Álvaro Aledo Tornero Y Antonio Vergara Moya

48129547T y 48851755P

Descripción breve

Documento entregable con información útil para la implementación, entendimiento y evaluación de la tarea práctica de la asignatura Redes de Comunicaciones

Nanofiles: Documentación de diseño

(1904) REDES DE COMUNICACIONES [23/24]

Profesor: Epifanio Gaona Ramírez

Índice

**Introducción.**

En este documento se especifica el diseño de los protocolos TCP y UDP para la comunicación entre procesos *Nanofiles* entre ellos mismos y un proceso servidor *Directory*.

Las mejoras que actualmente tenemos planeadas son todas las disponibles en el proyecto, deteniéndonos en el momento en el que nuestro proyecto esté lo suficientemente maduro como para tener una buena calidad y funcionalidad.

**Formato de los mensajes del protocolo de comunicación con el Directorio**

Para definir el protocolo de comunicación con el *Directorio*, vamos a utilizar mensajes textuales con formato “campo:valor” que implementaremos mediante la creación de la clase DirMessage. El valor que tome el campo “operation” (código de operación) indicará el tipo de me nsaje y por tanto su formato (qué campos vienen a continuación).

Mensajes de solicitud

**login**

Descripción: Este mensaje lo envía el cliente de NanoFiles al Directorio para solicitar “iniciar sesión” y registrar el nickname indicado en el mensaje.

Ejemplo:

operation: login\n

nickname: alicia\n

\n

**logout**

Descripción: Este mensaje lo usa el cliente NanoFiles con una sesión iniciada para indicarle al Directorio que quiere cerrar su sesión actual.

Ejemplo:

operation: logout\n

sessionKey: 1234\n

\n

**userlist**

Descripción: Mensaje disponible para los clientes NanoFiles que solita al Directorio la lista de los usuarios que están conectados en el momento del procesamiento de la solicitud.

Ejemplo:

operation: userlist\n

sessionKey: 1234\n

\n

Mensajes de respuesta

**loginok**

Descripción: Este mensaje lo manda el Directorio en respuesta a un mensaje de operación login cuando el login ha sido exitoso, adjuntándole la sessionKey correspondiente al usuario.

Ejemplo:

operation: loginok\n

sessionKey: 1234 \n

\n

**login\_failed**

Descripción: Este mensaje se manda cuando la operación de login no se ha podido realizar. Se suele usar cuando un usuario que ya tiene una sesión iniciada intenta volver a iniciar sesión.

Ejemplo:

operation: login\_failed\n

\n

**logoutok**

Descripción: Este mensaje se manda cuando se ha cerrado sesión exitosamente.

Ejemplo:

operation: logoutok\n

\n

**logout\_failed**

Descripción: Este mensaje se manda cuando no se ha podido cerrar sesión.

Ejemplo:

operation: logout\_failed\n

\n

**userlistok**

Descripción: Este mensaje se usa para mandar la lista de usuarios al cliente nanofiles.

Ejemplo:

operation: userlistok\n

users: “usuario1, usuario2 ,…”\n

\n

**userlist\_failed**

Descripción: Este mensaje lo manda Directory para indicar al usuario que ha fallado la operación de userlist.

Ejemplo:

operation: userlist\_failed\n

\n

**Formato de los mensajes del protocolo de transferencia de ficheros**

Para definir el protocolo de comunicación con un servidor de ficheros, vamos a utilizar mensajes binarios multiformato. El valor que tome el campo “opcode” (código de operación) indicará el tipo de mensaje y por tanto cuál es su formato, es decir, qué campos vienen a continuación.

**Tipos y descripción de los mensajes**

Mensaje: **FILE\_NOT\_FOUND (opcode = 1)**

Sentido de la comunicación: Servidor de ficheros → Cliente

Descripción: Este mensaje lo envía el par servidor de ficheros al par cliente (receptor) de fichero para indicar que no es posible encontrar el fichero con la información proporcionada en el mensaje de petición de descarga.

Ejemplo:

|  |
| --- |
| **Opcode**  **(1 byte)** |
| **-1** |

Mensaje: DOWNLOAD

Sentido de la comunicación: Cliente -> Servidor de ficheros

Descripción: Este mensaje se envía para solicitar la descarga de un fichero.

Ejemplo:

|  |  |
| --- | --- |
| Opcode  (1 byte) | HashFile  (4 bytes) |
| 1 | 1234 |

Mensaje: DOWNLOAD\_FROM

Sentido de la comunicación: Cliente -> Servidor de ficheros

Descripción: Este mensaje se usa para descargar un fichero de un usuario en concreto en caso de que este exista.

Ejemplo:

|  |  |  |
| --- | --- | --- |
| Opcode  (1 byte) | HashFile  (4 bytes) | serverNick  (1-n bytes) |
| 2 | 1234 | “server1” |

Mensaje: FILELIST

Sentido de la comunicación: Cliente -> Servidor de ficheros

Descripción: Este mensaje se usa para solicitar la lista de ficheros disponibles para descargar.

Ejemplo:

|  |
| --- |
| Opcode  (1 byte) |
| 3 |

Mensaje: DOWNLOAD\_OK

Sentido de la comunicación: Servidor de ficheros -> Cliente

Descripción: Este mensaje se usa para indicar que la solicitud de descarga ha sido aceptada. Tiene adjuntado el fichero.

|  |  |  |
| --- | --- | --- |
| Opcode  (1 byte) | fileLength  (4 bytes) | File  (fileLength bytes) |
| 4 | 5 | file |

Mensaje: FILELIST\_OK

Sentido de la comunicación: Servidor de ficheros -> Cliente

Descripción: Este mensaje se usa para indicar que la solicitud de la lista de usuarios ha sido exitosa. Contiene la propia lista.

|  |  |
| --- | --- |
| Opcode  (1 byte) | fileList  (1-n bytes) |
| 5 | “fichero 1”, “fichero 2”, …. |

**Autómatas de protocolo**

Autómata rol Nanofiles

Diagrama

Descripción generada automáticamente

Autómata rol Directorio

Diagrama

Descripción generada automáticamente

**Ejemplo de intercambio de mensajes**

Incluir en esta sección ejemplos de “conversaciones” ficticias (con valores inventados) haciendo uso de los mensajes definidos en las secciones anteriores y comentando cómo el autómata restringe qué mensaje(s) puede enviar recibir cada extremo de la comunicación en cada instante de la conversación (estado del autómata).

CLIENTE: inicia sesión

operation: login

nickname: alumno

DIRECTORIO: deniega inicio de sesión

…

CLIENTE: inicia sesión

operation: login

nickname: alicia

DIRECTORIO: confirma inicio de sesión

…

CLIENTE: pide la lista de usuarios

…