

Datos

Profesores de Prácticas: **Sergio Villar, Carlos Montemuiño**

Material de las prácticas

Para las prácticas en las máquinas Linux, la documentación serán los HOWTOs y manpages del sistema. La presente guía es orientativa (no funcionará hacer copy/paste sin saber lo que se hace).

Las llamadas RPC (Remote Procedure Call) nos proporcionan una colección de herramientas y funciones de biblioteca para realizar llamadas entre cliente-servidor. Un servidor RPC consiste en una colección de procedimientos que un cliente puede solicitar por el envío de una petición RPC junto con los parámetros del procedimiento. El servidor invocará el procedimiento solicitado por el cliente, y le enviará de vuelta el resultado de la llamada. Para realizar estas llamadas RPC, se utilizan sockets estándar UDP y TCP para transportar los datos entre el cliente y el servidor.

1. Selección del entorno

Para la realización de esta práctica diseñaremos un entorno con dos VM: una máquina actuará de Master con dos interfaces de red y la otra de Worker con una única interfaz. La imagen a utilizar tanto para el Master como para el Worker será **CentOS7-KVM-X**.

Para la máquina Master, se deberá tener una interfaz con acceso a Internet (Interface *Internet* con IP 10.10.10.0/24) y otra interfaz de red interna (Interface *Middle* con IP 20.20.20.0/24 o 20.20.21.0/23).

En el caso de la máquina Worker, se debe tener una única interfaz con red interna (Interface *Middle* con IP 20.20.20.0/24 o 20.20.21.0/23), utilizada para comunicarse con el Master y obtener acceso a Internet a través de la máquina Master.

2. Configuración de la red

Una vez creadas las imágenes de las máquinas virtuales se debe modificar su nombre. Para ello, abrimos el fichero de configuración `/etc/hostname` utilizando un editor de texto, como por ejemplo *gedit*:

```
gedit /etc/hostname
```

Abierto el fichero, cambiamos `localhost.localdomainworker` por `master` -fijaos que `master` está en minúsculas-. Después del cambio se debe reiniciar la máquina para actualizar el nombre.

```
sudo reboot
```

Una vez hemos reiniciado la máquina, abrimos la consola y verificamos el cambio del nombre abriendo otra vez el fichero */etc/hostname*.

Ahora se debería hacer el mismo proceso para la máquina Worker. Abriremos el fichero */etc/hostname* y cambiaremos el nombre por worker -fijaos que worker está en minúsculas-.

Además, en las dos máquinas se debe añadir en el fichero */etc/hosts* las IPs de las máquinas, quedando las primeras líneas del fichero similares a las de la Figura 1.

Atención: fijaos bien en las direcciones de vuestras máquinas para configurar bien vuestra red, no tienen porque coincidir con las que se muestran en la figura.

127.0.0.1	localhost
::1	localhost
10.10.10.7	MasterI
20.20.20.4	MasterM
20.20.20.8	worker

Figura 1: Contenido del fichero */etc/hosts*

Para comprobar el correcto funcionamiento de la modificación de los ficheros, se debe hacer ping entre las dos máquinas y comprobar si las dos máquinas se comunican correctamente.

Desde el master:

```
ping worker
```

Desde el worker:

```
ping masterM
```

Realizar también un ping a masterI, debería fallar.

Configuración de red

Ahora configuramos Master y Worker para conseguir acceso a Internet desde la máquina Worker.

En la máquina Master abrimos el archivo */etc/sysctl.conf* y añadimos la línea `net.ipv4.ip_forward=1` para activar de forma permanente el ip forwarding y guardamos el fichero.

También debemos añadir una nueva regla a las iptables para redireccionar los paquetes del Worker enviados al Master con el exterior. Es importante que la dirección pública del Master sea la adecuada a nuestra red.

Importante: En función del rango de IP que contengan nuestras máquinas virtuales (20.20.20.XX o 20.20.21.XX), las máscara cambiará:

Rango 20 (IP 20.20.20.XX):

```
iptables -t nat -A POSTROUTING -s 20.20.20.0/24 -o ens3 -j SNAT --to 10.10.10.7
```

Rango 21 (IP 20.20.21.XX):

```
iptables -t nat -A POSTROUTING -s 20.20.21.0/23 -o ens3 -j SNAT --to 10.10.10.7
```

Además, Centos 7 lleva incluido un firewall (firewall-cmd) y debemos añadirle unas reglas al firewall para poder permitir el enmascaramiento de paquetes y el tráfico a través de la máquina master.

```
firewall-cmd --direct --add-rule ipv4 nat POSTROUTING 0 -o ens3 -j MASQUERADE

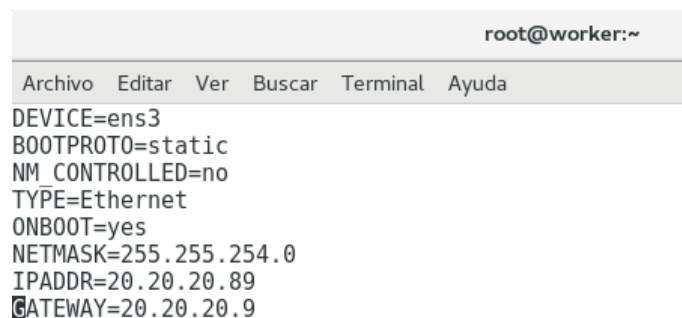
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i ens3 -o ens4
-m state --state RELATED,ESTABLISHED -j ACCEPT

firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i ens4 -o
ens3 -j ACCEPT
```

Con la finalidad de tener las reglas activadas al reiniciar la máquina, modificamos el fichero */etc/rc.local* y añadimos las reglas al final del fichero. Finalmente, ejecutamos el comando comentado dentro del fichero.

Ahora se reinicia la máquina Master y, mientras esperamos, entramos a la máquina Worker y cambiamos la configuración de las interfaces de red con el objetivo de asignar una puerta de entrada a la máquina.

En el fichero */etc/sysconfig/network-scripts/ifcfg-ens3* se añade el Gateway tal y como se muestra en la figura 2. Como se muestra en la figura, la IP del Gateway corresponde con la IP interna (*Middle*) de la máquina Master.



```
root@worker:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
DEVICE=ens3
BOOTPROTO=static
NM_CONTROLLED=no
TYPE=Ethernet
ONBOOT=yes
NETMASK=255.255.254.0
IPADDR=20.20.20.89
GATEWAY=20.20.20.9
```

Figura 2: Configuración del Gateway

Comprobamos si funciona el cambio cargando de nuevo la interfaz de red con el comando:

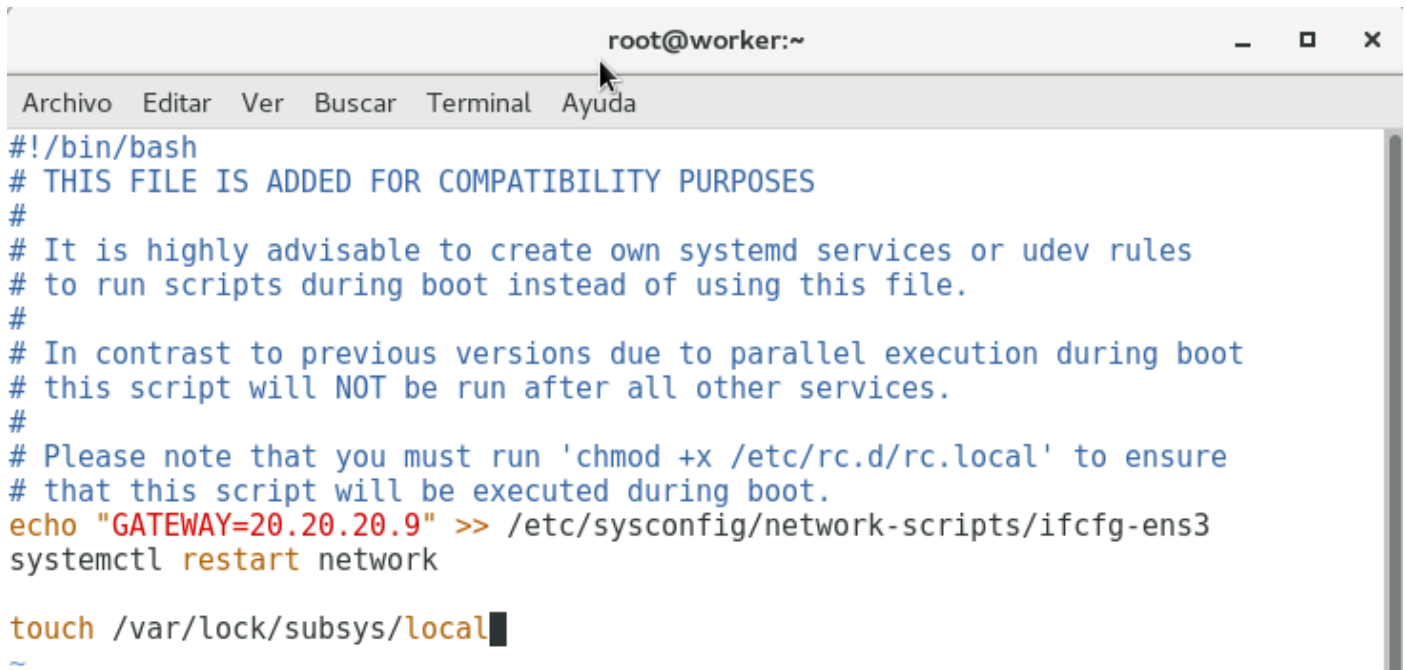
```
systemctl restart network
```

Para comprobar si funciona la conexión a internet de la máquina Worker, ejecutamos el siguiente comando:

```
wget www.uab.cat
```

Este comando nos debería descargar la página inicial de la UAB (index.html) en nuestra máquina. En caso de no encontrar la página dando un mensaje de error, deberemos repasar la configuración tanto del master como del Worker, ya que habremos cometido un error en la configuración.

Para no modificar después de cada reinicio el archivo de configuración de red, modificamos el fichero `/etc/rc.local` tal y como se muestra en la figura 3:



```
root@worker:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.
echo "GATEWAY=20.20.20.9" >> /etc/sysconfig/network-scripts/ifcfg-ens3
systemctl restart network

touch /var/lock/subsys/local
~
```

Figura 3: Modificación del fichero `/etc/rc.local`

3. Creación de usuario con sudo

A continuación, crearemos un nuevo usuario llamado *alumno* en cada una de las máquinas virtuales. El usuario *alumno* debe ser creado mediante el comando *adduser*.

```
adduser alumno
```

Este comando nos creará el usuario y su correspondiente home, reduciendo así la cantidad de pasos a realizar para crear un usuario si se compara con el comando *useradd*. A continuación, le asignamos una contraseña mediante el comando *passwd*:

```
passwd alumno
```

Seguidamente, al usuario *alumno* le daremos permisos de sudo mediante el fichero `/etc/sudoers`. Abrimos el fichero `/etc/sudoers` con un editor de texto después de hacer una copia por si hay alguna modificación errónea.

```
cp /etc/sudoers /etc/sudoers.backup
```

```
gedit /etc/sudoers
```

Atención: Fijaos bien en el nombre del fichero, si al abrirlo te sale completamente vacío has cometido un error al escribir el nombre del fichero.

En este fichero añadimos la línea siguiente:

```
alumno ALL=(ALL:ALL) ALL
```

Ahora, para comprobar si los cambios se han llevado a cabo correctamente, abrimos un nuevo terminal. En el nuevo terminal cambiamos de usuario e intentamos actualizar la lista de paquetes disponibles en nuestra distribución Linux.

```
su - alumno
```

```
sudo yum update
```

No es necesario actualizarlo ya que suele tardar bastante tiempo, puedes hacer “ctrl + c” para detener las actualización de los paquetes y continuar la práctica.

Si todo se ha ejecutado correctamente, seguid los mismos pasos para la otra máquina virtual.

4. Preparando el entorno para la practica de RPC¹

En este apartado configuraremos las herramientas RPC. El paquete que contiene las herramientas RPC es el “port mapper”. Puesto que las imágenes de las máquinas virtuales ya tienen instalado el paquete no será necesario volverlo a instalar.

Lo primero que haremos será dar permisos al firewall de Centos para que acepte llamadas RPC remotas. Para ello ejecutamos los siguientes comandos en la máquina Master:

```
alumno@master:~$ sudo firewall-cmd --permanent --add-service=rpc-bind
alumno@master:~$ sudo firewall-cmd --permanent --add-port=10000-64000/tcp
alumno@master:~$ sudo firewall-cmd --permanent --add-port=10000-64000/udp
alumno@master:~$ sudo firewall-cmd --reload
```

A continuación comprobaremos si el servicio que atiende llamadas RPC (rpc-bind) está activo. Para ello,

¹ https://es.wikipedia.org/wiki/Llamada_a_Procedimiento_Remoto

ejecutaremos el comando *rpcinfo* que muestra los servicios que están activos y listos para recibir peticiones en un servidor determinado. Para listar los servicios activos de un servidor determinado, utilizamos la opción *-p* seguido del *hostname* del servidor que queremos listar sus servicios activos.

```
alumno@master:~$ rpcinfo -p masterM
```

program	vers	proto	port	service
100000	4	tcp	111	portmapper
100000	3	tcp	111	portmapper
100000	2	tcp	111	portmapper
100000	4	udp	111	portmapper
100000	3	udp	111	portmapper
100000	2	udp	111	portmapper

Puesto que en la máquina Worker no queremos tener ningún servicio portmap, sólo las herramientas del cliente, desactivamos su servicio *rpcbind*:

```
$ sudo systemctl disable rpcbind.service
```

Quizá pensemos que ya tenemos parado el servicio, sin embargo, si ejecutamos el commando *rpcinfo* observamos lo siguiente:

```
alumno@worker:~$ rpcinfo -p worker
```

program	vers	proto	port	service
100000	4	tcp	111	portmapper
100000	3	tcp	111	portmapper
100000	2	tcp	111	portmapper
100000	4	udp	111	portmapper
100000	3	udp	111	portmapper
100000	2	udp	111	portmapper

Pregunta 1.1 (0.5 pts): ¿Porque sigue levantado el *prortmapper* del *worker*?

Para que los cambios tengan efecto, será necesario reiniciar la máquina:

```
$ sudo reboot
```

Cuando la máquina se haya reiniciado comprueba que el servicio *rpcbind* (*portmap*) ya no está levantado. Si ejecutamos otra vez *rpcinfo* obtendremos lo siguiente:

```
alumno@worker:~$ rpcinfo -p worker
rpcinfo: can't ontact portmapper: RPC: Remote system error -
Connection refused
```

Puesto que una vez reiniciada la máquina el servicio ya está desactivo, el sistema rechaza la conexión. A modo informativo (no ejecutar el comando), si quisiésemos volverlo a activar, bastaría con ejecutar el comando:

```
$ sudo systemctl enabled rpcbind.service
```

Pregunta 1.2 (1 pt): ¿Cuál es el resultado que aparece ahora y por qué?

Ahora comprueba lo siguiente:

```
alumno@worker:~$ sudo rpcinfo -p masterM
```

Pregunta 1.3 (0.5 pts): ¿En qué máquina está levantado el servicio que está respondiendo al *rpcinfo* ?

5. Ajustes en la seguridad del “port mapper”

Por defecto y por motivos de seguridad, el “port mapper” no aceptará que un programa RPC diseñado y compilado por un usuario sea utilizado por otras máquinas de la red.

Para hacer que “port mapper” acepte las peticiones del servidor RPC que vamos diseñar y compilar, será necesario modificar el fichero `/etc/sysconfig/rpcbind` en el Master, que es la máquina que alojará el nuevo servicio RPC. Para ello añadimos la siguiente línea:

```
RPCBIND ARGS="-i -w"
```

6. Compilando y ejecutando un ejemplo de RPC

El **objetivo** en este ítem es compilar y ejecutar un ejemplo sencillo de un programa con RPC en el entorno preparado.

En las máquinas Master y Worker se debe descargar un fichero “.zip”. Este “.zip” contiene cuatro ficheros en una carpeta nombrada “**example**”. El enlace de descarga del fichero es: <https://cutt.ly/5ehIP4J>

Para extraer el contenido del fichero “**EjemploRPC.zip**” puede usar el comando “**unzip**”:

```
alumno@master:~$ unzip EjemploRPC.zip
Archive:  EjemploRPC.zip
  inflating: example/dateproc.c
  inflating: example/date.x
  inflating: example/Makefile
  inflating: example/rdate.c
```

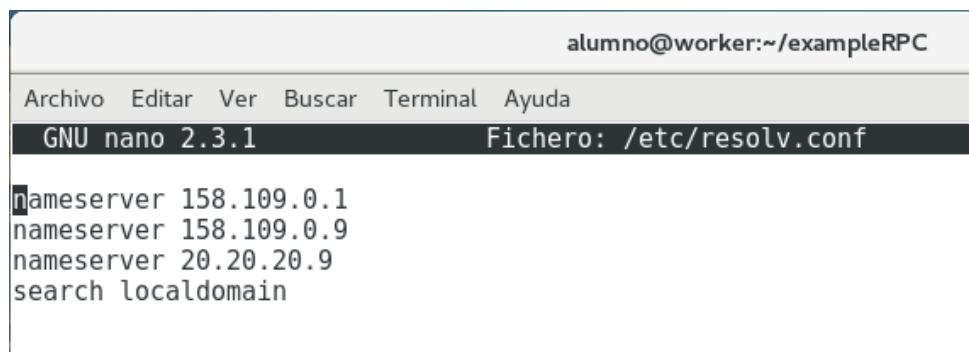
El ejemplo está programado utilizando el lenguaje de programación “C” y contiene un “**Makefile**” para compilarlo.

Antes de compilar el programa tendremos que instalar el compilador GCC y todas sus dependencias, para ello ejecutamos los siguientes comandos en la maquina Master:

```
alumno@master:~$ sudo yum clean all
alumno@master:~$ sudo yum clean metadata
alumno@master:~$ sudo yum install gcc
```

Mientras esperamos que se actualicen los paquetes en la máquina Master, podemos ir trabajando en la máquina Worker.

En la máquina Worker será necesario previamente a actualizar los paquetes configurar el fichero */etc/resolv.conf* añadiendo una nueva línea con un nameserver valido. En este caso añadiremos la IP *middle* del master, tal y como se muestra en la figura 4.



```
alumno@worker:~/exampleRPC
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.3.1  Fichero: /etc/resolv.conf
nameserver 158.109.0.1
nameserver 158.109.0.9
nameserver 20.20.20.9
search localdomain
```

Figura 4: Configuración del archivo resolv.conf

Una vez hemos añadido esta línea y hemos reiniciado la máquina, ejecutaremos los mismos pasos que en la máquina Master para instalar el compilador GCC:

```
alumno@worker:~$ sudo yum clean all
alumno@worker:~$ sudo yum clean metadata
alumno@worker:~$ sudo yum install gcc
```

Una vez hemos instalado el compilador GCC en las dos máquinas, el siguiente paso será compilar el programa RPC. Para compilar el programa entre en la carpeta “**example**” y ejecute el comando “**make**” tanto en la máquina Master como Worker.

```
alumno@master:~/example$ make
rpcgen date.x
gcc -o rdate rdate.c date_clnt.c
gcc -o dateproc dateproc.c date_svc.c
alumno@master:~/example$ ls -l
total 56
-rw-rw-r-- 1 alumno alumno 858 oct 22 11:58 date_clnt.c
```



```
-rw-rw-r-- 1 alumno alumno 891 oct 22 11:58 date.h
-rwxrwxr-x 1 alumno alumno 13872 oct 22 11:58 dateproc
-rw-r--r-- 1 alumno alumno 305 oct 2 2012 dateproc.c
-rw-rw-r-- 1 alumno alumno 2316 oct 22 11:58 date_svc.c
-rw-r--r-- 1 alumno alumno 204 oct 5 2012 date.x
-rw-r--r-- 1 alumno alumno 248 oct 2 2012 Makefile
-rwxrwxr-x 1 alumno alumno 9280 oct 22 11:58 rdate
-rw-r--r-- 1 alumno alumno 724 oct 2 2012 rdate.c
```

Si observamos la salida por pantalla del comando Make, nos muestra los pasos realizados para compilar el programa:

1. Ejecuta el “**rpcgen**” del fichero “**date.x**” para generar los ficheros “**date_clnt.c**”, “**date.h**” y “**date_svc.c**”; A partir de una sintaxis RPC (.X), genera el equivalente en C.
2. Después compilar el programa ejemplo cliente “**rdate**” basándose en los ficheros “**rdate.c**” y “**date_clnt.c**”; Es el cliente
3. Por fin si compila el programa ejemplo servidor “**dateproc**” basándose en los ficheros “**dateproc.c**” y “**date_svc.c**”. Es el servidor que se instalará en el servicio RPC.

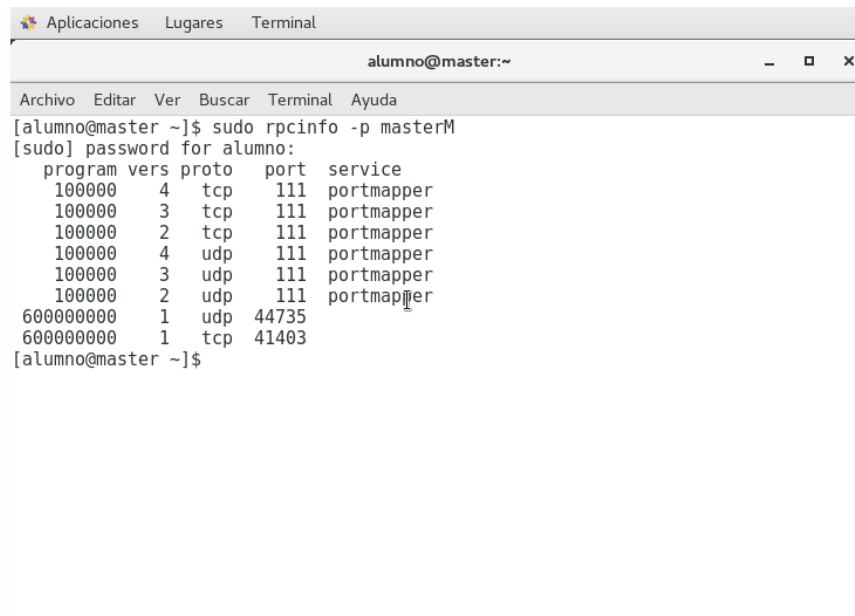
Una vez compilado, arranca en el Master el programa llamado “**dateproc**” que se instalará en un puerto del servicio RPC.

```
alumno@master:~/example$ ./dateproc
```

Para verificar si el programa servidor RPC que hemos compilado está activo y publicado como disponible para recibir peticiones, use en una nueva ventana de terminal el comando de diagnóstico del “port mapper” *rpcinfo*.

```
alumno@master:~/example$ sudo rpcinfo -p masterM
```

La salida del comando se muestra en la figura 5. Como se puede apreciar el servicio está activo en los puertos 44735 (upd) y 41403 (tcp).



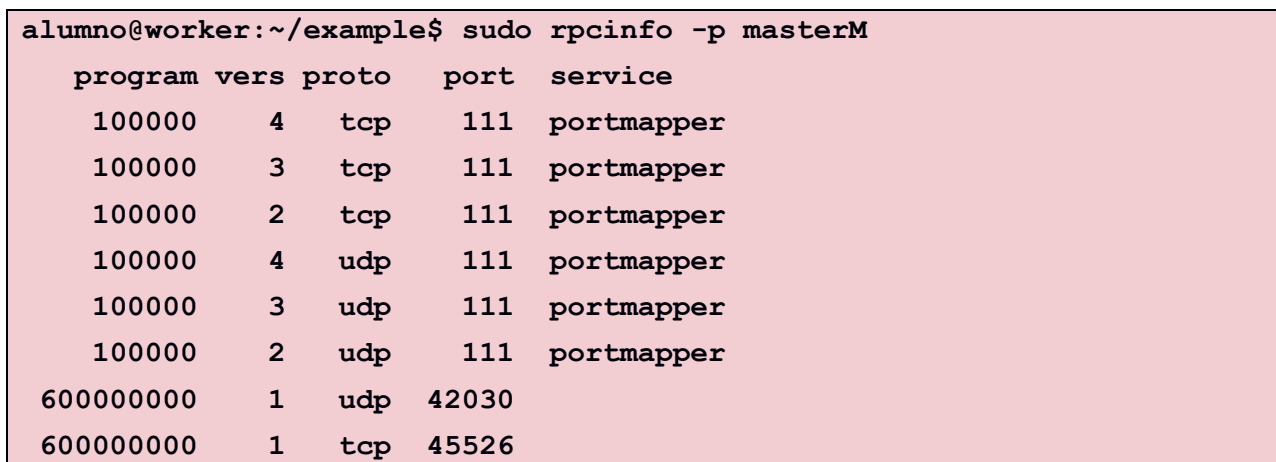
```
alumno@master:~  
[alumno@master ~]$ sudo rpcinfo -p masterM  
[sudo] password for alumno:  
  program vers proto  port  service  
    100000   4   tcp    111  portmapper  
    100000   3   tcp    111  portmapper  
    100000   2   tcp    111  portmapper  
    100000   4   udp    111  portmapper  
    100000   3   udp    111  portmapper  
    100000   2   udp    111  portmapper  
 600000000   1   udp   44735  
 600000000   1   tcp   41403  
[alumno@master ~]$
```

Figura 5: Salida del comando rpcinfo con los servicios que están activos

Cuando necesite, para interrumpir la ejecución del programa “dateproc” use la combinación de teclas “Ctrl + Z” en la ventana de terminal donde el programa esté ejecutándose.

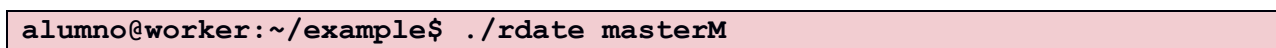
Pregunta 2.1 (1.5 pts): ¿Cómo podemos verificar si las nuevas informaciones que aparecen en la salida del comando de diagnóstico del “port mapper” son las del programa ejemplo compilado y ejecutado?

Para comprobar que el “port mapper” en el servidor está aceptando peticiones de la máquina cliente, podemos ejecutar el comando “rpcinfo -p” pasando como parámetro el nombre de la máquina que queremos diagnosticar (“masterM”):



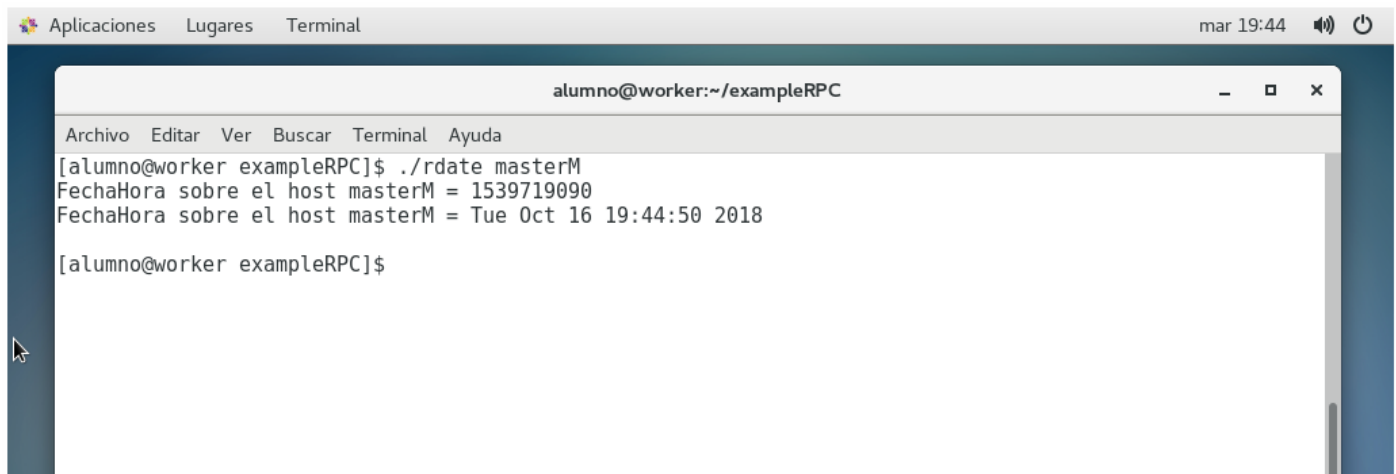
```
alumno@worker:~/example$ sudo rpcinfo -p masterM  
  program vers proto  port  service  
    100000   4   tcp    111  portmapper  
    100000   3   tcp    111  portmapper  
    100000   2   tcp    111  portmapper  
    100000   4   udp    111  portmapper  
    100000   3   udp    111  portmapper  
    100000   2   udp    111  portmapper  
 600000000   1   udp   42030  
 600000000   1   tcp   45526
```

Para ejecutar el programa cliente del ejemplo deberemos informarle el nombre de la máquina que está ejecutando el servidor:



```
alumno@worker:~/example$ ./rdate masterM
```

La salida de la llamada, será la fecha y hora del servidor remoto, tal y como se muestra en la figura 6.



```
alumno@worker: ~/exampleRPC
[alumno@worker exampleRPC]$ ./rdate masterM
FechaHora sobre el host masterM = 1539719090
FechaHora sobre el host masterM = Tue Oct 16 19:44:50 2018
[alumno@worker exampleRPC]$
```

Figura 6: Salida de la llamada RPC remota

Para el dateproc (sin parar el rpcbind) y vuelve a ejecutar el rdate desde el Worker

Pregunta 2.2 (2 pts): ¿Cual es el resultado y por qué?

Pregunta 2.3 (0.5 pts): ¿Cual es identificador del programa dateproc y el puerto(s) en el que está a la escucha?

7. Modificación del programa RPC

Pregunta 3.1 (1 pt): ¿Cuales son los ficheros de los códigos fuente originales de nuestro ejemplo y cuáles son los ficheros de código fuentes generados automáticamente por **rpcgen**?

Pregunta 3.2:

Basándose en los códigos fuente del ejemplo colgado en el Campus Virtual, modifíquelos añadiéndole dos funcionalidades más:

3.2.1 (2 pts) Una que devuelva en valor numérico (timestamp) la diferencia del reloj pasado como parámetro con respecto al reloj del servidor

3.2.2 (1 pts) Otra que haga lo mismo que la anterior, pero que muestre el resultado en formato texto usando para la conversión timestamp a texto el procedimiento correspondiente del servidor.

En el documento de la entrega, además de las pantallas con los resultados **adjunte** los **códigos fuentes modificados** (¡¡no adjunte los códigos fuente generados mediante **rpcgen**!!)