede utilizar SysVinit or Systemd service. 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 DocServer 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. En la figura 1 se muestra un ejemplo de la ejecución de las funciones.

4.2. Cliente

El cliente se debe de ejecutar en un contenedor (se aconseja utilizar docker y una imagen oficial como CentOS) y será el responsable de enviar los documentos de cualquier tamaño y de manera secuencial, hasta que el usuario digite “end”. Además, deberá recibir la respuesta del servidor y mostrarla. Cabe destacar que no se requiere que sea con interfaz gráfica, pero si debe permitir que se configure la ruta del documento que se desea enviar, el puerto y la ip.

En la figura 2 se muestra la arquitectura que se desea implementar con distintos elementos en acción.

4.3. Requerimientos técnicos

- El contenedor se debe crear a partir de una imagen oficial.
- Este proyecto se debe realizar en el lenguaje de programación C. Pueden utilizar las bibliotecas que sean necesarias.
- Debe ser implementado en Linux y se debe proporcionar todos los comandos en el makefile. No se permite realizar configuración alguna una vez ejecutado el contenedor.

```
# service DocServer stop
Stopping DocServer... done

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

#systemctl stop DocServer

# service DocServer restart
Restarting DocServer... done

# /etc/init.d/DocServer stop
Restarting DocServer... done

#systemctl restart DocServer

# service DocServer start
Starting DocServer... done

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

#systemctl start DocServer

# service DocServer status
daemon: DocServer is running (pid 78017)

# /etc/init.d/DocServer status
daemon: DocServer is running (pid 78017)

#systemctl status DocServer
```

Figura 1: Ejemplo de ejecución de las funcionalidades

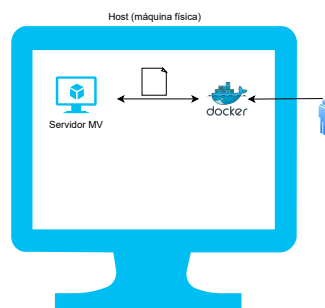


Figura 2: Arquitectura general

- Se permite el uso de dos máquinas separadas o una misma pc donde se ejecute el contenedor y la máquina virtual.

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

- Abstract: Síntesis del proyecto, Generalmente contiene entre 150-200 palabras, debe indicar en qué consiste el proyecto o tarea, cómo se realizó el proyecto o tarea y el principal hallazgo o conclusión.
- Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- Ambiente de desarrollo: Se debe todo lo que se ocupa para ejecutar el proyecto.
- Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto. Para el atributo de de Aprendizaje Continuo se debe mencionar sobre el valor del conocimiento adquirido y las estrategias implementadas para satisfacer las necesidades de aprendizaje. Mientras que para el atributo de Herramientas de Ingeniería se debe mencionar las herramientas ingenieriles que se crearon o que se adaptaron en la solución del problema.
- Diseño, tanto del software como del hardware (en caso de que aplique): Diagramas UML, secuencia, arquitectura, imágenes, descripciones entre otros, todo lo que sea necesario para entender de una mejor manera el diseño y funcionamiento del proyecto. Se espera como mínimo el de arquitectura, componentes, UML y secuencia.
- Instrucciones de cómo se utiliza el proyecto.
- Tabla de actividades: Actividades con el total de horas trabajadas por estudiante.
- Conclusiones: se espera la generalización de los resultados, recordar que las conclusiones no son un resumen de lo que se realizó.
- Sugerencias y recomendaciones.
- Referencias

6. Entregables

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

7. Evaluación

- Cliente en contenedor 20 %
- Integración 10 %
- Rastreador de llamadas 10 %
- Servidor como servicio en MV 30 %
- Contador de consonantes 10 %
- Documentación 20 %

8. Fecha de entrega

- 25 de Agosto. 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.
- En caso de tener alguna duda con algún aspecto de la tarea debe contactar al profesor por cualquier medio de comunicación establecidos.