Curso de Interoperabilidad entre Sistemas de Información en Salud

Tarea 2

Objetivo: probar la comunicación MLLP/HL7 con HAPI.

Planteo: se plantea crear un cliente y un servidor MLLP para comunicar mensajes HL7 v2.5.

Entrega: **¡leer atentamente!**

1. La entrega debe incluir el código fuente, el binario compilado, documentación de la solución propuesta, problemas que encontró y cómo los solucionó.
2. La documentación debe incluir las instrucciones exactas para compilar y ejecutar él o los programas o scripts entregados. Si no se incluyen, **no se podrá evaluar la tarea**.
3. Si utiliza librerías o código de otros proyectos, debe agregar una referencia a qué proyectos y qué versiones de los mismos está utilizando. Siempre debe acreditar el trabajo de otros colegas.
4. Seguir la estructura de proyecto en carpetas:
    a. /src < código fuente
    b. /lib < librerías externas y dependencias
    c. /bin < binarios compilados
5. **Entregar proyecto en archivo ZIP**
    a. **Nombre: "nombre_apellido_tarea_2.zip"**
6. Se sugiere utilizar Java o Groovy para el desarrollo. Se aceptarán entregas en otros lenguajes de programación mientras incluyan instrucciones para correr las soluciones, incluyendo construcción del entorno y dependencias de librerías externas.
7. Incluir las capturas de paquetes de WireShark* en formato **pcapng**.
    a. pcapng es un formato de texto que permite expresar información de tráfico en redes: https://www.winpcap.org/ntar/draft/PCAP-DumpFileFormat.html

Plazo: ver en campus virtual de ACHISA

Puntaje máximo: 20 / 100

* Si tiene problemas con la captura de paquetes locales en máquinas Windows, tiene dos opciones: 1. crear una máquina virtual y comunicar el host con la virtual, 2. probar con RawCap http://www.netresec.com/?page=RawCap (RawCap genera archivos pcap que pueden abrirse con Wireshark para analizar). Ver: http://carminedimascio.com/2014/03/rawcap-and-wireshark-how-to-capture-and-analyze-local-traffic-from-host-machine-to-itself/

Planteo:

Implementar un Servidor y un Cliente MLLP utilizando HAPI. Cliente y Servidor deben ser dos aplicaciones separadas.

Descripción del flujo básico:

1. El cliente debe generar mensajes ADT^A01 de HL7 v2.5, con datos para los segmentos MSH y PID (cómo mínimo), creando el mensaje mediante la API de HAPI (código visto en clase, no se permite crear el mensaje como un String)
    a. Datos para MSH: caracteres de codificación, aplicación y centro emisor, aplicación y centro receptor, fecha y hora del mensaje, tipo de mensaje y evento, id de control, id de procesamiento y versión de HL7.
    b. Datos para PID: un identificador, nombre, apellido, fecha de nacimiento y sexo.
2. El cliente debe imprimir en la consola el valor del segmento MSH-10 (Message Control ID).
3. El servidor debe esperar por conexiones MLLP en un puerto configurable, esperando recibir ADT^A01.
4. El cliente debe enviar cada mensaje al puerto donde espera el servidor.
5. El servidor debe recibir y procesar cada mensaje con HAPI, e imprimir los valores de cada campo en la consola, indicando qué campo es, ej. PID-5-1 (apellido) = Pazos, PID-5-2 (nombre) = Pablo, etc., incluyendo el MSH-10 Message Control ID.
6. El servidor debe generar un mensaje ACK para el ADT recibido, y debe ser enviado al cliente.
7. El cliente debe procesar cada mensaje ACK recibido, también con HAPI. e imprimir los valores de los campos del segmento MSA (segmento de acknowledgment) en consola.
    a. Recordar: si el cliente envía más de un mensaje, puede recibir los ACK en desorden, por ejemplo: envía ADT 1, 2, 3 recibe ACK 2, 1, 3.
8. El cliente debe cerrar la conexión.
9. El servidor no debe fallar cuando el cliente cierra la conexión.
10. El servidor debe seguir escuchando por conexiones cuando el cliente cierre la conexión.

Requerimientos:

1. El servidor debe aceptar la conexión de varios clientes simultáneos (loop con accept).
    a. Debe probarlo levantando varios clientes y conectándolos al mismo servidor.
2. El servidor debe aceptar la recepción de varios mensajes del mismo cliente (loop de receive / read).
    a. Debe ser probado haciendo que cada cliente genere y envié varios ADT distintos (controle que el Message Control ID de cada mensaje es distinto).
3. Probar cliente y servidor con HAPI Test Panel y documentar pruebas.
    a. Adjuntar capturas de pantalla en la documentación y con sus respectivas descripciones.

4. Monitorear comunicaciones con Wireshark* (direcciones IP, puertos, protocolos utilizados, mensajes intercambiados, etc.) y adjuntar captura del tráfico, señalando cuáles paquetes corresponden a las comunicaciones entre su servidor y su cliente.
    a. Se recomienda adjuntar capturas de pantallas de Wireshark a la documentación y describir ahí qué es lo que está pasando con las comunicaciones.
    b. Salvar la captura de paquetes en el formato de Wireshark (pcapng)

* Si tiene problemas con la captura de paquetes locales en máquinas Windows, tiene dos opciones: 1. crear una máquina virtual y comunicar el host con la virtual, 2. probar con RawCap http://www.netresec.com/?page=RawCap (RawCap genera archivos pcap que pueden abrirse con Wireshark para analizar). Ver: http://carminedimascio.com/2014/03/rawcap-and-wireshark-how-to-capture-and-analyze-local-traffic-from-host-machine-to-itself/