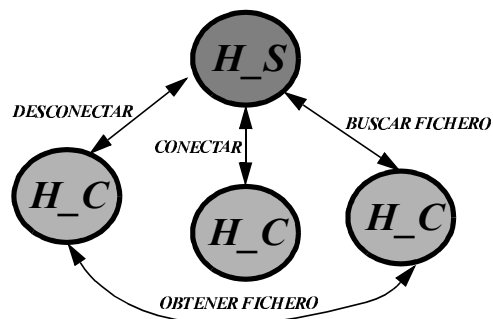


Hamster

La práctica consiste en la realización de una aplicación que permita a usuarios de diferentes máquinas intercambiar ficheros. Dado que vamos a trabajar con protocolos TCP/IP, la realización de la práctica se hará utilizando la *interfaz socket*. Con esta práctica se pretende que el alumno asimile los conceptos de modelo cliente-servidor, servicio orientado a la conexión y no orientado a la conexión, servidor concurrente y servidor interactivo así como una familiarización con la interfaz de sockets de la capa de transporte.

La aplicación que se pretende realizar, se denominará *Hamster*. Esta aplicación ha de permitir a diferentes usuarios, poder intercambiar ficheros. Para ello, se dispondrá de un servidor Hamster que actuará de intermediario manteniendo información de los usuarios que en ese momento están conectados, donde están ubicados y de los ficheros que estan dispuestos a compartir.



1. Hamster_Server y Hamster_Client

La aplicación Hamster constará de los dos programas necesarios para poder permitir a diferentes usuarios compartir ficheros. Cada uno de los usuarios ejecutará un programa cliente denominado Hamster_Client y a través de la aplicación servidor Hamster_Server obtendrán información de los diferentes usuarios que en ese momento están conectados y de los ficheros que están dispuestos a compartir. Una vez un usuario obtenga la ubicación del fichero que desea obtener a través de la información que le proporciona el servidor, se conectará al usuario

determinado para obtener dicho fichero directamente. En la figura se muestran las posibles acciones que se pueden realizar en el entorno, y a continuación se detallan los nombres y parámetros del cliente y del servidor.

Nombre Servidor: Hamster_Server
Argumentos: puerto

Nombre Cliente: Hamster_Client
Argumentos: nombre_host_servidor puerto_servidor

Las decisiones de implementación, especificación de los diálogos entre el cliente y el usuario así como del tipo de servidor (interactivo, concurrente o ambos) y servicio de comunicación (orientado, no orientado o ambos) quedan a cargo del alumno.

2. Requerimientos Mínimos

A continuación se detallan las funcionalidades mínimas que ha de proporcionar la infraestructura.

- Se utilizará el lenguaje de programación C.
- El usuario al conectarse por primera vez al servidor ha de proporcionar su nombre. El servidor no puede permitir que se creen dos usuarios con el mismo nombre de usuario.
- El usuario al conectarse al servidor, le proporciona un listado de los ficheros que desea compartir, su ubicación (máquina y puerto) y posteriormente, acaba la conexión con el servidor. A partir de ese momento, el usuario se queda a la espera de recibir peticiones de ficheros de otros usuarios.
- El usuario realiza búsquedas de ficheros a través de la base de datos de nombres que mantiene el servidor. El servidor no ha de tener los ficheros, solo información de clientes conectados y de sus ficheros.
- El servidor proporcionará la información necesaria a un usuario para que pueda conectarse a otro usuario y obtener el fichero deseado. Esta conexión es directa entre usuarios y no ha de actuar el servidor como intermediario en el intercambio.
- Ha de ser posible que un usuario pueda obtener varios ficheros a la vez de diferentes usuarios, y que varios usuarios obtengan ficheros al mismo tiempo del mismo usuario.
- Un usuario avisa al entorno que no va a proporcionar mas ficheros, en el momento que envía una solicitud de desconexión al servidor. A partir de ese momento, el usuario deja de ser visible al resto de usuarios.
- La infraestructura ha de ser lo más robusta a fallos posible.

3. Notaciones

- Las prácticas se realizarán individualmente o en grupos de dos personas como máximo.
- Respetar los nombres y argumentos especificados de cada uno de los programas.
- Se realizará una entrevista y prueba de la aplicación con todos los miembros del grupo.
- Mejoras y funcionalidades adicionales se tendrán en cuenta en la nota final.

4. Documentación a entregar.

- Diskette con los archivos fuentes y ejecutables. (libre de virus !!).
- Informe con las decisiones de diseño y los programas en alto nivel debidamente comentados.
- Se han de especificar claramente todos los tipos de comunicaciones y dialogos que se producen entre servidor-cliente y entre cliente-cliente justificando las decisiones.
- El tipo de servidor así como los diferentes procesos que intervienen en cada programa tambien se han de comentar.