Nicolás Navarro Garrido
Desarrollo de Software para Syslab

Documentación de Software “Servidor de Recepción de Mensajería Estándar HL7”

El software fue desarrollado con la finalidad de funcionar como un servidor de mensajería, el cual recibe mensajes con un estándar de comunicación HL7, este estándar es utilizado en los equipos informáticos de laboratorios o asociados al área de laboratorio de la medicina. Los mensajes son recibidos y posteriormente se almacenan en un directorio previamente especificado, con los datos de fecha/hora/etc, con extensión (.res), posteriormente envía un mensaje de verificación llamado ACK, en donde estipula la fecha e Id de verificación del mensaje recibido.

Instalación de Python:

Para instalar python 3 en el equipo a utilizar, este debe de ser descargado desde su sitio web oficial (https://www.python.org/downloads/), asegurándose de buscar la versión más reciente, luego de descargar el instalador, este se debe de ejecutar haciendo click sobre él.

Se deben de seguir las instrucciones al pie de la letra, tal como lo mantiene estipulado la pestaña del instalador, antes de la instalación final, se debe seleccionar la opción “Add Python 3.x to PATH”, esta opción sirve para agregarle bibliotecas que no posee por defecto python de una manera mucho más simple y también para poder actualizarlo constantemente.

Para ejecutar el código del servidor, simplemente se debe de hacer click sobre él y aparecerá una ventana tipo consola de comandos, donde se vera la información recibida y relevante.

Funciones Desarrolladas:

- ***manejar_interrupcion(signum, frame):*** Maneja la señal de interrupción (Ctrl+C) y establece la bandera Apagar_Server en True, esta función fue desarrollada con la finalidad de apagar el servidor de manera manual y no solo cerrando el programa.

- ***generar_nombre_archivo():*** Genera un nombre único para los archivos basado en la fecha, hora, minutos, segundos y milisegundos en el cual se solicita la función dentro del programa.

- ***recibir_archivo(conexion, directorio):*** Recibe mensajes HL7 desde un cliente, almacena el mensaje en un archivo y envía un ACK al cliente, esta función recibe como parámetro una conexión socket y un directorio donde se almacenarán los mensajes.

- ***iniciar_servidor(puerto, directorio):*** Inicia el servidor, es decir, abre el puerto correspondiente y se mantiene en un sistema de "escucha" constante hasta que logra recibir una conexión desde un equipo especifico, este al recibir la conexión pasa a ejecutar las funciones previamente mencionadas, tomando en cuenta la prioridad de cada una de estas y el momento adecuado en donde se necesita la información entregada por cada una de ellas.

Variables Globales:

- Apagar_Server (boolean): Indica si el servidor debe detenerse, esta se inicializa en un valor "False", cuando se solicita detener el servidor, esta variable toma el valor "True".

Bibliotecas:

- ***Socket:*** Esta biblioteca es utilizada para acceder a las conexiones de los puertos de datos, siguiendo el estándar del modelo OSI.

- ***Signal:*** Biblioteca necesaria para recibir señales entrantes al dispositivo, utilizada específicamente para finalizar conexión del servidor.

- ***Os:*** Utilizada para acceder a funciones naturales del sistema operativo, en este caso para la lectura y validación de la existencia de directorios y el almacenaje de los mensajes.

- ***From datetime import datetime:*** Esta biblioteca es utilizada para extraer información del reloj interno del procesador, por ejemplo, fecha, hora, minutos, segundos, entre otros. Se usa para dar nombre a los archivos a guardar.

.

- ***Configparser:*** Es utilizada para leer archivos con extensión .ini y acceder a los datos relevantes dentro de ellos, en este caso el puerto y el directorio asignado a la mensajería.

Nicolás Navarro Garrido
Desarrollo de Software para Syslab

Código Fuente del Servidor:

```
import socket #Biblioteca para Trabajar con los Puertos de Conexion

import signal #Biblioteca para Realizar acciones con las señales entrantes al dispositivo

import os #Biblioteca para Acceder a Funciones Naturales del Sistema Operativo

from datetime import datetime #Biblioteca para Extraer datos Temporales del Reloj

import configparser #Biblioteca para Leer Archivos de Extensión .Ini


# Variable global para indicar si el servidor debe detenerse

Apagar_Server = False


def manejar_interrupcion(signum, frame):

    """Maneja la señal de interrupción (Ctrl+C) y establece la bandera Apagar_Server en True.

    Parámetros:

        - signum: Número de la señal.

        - frame: Marco de la señal.

    Retorna: None """


    global Apagar_Server

    print("\n Recibida Señal de Detención. Deteniendo el Servidor...")

    Apagar_Server = True


def generar_nombre_archivo():

    """Genera un nombre único para los archivos basado en la fecha y hora actual.

    Retorna: - str: Nombre del archivo único.    """

    fecha_hora_actual = datetime.now().strftime("%Y%m%d%H%M%S")

    nombre_archivo = f"HL7_ORU_{fecha_hora_actual}"
```

```
    return nombre_archivo


def recibir_archivo(conexion, directorio):
    """ Recibe mensajes HL7 desde un cliente, almacena el mensaje en un archivo y envía un ACK al cliente.
    Parámetros:
        - conexion: Socket de conexión con el cliente.
        - directorio: Directorio donde se almacenarán los archivos.
    Retorna: None"""
    datos = conexion.recv(8192).decode('latin-1')
    if not datos:
        Apagar_Server = True
        print("\n Conexion cerrada", Apagar_Server)
    # Recibir el nombre del archivo
    nombre_archivo = generar_nombre_archivo() + ".res"
    print("\n Archivo en Transmision: ", nombre_archivo)
    if not nombre_archivo:
        return
     # Crear un directorio para cada archivo
    ruta_directorio = os.path.join(directorio).replace('\\','/')
    print("\n Archivo por Guardar en: ", ruta_directorio)
    os.makedirs(ruta_directorio, exist_ok=True) # Verificacion de que el directorio existe
    # Crear y escribir el archivo en el directorio
    ruta_archivo = os.path.join(ruta_directorio, nombre_archivo)
    print("\n Direccion del Archivo a Guardar: ", ruta_archivo)
    text_file = open(ruta_archivo, "w")
    n = text_file.write(datos)
```

```
    text_file.close()

sop = ""

    Auxiliar = datos.splitlines()

    for linea in Auxiliar:

        campo = linea.split('|')

        if campo[0] == 'MSH':

            mensaje_id = campo[6]

            sop = mensaje_id

    Mensaje = (

        f"MSH|^~\\&|ReceivingApp|ReceivingFacility|SendingApp|SendingFacility|"

        f"{datetime.now().strftime('%Y%m%d%H%M%S')}||ACK^R01|{sop}|P|2.5||||||latin-1|||NE\r"

        f"MSA|AA|{sop}|\r")


    conexion.sendall(Mensaje.encode('latin-1'))

    print(Mensaje, ":", nombre_archivo)

def iniciar_servidor(puerto, directorio):

    """Inicia el servidor, configurando el manejador de señales y esperando conexiones entrantes.

    Parámetros:

        - puerto: Número de puerto para escuchar.

        - directorio: Directorio donde se almacenarán los archivos.

    Retorna: None"""

    global Apagar_Server

    # Configurar el manejador de señales para Ctrl+C

    signal.signal(signal.SIGINT, manejar_interrupcion)

    # Crear un socket IPv4 y TCP

    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as servidor:
```

```
        # Enlazar el socket a la dirección y puerto especificados
        servidor.bind(('0.0.0.0', puerto))
        print("Servidor Escuchando el Puerto", puerto)
        # Establecer el servidor en modo de escucha
        servidor.listen()
        conexion, direccion_cliente = servidor.accept()
        print("Conexión Establecida desde", direccion_cliente)
        while not Apagar_Server:
            try:
                # Iniciar la recepción de archivos en el directorio especificado
                recibir_archivo(conexion, directorio)
            except socket.error as e:
                print("Error en la Conexion: ", e)
                Apagar_Server = True
if __name__ == "__main__":
    Parametros = configparser.ConfigParser()
    Parametros.read('Config.ini')
    Puerto = int(Parametros.get('Servidor', 'Puerto'))
    Directorio = Parametros.get('Servidor', 'Directorio')
    iniciar_servidor(Puerto, Directorio)
```