Monitor de Disponibilidad de Servicios

Gabriel Ahumada
2821039-6
gabriel.ahumada@alumnos.usm.cl

Constanza Valdés
2821143-0
constanza.valdes@alumnos.usm.cl

16-10-12

1 Objetivo

Se desea monitorizar ciertos servicios de forma remota. Si existe alguna irregularidad, un correo es enviado al administrador con la descripción del problema. Para ello, se ha escrito un programa in `C` llamado `msd.c` que permite mediante su ejecución cada 60 segundos revisar los siguientes requerimientos:

1. El servidor `DNS` dado responde consultas `DNS`, en particular para ubicar a `www.google.cl`.
2. La página web indicada no retorna un ćódigo `HTTP` de error.
3. El uso de la `CPU` no supera el valor porcentual dado.
4. El uso de la memoria no supera el valor porcentual dado.

2 Funcionamiento

El programa se divide en 3 secciones: la principal, la de verificación y finalmente la de envío.

2.1 Sección principal

Esta sección es la codificada como función `main()`. Está encargada de recoger los argumentos extraídos por el usuario y asignarlos a las funciones de verificación. Además incorpora una alarma que contabiliza 60 segundos *reales* entre cada iteración donde son invocadas las funciones iterativamente.

2.2 Sección de verificación

Esta sección está compuesta por tres funciones codificadas como `checkDNS()`, `checkWebsite()` y `checkPerformace()`.

1. `checkDNS()`: Función encargada de verificar el correcto funcionamiento del DNS dado por el usuario. Lo realiza a través de un proceso hijo, el cual se comunica con el padre a través de un *pipe*. La labor del primero es ejecutar el comando `nslookup` en modo no interactivo que realiza la consulta. La respuesta es procesada tal que se espera que coincida con la particular ubicación de `www.google.cl`.

2. `checkWebsite()`: Función que corrobora si la `URL` entregada está disponible. Igual que en la función anterior, un proceso hijo es el que ejecuta el comando `curl` con opciones que arrojan sólo el encabezado de la petición `HTTP`. A través de un *pipe*, el proceso padre recibe y procesa lo entregado por el servidor. Si el código `HTTP` arroja valores distintos de 2XX y 3XX, se entiende que la página web no está disponible.

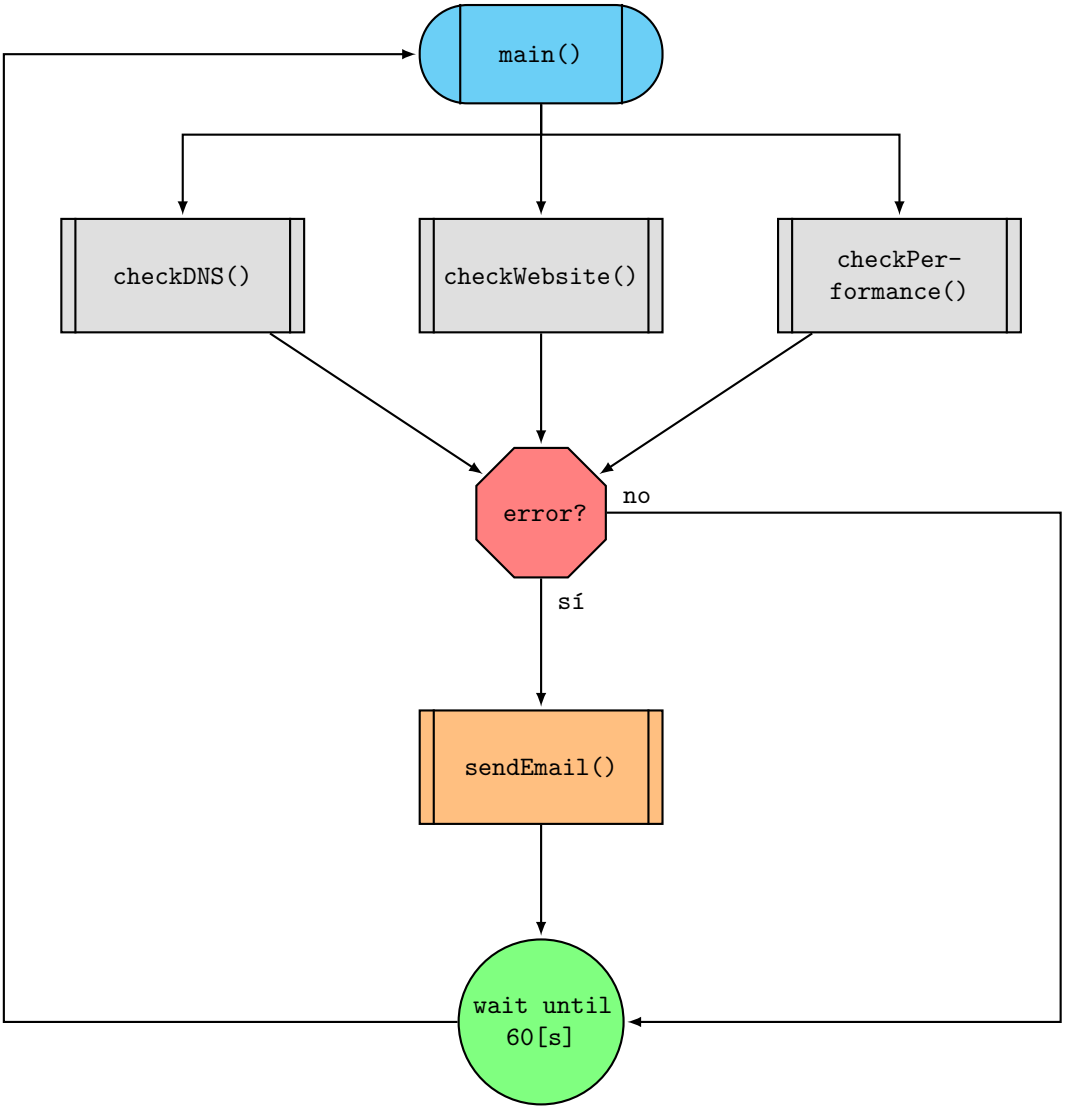
3. `checkPerformace()`: Función que inspecciona el estado actual de la máquina donde es ejecutado el programa `mds`. Siguiendo el mismo prototipo de las funciones anteriores, un proceso hijo es el que ejecuta el comando `ps` con opciones tales que sólo se obtengan los porcentajes de uso de la cpu y memoria. Con esta información - también por medio de un *p*ipe - el proceso padre analiza la salida, sumando cada fila obteniendo así el rendimiento actual. Finalmente, lo compara con el límite que el usuario establece a través de los argumentos dados.

2.3 Sección de envío

Esta sección está codificada como la función `sendEmail()`. Cuando se detecta una falla o el rendimiento no es el esperado, la sección previa invoca esta función. Por ello, según el argumento recibido agrega las respectivas alertas al correo a enviar. Por medio de un proceso hijo ejecuta el comando `openssl` con las opciones necesarias para efectuar la comunicación con el servidor de correos. El proceso padre se encarga de redactar el mensaje y dirigirlo según la información dada por el usuario.

3 Diagrama de Flujo

Se detalla un diagrama de alto nivel que representa el funcionamiento del programa a través de un flujo.



4 Resultados

Se muestra la ejecución del programa con una falla y posteriormente la recepción en el correo señalado, con la alerta respectiva.

```
gabriel@Gabriel-PC:~/USM/Prog de Sist/Tarea 2$ ./mds 200.1.17.4 www.google.cl 90
 30 smtp.gmail.com gabriel.ahumada.gonzalez@gmail.com AGdhYnJpZWwuYWh1bWFkYS5nb2
56YWxlekBnbWFpbC5jb20Ac3Ryb2tlczEy
> > DNS is OK
Website is OK
CPU is OK
Memory is overload
```

Figure 1: Ejecución del programa con una sobrecarga en la memoria.

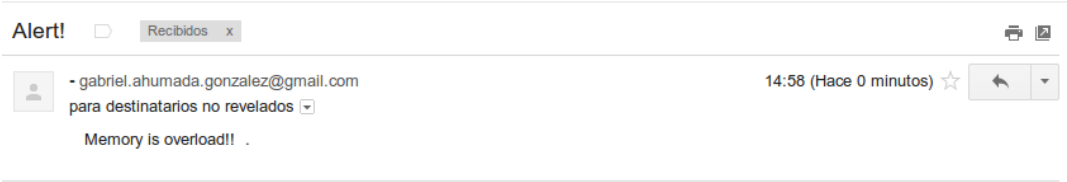


Figure 2: Recepción de la alerta en el correo.