



Mini-cliente de consulta de servidores DNS

Enunciado:

Empleando el lenguaje de programación C, se desea implementar un programa que tenga como propósito servir de herramienta de consulta y exploración de la jerarquía *DNS* bajo el sistema operativo *GNU/Linux*. El mismo se ejecutará desde la consola y deberá interactuar con uno o más servidores *DNS*, respetando siempre el intercambio de mensajes estipulado en los *RFC 1034* y *RFC 1035*.

Opciones del programa:

El programa a ser implementado se debe denominar `dnsquery`. Los parámetros opcionales que se deben contemplar (los cuales pueden aparecer en cualquier orden), son los siguientes:

```
query consulta @servidor[:puerto] [-a | -mx | -loc] [-r | -t] [-h]
```

El significado de los diversos parámetros es el siguiente (recordar que las opciones entre corchetes resultan opcionales y las separadas por la barra vertical son alternativas excluyentes):

- `consulta`: este argumento, el cual debe estar siempre presente salvo que se esté invocando a la ayuda a través del parámetro opcional `-h`, es la consulta que se desea resolver (en general, la cadena de caracteres denotando el nombre simbólico que se desea mapear a un **IP**).
- `@servidor`: el cliente debe resolver la consulta suministrada contra el servidor **DNS** que se especifique con este argumento. Caso contrario, de no estar presente este argumento, se debe resolver la consulta suministrada contra el servidor **DNS** por defecto.
- `[:puerto]`: este parámetro opcional, el cual sólo puede ser especificado si también se especificó un servidor, permite indicar que el servidor contra el cual se resolverá la consulta no está ligado al puerto **DNS** estándar. Caso contrario, al no indicarse el puerto que se deba usar se asumirá que las consultas serán dirigidas al puerto estándar del servidor (ya sea especificado a través del parámetro anterior o bien el configurado por defecto).
- `[-a | -mx | -loc]`: estos parámetros opcionales, excluyentes entre sí, denotan el tipo de consulta que se está realizando. Si se especifica el parámetro `-a`, la consulta se trata de un nombre simbólico y se desea conocer su correspondiente **IP** numérico asociado. Si se especifica el parámetro `-mx`, se desea determinar el servidor a cargo de la recepción de correo electrónico para el dominio indicado en la consulta. Finalmente, si se especifica el parámetro `-loc`, se desea recuperar la información relativa a la ubicación geográfica del dominio indicado en la consulta. En caso de no indicarse el tipo de la consulta, se debe asumir por defecto que se trata de una consulta sobre los registros **DNS** de tipo **A** (es decir, el cliente se comporta como si se hubiese especificado el parámetro opcional `-a`).



PROYECTO N° 1

REDES DE COMPUTADORAS

Licenciatura en Ciencias de la Computación - Primer cuatrimestre de 2019
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



- `[-r | -t]`: estos parámetros opcionales, excluyentes entre sí, denotan la manera en la cual se desea resolver la consulta. En caso de especificarse el parámetro `-r`, se está solicitando que la consulta contenga el *bit recursion desired* activado (es decir, puede pasar una de dos cosas, o bien el servidor acepta la consulta recursiva y retorna la respuesta definitiva, o bien rechaza ese tipo de consulta, por lo que no se podrá obtener la respuesta definitiva bajo esas condiciones). Caso contrario, al especificarse el parámetro `-t`, se está solicitando que la consulta se resuelva iterativamente, mostrando una traza con la evolución de la misma (esto es, mostrando las distintas respuestas parciales que se van obteniendo al efectuar una consulta no recursiva hasta dar con la respuesta final, similar a la funcionalidad provista por el comando: `dig` con la opción `--trace`). Por último, en caso de no especificarse ni uno ni otro, se asumirá por defecto que la consulta debe ser resuelta de manera recursiva (es decir, retornando directamente la respuesta final).
- `[-h]`: si se especifica el parámetro opcional `-h`, el programa sólo deberá mostrar una pequeña ayuda por pantalla, la cual consiste en listar las opciones de invocación del programa (sintaxis) y cuál es su propósito. En caso de estar presente esta opción no se deberá iniciar el cliente, tan sólo se muestra el texto de ayuda.

Consideraciones de implementación:

Las conexiones se deberán implementar bajo protocolo *UDP* haciendo uso de la librería *sockets*.

Tanto la especificación respecto al intercambio de mensajes entre el cliente y los servidores, así como el formato y significado de los mismos puede consultarse en los *RFC 1034* y *RFC 1035*.

Para probar el correcto funcionamiento del cliente se puede hacer uso de cualquier servidor *DNS*, ya sea local o externo. En caso de usar un servidor externo, verificar que no exista un *firewall* en el medio que esté bloqueando las consultas salientes o bien las respuestas entrantes.

Observaciones:

- En el código, entre eficiencia y claridad, se debe optar por la claridad. Toda decisión en este sentido debe constar en el informe que acompaña al programa implementado.
- Deben efectuar la programación utilizando técnicas de programación modular (incluyendo funciones y procedimientos), estudiadas durante la carrera.
- Se tendrán en cuenta aspectos relacionados con buenas técnicas de programación, simplicidad y modulación de los algoritmos implementados.
- El código entregado deberá estar comentado e indentado adecuadamente.



Documentación:

- Deberá confeccionar la documentación orientada al programador, comentarios en el programa, y aclaraciones sobre parámetros de compilación (si los hubiese).
- Cada procedimiento/función deberá estar comentado, se deberá especificar el prototipo de la misma, explicando cada uno de los parámetros de entrada y/o salida.

Condiciones de entrega:

- El proyecto debe ser realizado en comisiones de **dos personas (sin excepciones)**.
- El módulo principal del programa se deberá denominar: **`dnsquery.c`**
- ☞ El trabajo debe ser **entregado y defendido**, el día 3 de junio de 2019, en el horario de la clase práctica.
- Deberá entregarse un folio de plástico correctamente cerrado que contenga:
 - ☑ Un *pendrive* con una carpeta llamada: `proyecto`, la cual debe contener:
 - Una carpeta `bin`, con el binario del programa (`cliente`) compilado y listo para ser ejecutado.
 - Una carpeta `src`, con todos los archivos fuentes junto con cualquier otro archivo adicional que se necesite para poder compilar al código fuente que dio origen al ejecutable contenido en la carpeta `bin`.
 - ☑ El informe impreso (**con nombre y LU de cada uno de los alumnos**)
 - ☑ El código impreso de todos los módulos implementados contenidos en la carpeta: `src`.

Condiciones de aprobación:

- No serán corregidos programas que generen errores durante la compilación.
- No serán corregidos los proyectos donde los archivos no respeten los nombres solicitados.
- En consecuencia: proyectos **no corregidos** serán desaprobados. ☹