



PRÁCTICA DE NIVEL DE TRANSPORTE (SOCKETS)

(CC) Moreno, A. M. & Bravo, S. & Cruz, D. (Redes I, 2024)

Tabla de contenido



- Presentación
 - ▣ Objetivos y Entorno de desarrollo
 - ▣ Servicio “Finger”
 - Especificaciones del protocolo
 - Mensajes
 - Ejemplo de dialogo
 - Requisitos

Objetivos y Entorno de desarrollo



- El objetivo de esta práctica es implementar una aplicación en red como
 - ▣ usuario del nivel de transporte y
 - ▣ según la arquitectura o modelo de programación cliente-servidor
- Entorno de desarrollo
 - ▣ Estación de trabajo con S.O. Debian GNU/Linux 9 (*stretch*) (nopal.usal.es)
 - ▣ Sockets de Berkeley
 - ▣ Lenguaje de programación C

Especificaciones del protocolo

- El servicio que vamos a implementar se denomina “Finger”
- El servicio finger es un protocolo que proporciona información de los usuarios de una máquina, estén o no conectados en el momento de acceder al servicio.
 - ▣ Tiene asignado el puerto de red 79, pudiendo emplear los protocolos TCP y UDP, si bien el primero es más usual.
 - ▣ Se describe en el [RFC 1288](#) (Zimmerman, 1991).

Mensajes Finger (I)

5

- PC emplea dos tipos de mensajes:
 - ▣ Peticiones de los clientes a los servidores
 - ▣ Respuestas de los servidores a los clientes
 - ▣ Las líneas de los mensajes de Finger
 - Son siempre líneas de caracteres terminadas con los caracteres CR-LF (retorno de carro ‘\r’ (ASCII 13 (0x0D)), - avance de línea ‘\n’ (ASCII 10 (0x0A)))
 - La longitud máxima de las líneas no debe exceder 516 bytes, contando todos los caracteres (incluido el CR-LF final)

Mensajes Finger (II)

6

□ Sintaxis del cliente

- `finger [usuario] [usuario@host]`
- Si no se especifican argumentos (solo se envía un `<CRLF>`), se presentará una entrada para cada usuario actualmente conectado al sistema
- Si se especifica un nombre de usuario (por ejemplo, `FOO<CRLF>`), la respuesta debe referirse sólo a ese usuario en particular, aunque no haya iniciado sesión
 - Si no se recuerda el nombre de la cuenta podemos utilizar el nombre real del usuario (no diferencia mayúsculas de minúsculas)
- Si no se especifica host la información se refiere al equipo local
- Si se especifica un equipo la información se requerirá al servidor del equipo especificado

Ejemplos

```
$ finger -l
```

Login: mortadelo	Name: Mortadelo Agente Secreto
Directory: /home/mortadelo	Shell: /bin/bash
Office: 13, 555-123456	Home Phone: 555-789123
On since Mon Mar 8 11:32 (CET) on tty2	1 hour 28 minutes idle
(messages off)	

Login: anacleto	Name: Anacleto Agente Secreto
Directory: /home/anacleto	Shell: /bin/bash
Office: 16, 555-123458	Home Phone: 555-466789
On since Mon Mar 8 11:33 (CET) on tty3	1 hour 27 minutes idle
(messages off)	

Login: filemon	Name: Filemón
Directory: /home/filemon	Shell: /bin/bash
Office: 14, 555-123457	Home Phone: 555-789456
On since Mon Mar 8 11:33 (CET) on tty4	1 hour 27 minutes idle
(messages off)	

Ejemplos

```
$ finger agente
```

```
  Login: anacleto
```

```
  Name: Anacleto Agente Secreto
```

```
  Directory: /home/anacleto
```

```
  Shell: /bin/bash
```

```
  Office: 16, 555-123458
```

```
  Home Phone: 555-466789
```

```
  On since Mon Mar  8 11:33 (CET) on tty3    2 hours 9 minutes idle
```

```
    (messages off)
```

```
$ finger asdfg@equipo.dominio.com
```

```
  Welcome to Linux version 5.10.0-23-amd64 at equipo !
```

```
    09:15:01 up 489 days, 19:56,  1 user,  load average: 0,08, 0,07, 0,02
```

```
  finger: fghfgh: no such user.
```


Obteniendo la información en el servidor

Ficheros	Descripción
/var/log/wtmp	Contiene información de todos los accesos al equipo, desde la IP que se accedió y el día y hora del inicio y fin de la sesión, pero no se puede consultar directamente sino con <i>last</i> o con <i>utmpdump</i> .
/run/utmp	Registros de las sesiones de inicio de sesión actuales.
/var/log/lastlog	Accesos al equipo.
/etc/passwd	Contiene información del usuario.

Obteniendo la información en el servidor

Comando	Descripción
hostname	Nombre del equipo
uname -a	Nombre del equipo y versión del SSOO
lsb_release -d	Muestra la distribución y el número de versión actual.
uptime	Mostrar el tiempo actual del sistema, el tiempo de actividad del equipo, el número de usuarios que han iniciado sesión actualmente en el sistema y los promedios de carga de los últimos 1, 5 y 15 minutos.
last -s today (accesos de hoy) last -F (accesos con la fecha completa) last usuario (accesos del usuario especificado)	Información de todos los accesos al equipo, desde la IP que se accedió y el día y hora del inicio y fin de la sesión.
who	Consultar los usuarios conectados
w	Consultar los usuarios conectados y qué están haciendo
lastlog lastlog -u usuario (último acceso del usuario especificado)	Lista de todos los usuarios con sus últimos intentos de inicio de sesión
getent passwd [usuario]	Muestra lo mismo que si consultamos el fichero /etc/passwd, pero si hay otros (ejemplo nuestro LDAP desde nogal) también lo muestra.
utmpdump /var/log/wtmp	Volcar el contenido de los ficheros binarios a texto.

Requisitos (I)

□ Programa Servidor

- Aceptará peticiones de sus clientes tanto en TCP como en UDP
 - Registrará todas las peticiones en un fichero de "log" llamado `peticiones.log` en el que anotará:
 - Fecha y hora del evento
 - Descripción del evento:
 - Comunicación realizada: nombre del host, dirección IP, protocolo de transporte, nº de puerto efímero del cliente
 - Orden recibida: nombre del host, dirección IP, protocolo de transporte, el puerto del cliente y la orden.
 - Respuesta enviada: nombre del host, dirección IP, protocolo de transporte, puerto del cliente y la respuesta mandada.
 - Comunicación finalizada: nombre del host, dirección IP, protocolo de transporte, nº de puerto efímero del cliente
- Se ejecutará como un "daemon"

□ Programa Cliente

- Se comunicará con el servidor bien con TCP o con UDP
- Las sintaxis será la del comando *finger* añadiendo como parámetro el protocolo de transporte TCP o UDP de la siguiente forma:
 - cliente TCP/UDP [usuario] [usuario@host]
- Realizará peticiones al servidor como se ha indicado anteriormente
- Realizará las acciones oportunas para su correcta finalización

Requisitos (II): pruebas

12

- Durante la fase de pruebas el cliente podrá ejecutarse como se muestra en el ejemplo de diálogo anterior, pero en la versión para entregar el cliente
 - ▣ Escribirá las respuestas obtenidas del servidor y los mensajes de error y/o depuración en un fichero con nombre el número de puerto efímero del cliente y extensión .txt

Requisitos (III): versión entregable

13

- Para verificar que esta práctica funciona correctamente y permite operar con varios clientes, se utilizará el *script* `lanzaServidor.sh` que ha de adjuntarse obligatoriamente en el fichero de entrega de esta práctica
- El contenido de `lanzaServidor.sh` es el siguiente:

```
# lanzaServidor.sh
# Lanza el servidor que es un daemon y varios clientes
./servidor
./cliente TCP &
./cliente TCP David &
./cliente TCP @nagal.usal.es &
./cliente TCP zxcvb@nagal.usal.es &
./cliente TCP p1777001@nagal.usal.es &
./cliente UDP &
./cliente UDP david &
./cliente UDP @nagal.usal.es &
./cliente UDP zxcvb@nagal.usal.es &
./cliente UDP p1777001@nagal.usal.es &
```

Requisitos (IV): documentación

14

- Entregar un informe en formato PDF que contenga:
 - ▣ Detalles relevantes del desarrollo de la práctica
 - ▣ Documentación de las pruebas de funcionamiento realizadas