

Ejercicio 2

El servidor espera peticiones de conexión de clientes en `Accept()`. Si llega una, lanza un thread con los parámetros necesarios para la comunicación. Sino se bloquea hasta que llega una petición. El servidor termina su ejecución cuando se han aceptado 10 clientes.

Ejercicio 3

El cliente envía un mensaje que contiene: la url de la imagen + "ventana" + el numero de la ventana en la que quiere mostrar la imagen (0, 1 o 2, siendo 0 la ventana grande) + "duracion" + tiempo durante el que quiere mostrar la imagen. "ventana" y "duracion" son los delimitadores que buscara el servidor. Si el servidor acepta la petición recibirá una respuesta con el formato: el precio + "restante" + tiempo restante hasta mostrar el anuncio + "cola" + numero de anuncios antes del suyo en espera. Si no acepta la petición el servidor, envía la respuesta "rechazado" y el cliente hace un segundo intento. Finalmente el cliente envía el mensaje de fin de conexión al servidor para indicar que termina.

El servidor lanza un thread por cada accept para atender a los clientes. Recibe la petición de ventana y duración de cada cliente. Para almacenar las peticiones se utilizan dos colas. Como son recursos compartidos se utilizan dos monitores para acceder secuencialmente a estas. EL servidor llama a la función solicitar del monitor con los parámetros a almacenar. Si no se almacenan correctamente devuelve un precio igual a -1. El servidor devuelve como respuesta a cada cliente el precio, el tiempo restante aproximado y su posición en la cola. Realiza esta operación hasta que el cliente indica que termina el servicio.

Los procesos ventana llaman a la función del monitor mostrar, ya que acceden y modifican el recurso compartido cola. Esta devuelve la duración y la url de la imagen. El proceso se encarga de descargar la imagen y de mostrarla en la ventana correspondiente. Los procesos realizaran esta operación mientras el tamaño de la cola sea mayor que cero y mientras no se admitan mas clientes. Para indicar que no se admiten mas clientes el servidor utiliza el booleano compartido "fin" que pasa a ser true cuando han llegado el limite de clientes al servidor. Para evitar el bloqueo de las ventanas el servidor llama a la función del monitor signal ventanas que despierta a las ventanas que se encontraban en wait una vez a terminado el programa.

El servidor también puede terminar si se ejecuta un cliente con una cadena de caracteres como cuarto parámetro. En ese caso, el cliente envía una señal al servidor indicando que termine su servicio. Es necesario repetir este proceso una segunda vez porque el antes de recibir el mensaje del cliente el servidor vuelve a disponerse a aceptar otro cliente.

Dificultades: Acceder de forma excluyente a las colas, sincronizar el fin de programa con las ventanas además de evitar su bloqueo en algunos casos e indicar el fin del servicio del servidor.