

Arquitectura de Sistemas de Información

Grado en Ingeniería en Tecnología de Telecomunicación. 3º curso.

Prueba práctica

16 de junio de 2015

Tiempo total: 2 h.

Descripción de la prueba práctica

La práctica consistirá en un trabajo incremental, en el cual el monitor va a comunicarse y sincronizarse con una jerarquía de procesos “client” a través de diversos mecanismos de comunicación y sincronización (sockets, memoria compartida y semáforos).

El ejercicio se desarrollará progresivamente, añadiendo en cada paso nuevas funcionalidades a lo que se vaya desarrollando, manteniendo la compatibilidad hacia atrás. Esto es, el ejercicio 2 será una ampliación del ejercicio 1. Así sucesivamente hasta que el ejercicio 4 implemente todas las funcionalidades.

Inicialmente (primer ejercicio) el alumno deberá crear un proceso servidor concurrente que creará los procesos necesarios para atender las distintas conexiones que irá abriendo el proceso monitor. Se procurará, en la medida de lo posible, que los procesos hereden de sus padres los recursos necesarios (sockets, conexiones, memorias compartidas, semáforos). Sobre ese servidor concurrente se implementarán algunos comandos simples, que se probarán de forma concurrente en el ejercicio 2.

En una siguiente fase (ejercicio 3) se implementarán una serie de contadores estadísticos en un segmento de memoria compartida, así como algunos comandos para su consulta.

Finalmente (ejercicio 4), se controlará mediante semáforos el número de procesos concurrentes lanzados y se limitará, también mediante semáforos, el número máximo de procesos que pueden realizar operaciones simultáneamente y el acceso a la memoria compartida.

El programa monitor crea un segmento de memoria compartida de 1024 bytes, con una clave de acceso obtenida a partir del DNI del alumno sin letra (Ej: 11223344) expresada como un *long* en hexadecimal (Ej: 0x11223344L). Ese segmento será compartido entre todos los procesos cliente y el Monitor para intercambiar datos y cálculos.

También crea un array de 3 semáforos con la misma nomenclatura. El primero servirá para contar los procesos creados para atender conexiones, el segundo de ellos se utilizará para limitar el número máximo de procesos realizando operaciones y el último, para garantizar la exclusión mutua en el acceso a memoria compartida.

Ejercicio 1

En este ejercicio se espera que el alumno prepare la recepción de conexiones a un servidor TCP concurrente en el puerto 3010. El proceso que atiende cada conexión debe enviar un mensaje de login "login:" (sin las comillas) de bienvenida y esperar la identificación del usuario. Antes de comprobar esa identificación, el servidor debe enviar un mensaje de password "passwd:" y esperar la recepción del mismo. Todas las cadenas recibidas estarán finalizadas por un '\n'(y no contendrán '\0').

Trabajaremos con un único usuario válido de login *anonymous* y password *guest*. Si no se introduce un usuario válido, el servidor volverá a solicitar el login y password hasta 3 veces máximo. Al tercer intento fallido se desconectará la conexión en curso.

Si se obtiene una identificación adecuada el servidor presentará un prompt identificativo de la sesión "*usuario:\$*" (Ej. *anonymous:\$*) y se quedará pendiente de recibir comandos. Durante la sesión de comunicación se atenderán un número indeterminado de comandos, y sólo se saldrá de la sesión a solicitud del cliente, terminándose entonces el proceso de atención de esa conexión. Estas desconexiones se podrán recibir en cualquier momento y deberán ser atendidas correctamente.

El monitor realizará varias conexiones y desconexiones al servicio. Si se atiende la solicitud de una conexión enviándole el login se desvelará la clave <1>. Si se procesa adecuadamente una petición de conexión del usuario *anonymous/guest* presentando el prompt se dará la clave <2>. Si se rechazan correctamente las solicitudes mal realizadas durante tres intentos se desvelará la clave <3>.

Ejercicio 2

Las sesiones de los usuarios en el servidor, esperarán órdenes tipo comando (siempre terminadas con el carácter '\n'). La lista de comandos y sus respuestas son:

- `whoami` → *nombre del usuario*\n Ej. *anonymous*\n
- `pid` → *identificador de proceso*\n Ej. *1234*\n
- `quit` → se finalizará la conexión
- `bad command` → se responderá con `Error`\n

Además de los comandos anteriores, en este ejercicio se examinará la concurrencia de sesiones, para lo cual el Monitor establecerá varias sesiones y controlará que se crean los procesos adecuados que las atienden. Si es así, dará la clave <4>.

En cada sesión se podrá enviar uno o varios comandos que deben ser respondidos adecuadamente. En ese caso se proporcionará la clave <5>. También se procederá a la desconexión de sesiones vía el comando *quit* o forzando la desconexión del socket. El servidor no debe dejar procesos zombies. Si se hace bien se dará la clave <6>

Ejercicio 3

Se almacenará en memoria compartida alguna información estadística. El segmento de memoria tendrá un tamaño de 1024 bytes y como clave la parte numérica del DNI del alumno expresada en hexadecimal. (Ej. DNI=12345678H →clave=0x12345678L).

Al principio de esa memoria se situará una tabla de cuatro enteros que serán contadores. El 1º contará el número acumulado de conexiones que recibe el servidor. El 2º el nº acumulado de sesiones que han conseguido identificarse adecuadamente. El 3º cada uno de los intentos de identificación fallidos. El 4º el número de sesiones activas recibiendo y procesando comandos.

El programa monitor inicializará los contadores de estadísticas e intentará posteriormente varias conexiones exitosas y fallidas con diferentes usuarios. El servidor debe actualizar la información estadística.

Además, el servidor responderá a los siguientes comandos:

- `ncnx` → *número de conexiones recibidas*\n Ej. 5\n
- `nsess` → *número de sesiones establecidas*\n Ej. 6\n
- `nok` → *número de sesiones mal identificadas*\n Ej. 7\n
- `active session` → *número de sesiones activas*\n Ej. 8\n

Si se actualizan adecuadamente los contadores en la memoria se dará la clave <7> (todos los contadores deben estar bien). Si se responde adecuadamente a todos los comandos se dará la clave <8>. Si se responde correctamente manteniendo la concurrencia de procesos se dará la clave <9>.

Ejercicio 4

Se introduce en este apartado un control semaforizado para controlar el número de conexiones activas, el máximo de sesiones activas posibles, y el acceso a los datos de memoria compartida.

Para ello se utiliza un array de tres semáforos, donde el primero se utilizará como un contador de conexiones activas, el segundo lo inicializará el monitor a 5 para controlar un máximo de 5 sesiones activas y el tercero se utilizará para controlar la lectura/escritura sobre el array de enteros al principio del segmento de memoria compartida.

Se entiende por sesión activa las cinco primeras que han superado la fase de login (y enviado el primer prompt para indicarlo al cliente) y están a la espera de comandos. Las conexiones en curso pueden alcanzar la fase de login pero si ya hay 5 sesiones activas, tendrán que esperar a que finalice una de las activas para poder responder a los comandos.

La clave <10> se obtendrá si se usa adecuadamente el contador de conexiones activas. El control adecuado del límite de sesiones activas servirá para obtener la clave <11> y el control en la actualización de los contadores generales a la hora de responder a los comandos dará la clave <12>

.

Arquitectura de Sistemas de Información

Grado en Ingeniería en Tecnología de Telecomunicación. 3º curso.

Prueba práctica

16 de junio de 2015

Tiempo total: 2 h.

Nombre:.....

Grupo:..... DNI:.....

Aula:..... Fila:..... Columna:.....

Ejercicio 1

Clave 1:	Clave 2:	Clave 3:
----------	----------	----------

Ejercicio 2

Clave 4:	Clave 5:	Clave 6:
----------	----------	----------

Ejercicio 3

Clave 7:	Clave 8:	Clave 9:
----------	----------	----------

Ejercicio 4

Clave 10:	Clave 11:	Clave 12:
-----------	-----------	-----------