

SISTEMAS DISTRIBUIDOS

GRADO EN INGENIERÍA INFORMÁTICA



Diseño e implementación de un sistema peer-to-peer. Partes 2 y 3

Félix GARCÍA CARBALLEIRA
Francisco Javier GARCÍA BLAS

1 de abril de 2020

Índice

1. Objetivo	2
2. Parte 2 (RPC)	2
2.1. Descripción de la funcionalidad	2
2.2. Lenguajes de programación a utilizar	3
3. Parte 3 (Servicios Web)	3
3.1. Descripción de la funcionalidad	3
3.2. Lenguajes de programación a utilizar	3
4. Normas generales	4
5. Documentación a entregar	4
5.1. Fichero a entregar	6
6. Calificación de la práctica	7

1. Objetivo

El objetivo de esta práctica es que el alumno llegue a conocer los principales conceptos relacionados con las llamadas a procedimientos remotos (RPC, *Remote Procedure Call*) y servicios web.

2. Parte 2 (RPC)

2.1. Descripción de la funcionalidad

En esta parte de la práctica se trata de modificar la aplicación desarrollada en la Parte 1 para añadir un servicio, basado en RPC, que se encargue de almacenar toda la información sobre usuarios y contenidos publicados por ellos. El esquema es el que se muestra en la Figura 1. En la Parte 1 toda esta información la gestionaba directamente el servidor desarrollado (“Server” en la figura). El objetivo ahora es que el almacenamiento de la información lo realice un nuevo servidor RPC (Storage Server). De esta forma el servidor desarrollado en la Parte 1 tomará el papel de servidor para los clientes Java (User 1, etc.), y será cliente RPC para el servidor RPC que habrá que desarrollar en esta parte.

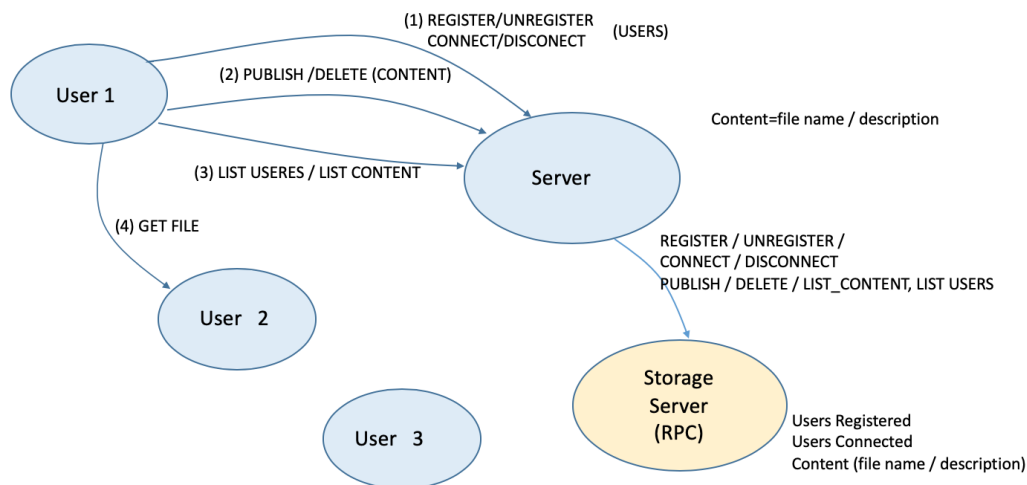


Figura 1: Sistema Peer-to-Peer con servidor RPC

Toda la información sobre usuarios registrados, conectados y contenido publicado debe almacenarse de forma persistente y gestionada por un servidor que utiliza el modelo RPC ONC-RPC. Este servicio de almacenamiento debe constar de al menos los siguientes servicios:

- Registrar a un usuario.
- Dar de baja a un usuario.
- Conectar a un usuario.

- Desconectar a un usuario.
- Publicar un fichero y la descripción asociada al mismo.
- Obtener la lista de usuarios conectados en el sistema.
- Obtener la lista de ficheros publicados por un usuario.

Cada vez que el servidor realizado en la parte anterior recibe una petición utilizando sockets, la encamina al servidor RPC utilizando las llamadas RPC que se definan. El protocolo y resto de la aplicación no cambia.

Se deja total libertad para definir la interfaz (fichero `.x`) que se considere más adecuada. El almacenamiento de la información en el servidor debe ser persistente, bien en ficheros simples o bases de datos, tal y como se está haciendo en la primera parte. Se deja libertad para elegir la solución que se desee. Para obtener la lista de usuarios y ficheros se pueden emplear listas dinámicas como las que dan entre los ejemplos de RPC que se encuentran en Aula Global.

2.2. Lenguajes de programación a utilizar

Para el desarrollo de esta parte se utilizará el lenguaje de programación C y el modelo ONC-RPC. El proceso intermediario (“Server” en la figura) es cliente del servicio RPC, mientras que el servidor ONC-RPC desarrollado ofrece el servicio.

3. Parte 3 (Servicios Web)

3.1. Descripción de la funcionalidad

El objetivo de la tercera parte es desarrollar y desplegar un servicio web utilizando JAX-WS. El servicio se desplegará, por simplicidad, en la máquina local donde ejecuta el programa cliente desarrollado en Java, aunque puede desplegarse en cualquier otra máquina de las aulas informáticas. Este servicio web tendrá una única operación que consistirá en convertir un texto enviado por el cliente Java en otro texto en el que todas las palabras del mismo estarán en mayúsculas. Este servicio se utilizará para transformar las descripciones enviadas por los clientes. Más específicamente, cada cliente hará uso del servicio web para re-escribir la descripción, cambiando todo a mayúsculas antes de que ésta sea enviada al servidor (ver Figura 2).

Una vez desarrollada la parte servidora, tendrá que incluirse en el cliente Java la parte cliente que permite invocar al servicio web. Cada vez que un usuario quiere publicar un fichero y su descripción, invocará al servicio web para convertir a mayúsculas el texto asociado a la descripción. Una vez convertido a mayúsculas, se enviará al servidor el fichero y la descripción tal y como se ha hecho en la Parte 1 de la práctica.

3.2. Lenguajes de programación a utilizar

Para el desarrollo de esta tercera parte se utilizará el lenguaje de programación Java y el API JAX-WS descrito en el material de la asignatura. Se deja libertad para elegir la interfaz y la definición del servicio que se considere más adecuada.

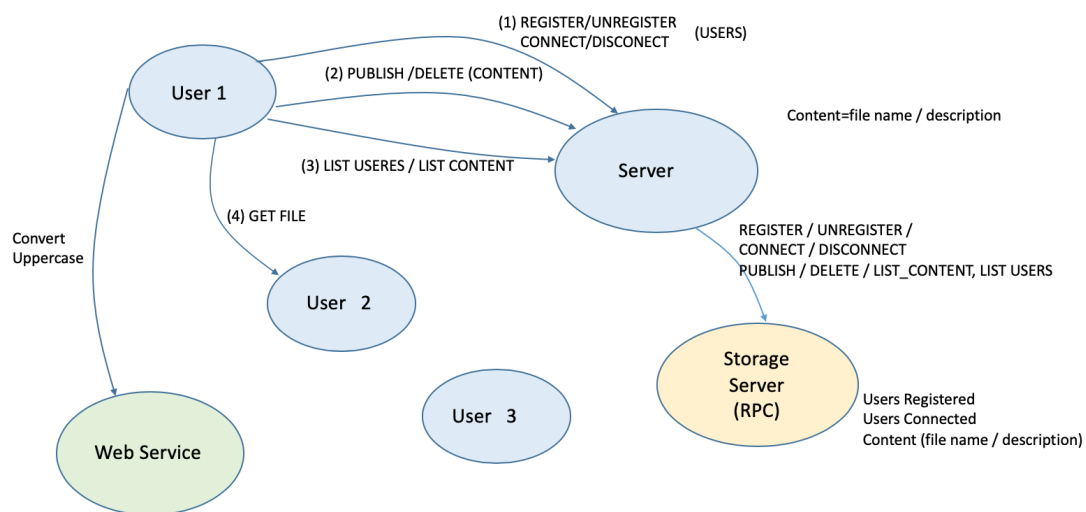


Figura 2: Sistema Peer-to-Peer con servicio web JAX-WS

4. Normas generales

Han de seguirse las mismas normas descritas en la Parte 1 de la práctica:

1. Las prácticas que no compilen o no se ajusten a la funcionalidad y requisitos planteados, **obtendrán una calificación de 0**.
2. Las prácticas que tengan *warnings* o no estén comentadas **serán penalizadas**.
3. Un programa no comentado, **obtendrá una calificación de 0**.
4. La entrega de la práctica se realizará a través de los entregadores habilitados. **No se permite la entrega a través de correo electrónico**.
5. Se prestará especial atención a detectar funcionalidades copiadas entre dos prácticas. En caso de detectar copia, se aplicará la normativa de la Universidad y los alumnos de los grupos involucrados en la copia tendrán un 0 en la nota final de la asignatura.
6. **La práctica debe funcionar en guernika.lab.inf.uc3m.es o en su defecto en máquinas Linux con Ubuntu**.
7. El sistema debe funcionar con todos los componentes del sistema ejecutando en máquinas distintas.
8. La memoria debe tener una **longitud máxima de 10 páginas aproximadamente**.

5. Documentación a entregar

La práctica se desarrollará en grupos de tres alumnos como máximo. La práctica sólo deberá ser entregada por un único integrante del grupo de prácticas en su grupo docente.

No se debe entregar la misma práctica de forma repetida por todos los integrantes del grupo.

El plazo de entrega de toda la práctica en su conjunto es el **Domingo 15 de Mayo**.

La entrega se realizará mediante Aula Global, a través de un entregador que se habilitará a tal efecto.

Se debe entregar un fichero comprimido en formato zip con el nombre **ssdd_p2_A_B_C.zip** donde A, B y C es el NIA de los integrantes que realizan la entrega.

El fichero debe contener:

- `autores.txt`, con los nombres y NIA de los integrantes del grupo.
- `memoria.pdf`
- `client.java`
- `server.c`
- Archivo `.x` que define la interfaz RPC.
- `servidorAlmacenamiento.c`, con la implementación del servicio de almacenamiento RPC.
- Fichero `Makefile` utilizado para compilar todos los archivos `.c`.
- Ficheros Java necesarios para el desarrollo del servicio web de conversión de texto.
- Fichero de texto de nombre **README** con instrucciones detalladas para la compilación y despliegue de todos los procesos involucrados en la aplicación.
- Cualquier otro fichero que se considere necesario para la compilación o evaluación de la práctica.

El fichero `servidor.c` entregado debe incluir la funcionalidad de todas las partes que se hayan completado. En particular, de las tres partes si se ha completado el desarrollo de toda la práctica.

Deben incluirse todos los archivos fuente necesarios para la compilación y un fichero de texto con nombre **README**, que incluirá instrucciones detalladas para la compilación y despliegue de todos los procesos involucrados en la aplicación.

La memoria de la práctica debe comentar los aspectos del desarrollo de la misma que considere más relevantes. Del mismo modo, puede exponer los comentarios personales que considere oportunos. Se deberá entregar un documento en formato PDF.

No descuide la calidad de la memoria de su práctica. Aprobar la memoria es imprescindible para aprobar la práctica, tanto como el correcto funcionamiento de la misma. **Si al evaluarse la memoria de su práctica, se considera que no alcanza el mínimo admisible, su práctica estará suspensa.**

La memoria tendrá que contener al menos los siguientes apartados:

- **Portada** donde figuren los autores (incluyendo nombre completo, NIA y dirección de correo electrónico).
- **Índice de contenidos**
- **Descripción del código** detallando las principales funciones implementadas. No incluir código fuente de la práctica en este apartado.
- **Descripción de la forma de compilar y obtener el ejecutable de todos los procesos involucrados.** Además se debe describir la forma de ejecutarlos.
- **Batería de pruebas** utilizadas y resultados obtenidos. Se dará mayor puntuación a pruebas avanzadas, casos extremos, y en general a aquellas pruebas que garanticen el correcto funcionamiento de la práctica en todos los casos.

Hay que tener en cuenta:

- Que el programa compile correctamente y sin *warnings* a ser posible.
- Evite pruebas duplicadas que evalúan los mismo flujos de programa. La puntuación de este apartado no se mide en función del número de pruebas, sino del grado de cobertura de las mismas. Es mejor pocas pruebas que evalúan diferentes casos, a muchas que evalúan siempre el mismo caso.
- **Conclusiones**, problemas encontrados, cómo se han solucionado, y opiniones personales.

Se puntuarán también los siguientes aspectos relativos a la **presentación**:

- La memoria debe tener números de página en todas las páginas (menos en la portada).
- El texto de la memoria debe estar justificado.

5.1. Fichero a entregar

Para crear el fichero a entregar se deben seguir los siguientes pasos:

- Se crea el directorio para preparar los materiales a entregar y se comprueba que se encuentra en el directorio de la entrega:

```
$ cd
$ mkdir ssdd_p2_AAAAAAAAAA_BBBBBBBBBB_CCCCCCCCC
$ cd ssdd_p2_AAAAAAAAAA_BBBBBBBBBB_CCCCCCCCC
```

- Después se procederá a copiar todos los ficheros con los programas desarrollados al directorio de la entrega y se procede a generar el fichero zip a ser entregado:

```
$ cd ..
$ ls
... ssdd_p2_AAAAAAAAAA_BBBBBBBBBB_CCCCCCCCC ...
$ zip -r ssdd_p2_AAAAAAAAAA_BBBBBBBBBB_CCCCCCCCC.zip
  ssdd_p2_AAAAAAAAAA_BBBBBBBBBB_CCCCCCCCC/
```

6. Calificación de la práctica

Sólo debe hacerse una entrega que podrá contener la funcionalidad completa de las tres partes, de solo dos, o de solo una. La práctica se calificará de la siguiente forma:

- La primera parte de la práctica se puntuará sobre 7 puntos.
- La entrega de la Parte 2 permitirá sumas dos puntos a la parte primera.
- La entrega de la Parte 3 permitirá sumar un punto a las partes anteriores.

De esta forma si solo se entrega la Parte 1, como máximo se obtendrán 7 puntos. La entrega de las Partes 1 y 2 permitiría obtener hasta 9 puntos, mientras que la entrega de las Partes 1 y 3 permitiría obtener hasta 8 puntos. La entrega de las tres partes permitiría obtener hasta 10 puntos. En todo caso, será obligatorio entregar la Parte 1.