



Error en codificación

Objetivo: Aprender a partir de un error de codificación. Analizar código C para encontrar un error (bug) en el programa, el cual provoca un comportamiento muy peculiar del programa en tiempo de ejecución (runtime). Vincular este comportamiento con los conceptos teóricos vistos en la unidad I y II.

Introducción:

El programa `server1.c` es un servidor que utiliza sockets tcp/ip para comunicarse con sus clientes. Los clientes envían mensajes al servidor y este los procesa. La función `main()` de `server1.c` llama a la función `server()` que se encuentra en `srvp.c` (dentro de esta función se encuentra el bug). Una de las primeras acciones dentro de la función `server()` es abrir el archivo `srvp.log` de output y grabar mensajes tales como:

```
...  
fprintf(data.fp, "main():inicio\n");  
...
```

Sin embargo, cuando se ejecuta el programa, el archivo `srvp.log` es creado pero el mismo permanece vacío, no se guarda ningún contenido dentro de él y las instrucciones que graban parecen haber sido ejecutadas. La tarea es tratar de develar este “misterio”.

Sugerencia:

Descomprimir en un directorio cualquiera el contenido del archivo `Unidad_II.tp.opcional.zip`, el mismo está formado por los siguientes archivos:

```
clip.h  
server1  
server1.c  
srvp.c  
srvp.h
```

Compilar el programa en un entorno linux, utilizando la sentencia:

```
gcc -Wall -o server1 server1.c srvp.c
```

Comprobar todos los archivos que hay en el directorio actual:

```
ls -l
```

Ejecutar el programa:

```
./server1
```

Podrá obtener una salida como esta si lo ejecuta desde una consola bash:

```
main():inicio  
Segmentation fault
```

Uno de los editores más usados para hacer programas C desde un entorno gráfico en Linux se llama SciTe, el mismo permite editar los programas, compilarlos y ejecutarlos desde una pequeña consola incorporada dentro del editor. Si ejecutamos este programa dentro de una consola SciTe, podemos obtener la siguiente salida (con signo `>` se indica la salida de la consola):



```
./server1  
>./server1  
>Exit code: 0 Signal: 11
```

Utilice el libro “Sistemas Operativos” de William Stallings, busque en el mismo la tabla de señales (signals) de UNIX o Linux y verifique de qué se trata la señal numero 11.

Con estos datos ya debemos tener una pista acerca del tipo de error que esta dando este programa en tiempo de ejecución.

Comprobar nuevamente todos los archivos que hay en el directorio actual:

```
ls -l
```

Si desea ver en la consola el contenido de un archivo, puede usar el comando `cat`:

```
cat srvp.log
```

Utilice su editor preferido para modificar la función `server()` y agregar mensajes, comentar líneas de código, etc.. Modifique el código y recompile el programa varias veces para tratar de determinar hasta que punto se ejecuta este programa. Observará que su comportamiento no parece ser muy lógico.

Una vez que logre una correcta ejecución del programa, puede detenerlo presionando la combinación de teclas: `Ctrl-C`

Responda:

1. ¿Qué tipo de señal es la número 11?
2. ¿Qué tipo de instrucción dentro de este programa podría causar esta señal?
3. ¿El programa se detiene porque así lo indica su código o bien hay una intervención del sistema operativo?
4. Si logró encontrar el bug que estamos buscando, ¿Cuál es la línea de código dentro de la función `server()` que provoca este problema? ¿Cuál fue el error del programador? A la luz de lo estudiado en las unidades I y II, ¿Por qué cree Ud. que se produce este comportamiento? ¿Con qué conceptos teóricos se vincula este comportamiento?
5. Discuta con sus compañeros y profesor las conclusiones derivadas de este problema.
6. Retomar esta discusión luego o durante el estudio de la unidad V, Gestión de Memoria, allí seguramente encontraremos otros elementos vinculados con este problema

Luego de la Unidad V - Gestión de Memoria Responda:

7. ¿Qué requisito de la Gestión de Memoria no se ha cumplido?
8. ¿Quién detectó este problema?
9. ¿De qué forma se detectó este problema?
10. ¿Qué acciones se tomaron al cabo ante la ocurrencia del problema?