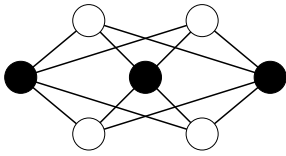


E52. Redes Informáticas



Práctica 8. El nivel de aplicación en Internet: Aplicaciones sobre sockets

Descripción del equipo y del software

1. Ordenadores de tipo PC con S.O. Linux.
2. Adaptador de red *Ethernet* con conector RJ-45 (estándares 10BaseT/100BaseT).
3. Conmutador 10BaseT.
4. Compilador *gcc* de C.

Nota:

Esta práctica corresponde con el final de la asignatura y como trabajo voluntario para la evaluación de la parte de programación con sockets. Las personas que decidan realizar el trabajo deberán haberlo concluido antes de la última sesión de laboratorio. En esa sesión se realizará la evaluación. A las personas que opten por la opción de trabajo se les requerirá que durante esa última sesión realicen una pequeña modificación sobre él. A las personas que obtengan la evaluación (sin trabajo) durante esa sesión se les requerirá que desarrollen un programa utilizando sockets que se les indicará al inicio de la sesión.

1. Desarrollo de la práctica

En la presente práctica se desarrollará un sistema de acceso a ficheros remotos. El sistema de acceso únicamente permitirá **operaciones asíncronas de lectura**. Para ello, de cara al usuario final se deberán proporcionar dos funciones básicas:

```
int init_read(char * file_name, int pos, int n_bytes);
```

donde se indica el nombre del fichero remoto en el que se debe realizar la lectura (`file_name`), la posición inicial a partir de la que realizar la operación y el número de bytes a leer. Una vez realizada la operación se puede seguir ejecutando código. El resultado de la función tendrá valor negativo en caso de producirse algún error (se especificarán los posibles errores más adelante) o un valor mayor o igual a cero que se tomará como identificador de la operación de lectura, y que se utilizará para la finalización de dicha operación. Cuando se desee recuperar los datos de la operación iniciada se deberá utilizar la función

```
int finalize_read(int read_id, void * data);
```

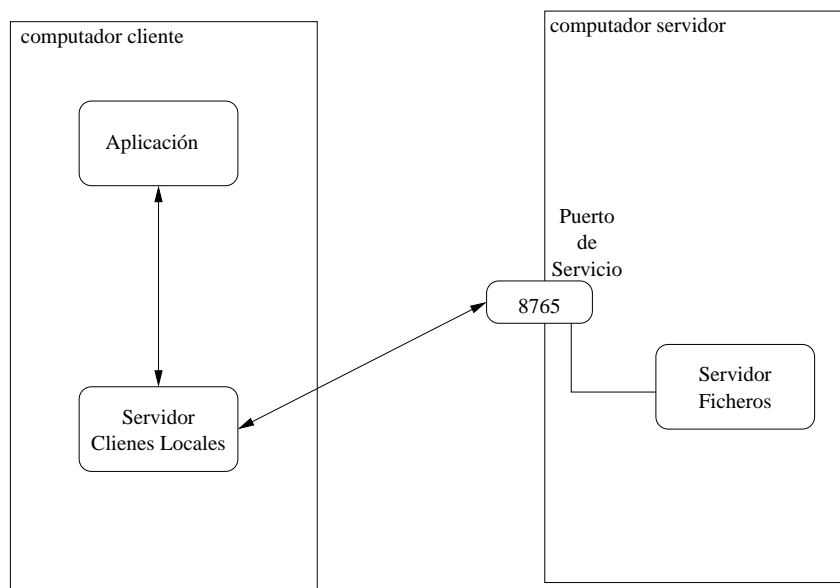


Figura 1: Lectura y escritura de un fichero mediante TFTP.

en donde el primer argumento debe ser un identificador válido obtenido anteriormente mediante una llamada a la función `init_read`. El segundo argumento indica la posición de memoria en la que se deben dejar los datos leídos. Como resultado devolverá un número negativo en caso de que se haya producido alguna situación de error, y un número mayor o igual a 0, indicando el número de bytes leídos, en caso contrario.

El esquema de la aplicación será el que se muestra en la figura 1. En dicha figura se aprecia que existen 3 aplicaciones diferentes:

- La aplicación cliente que se la que hará uso de las rutinas antes indicadas para realizar el acceso a los ficheros remotos.
- El *demonio local* que escuchará las peticiones de las aplicaciones locales para realizar las peticiones a los servidores.
- El *servidor remoto* que será el encargado de dar servicio a las peticiones que provienen de los demonios locales de las máquinas donde se ejecutan los clientes.

Las rutinas `init_read` y `finalize_read` se comunican con el *demonio local* mediante sockets UNIX, para realizar las peticiones y para recoger los resultados. La comunicación entre estas dos aplicaciones se realizarán siguiendo el siguiente protocolo.

- Para el **inicio de lectura**, mediante la rutina `init_read`, el cliente enviará un mensaje con el siguiente formato:
 - a) Dos caracteres '01' para indicar el tipo de mensaje.
 - b) Una cadena de 8 caracteres que identifica el nombre del fichero.
 - c) Una cadena de 6 caracteres que identifica el punto de inicio de la operación de lectura. Por ejemplo "000123" para la posición 123.

- d) Una cadena de 3 caracteres que identifica el número de caracteres a leer. Por ejemplo "012" para 12 caracteres.

Una vez recibido este mensaje, el cliente recibirá un mensaje con el siguiente formato:

- a) Dos caracteres '02' para indicar el tipo de mensaje.
- b) La cadena de caracteres '00' (cero cero) en caso de que la lectura se acepte, o la cadena de caracteres '01' (cero uno) en caso de que el número máximo de lecturas que en ese momento se estén realizando superen el máximo MAX_SREAD (100).
- c) En caso de tener como segundo campo '00', se enviará una cadena de dos caracteres, que indicarán un número entre 00 y 99 que servirá de identificador de la operación de lectura que se ha asignado.

Después de recibir este mensaje, la rutina `init_read` devolverá como resultado un número negativo que indicará si se ha producido alguna situación de error. Las posibles situaciones de error que se deben tener en cuenta son:

1. No se puede conectar con el demonio local, en cuyo caso se devolverá el valor -1.
 2. El servidor no acepta más lecturas por el momento, en cuyo caso se devolverá el valor -2.
- Cuando el demonio local ha aceptado una operación de lectura debe comunicarse con el servidor remoto. Lo hará con el siguiente protocolo de mensajes.
 - a) Un primer mensaje de petición que tendrá el siguiente formato:
 - Dos caracteres '03' para indicar el tipo de mensaje.
 - La cadena de 8 caracteres que identifica el nombre del fichero.
 - La cadena de 6 caracteres que identifica el punto de inicio de la operación de lectura.
 - La cadena de 3 caracteres que identifica el número de caracteres a leer.
 - b) El servidor remoto contestará con un mensaje que contenga el resultado de la operación de lectura. Este mensaje tendrá el siguiente formato:
 - Dos caracteres '04' para indicar el tipo de mensaje.
 - La cadena de caracteres '00' (cero cero) indicando que la operación se ha realizado con éxito, o la cadena de caracteres '01' (cero uno) para indicar que se ha producido un error porque no se ha podido acceder al fichero solicitado.
 - Una cadena de 3 caracteres indicando el número de bytes que se han leído.
 - El total de bytes leídos.
 - Cuando el cliente desea finalizar la operación de lectura, enviará un mensaje al demonio local. Dicho mensaje tendrá el siguiente formato:
 - Dos caracteres '05' para indicar el tipo de mensaje.
 - La cadena de dos caracteres indicando la operación de lectura de la que se requiere el fin, y que ha sido obtenida en el inicio de la operación.
 - Como consecuencia del mensaje anterior, el demonio local devolverá al usuario un mensaje que tendrá en siguiente formato:

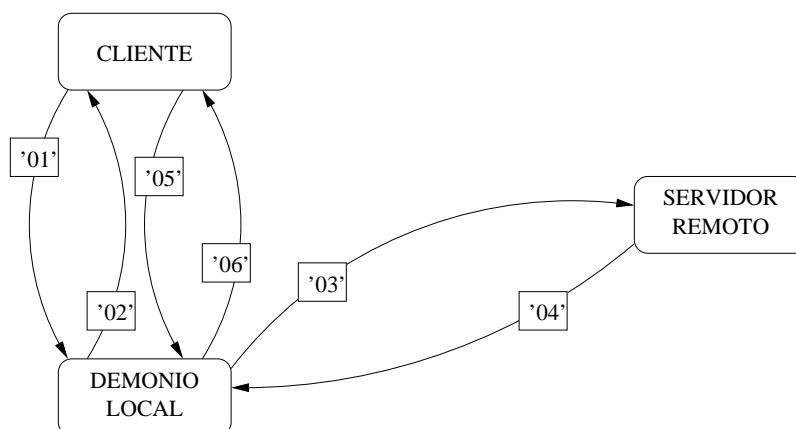


Figura 2: Secuencia de mensajes para una operación de lectura

- Dos caracteres '06' para indicar el tipo de mensaje.
- La cadena de caracteres '00' (cero cero) indicando que la operación se ha realizado con éxito, o la cadena de caracteres '01' (cero uno) para indicar que se ha producido algún error.
- Si la segunda cadena indicada en este mensaje es '00' se enviará una tercera cadena de 3 caracteres que indicará el número de bytes leídos, y por último el conjunto de bytes leídos.
- Si la segunda cadena indicada en este mensaje es '01' se enviará una tercera cadena de caracteres, finaliza con el caracter NULL, que indicará el tipo de error que se ha detectado.

La secuencia completa de mensajes que se enviarán entre las distintas aplicaciones se puede observar en la figura 2. Para la realización de las distintas aplicaciones se debe tener en cuenta:

- 1) La comunicación entre el cliente y el demonio local se realizará mediante sockets UNIX, con sockets del tipo SOCK_DGRAM, estando el demonio local escuchado a través de un socket conectado con el fichero `/tmp/prac8E52`.
- 2) El demonio local, cuando inicie su ejecución leerá del fichero `/${PWD}/default_server` el nombre de la máquina donde se encuentra el servidor remoto.
- 3) El puerto donde está escuchado el servidor remoto es el 8765.
- 4) Los ficheros a los que se tiene acceso se encuentran siempre en el directorio `/${PWD}` del servidor.
- 5) El máximo número de bytes que pueden leerse en una única operación de lectura es de 256 bytes.