

Arquitectura de sistemas

Abelardo Pardo

University of Sydney
School of Electrical and Information Engineering
NSW, 2006, Australia
Autor principal del curso de 2009 a 2012

Iria Estévez Ayres

Damaris Fuentes Lorenzo

Pablo Basanta Val

Pedro J. Muñoz Merino

Hugo A. Parada

Derick Leony

Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática
Avenida Universidad 30, E28911 Leganés (Madrid), España



Tabla de contenidos

[14.1. Actividades](#)

[14.1.1. Uso del depurador](#)

[14.1.2. Depurar un programa en C](#)

[14.1.3. Detección y corrección de errores en programas con el depurador](#)

[14.1.4. Detección y corrección de errores en programas con el depurador](#)

[14.1.5. Emergencia: "Necesitamos la clave para entrar"](#)

14.1. Actividades

14.1.1. Uso del depurador



Recursos

- Fichero `gdb_use.c` en la subcarpeta `Debugger` de tu carpeta de trabajo compartida.



Plan de trabajo

Lee el documento del depurador que se te ofrece como recurso. Haz en una pequeña hoja un resumen de los comandos más importantes. Esta hoja puedes tenerla a lo largo de la asignatura para poder depurar con más agilidad.

Asegúrate de que eres capaz de:

- Detener un programa en un punto cualquiera del código.
- Sabes utilizar las funciones de "Next" y "Step".
- Poner una condición en un punto de parada y verificar que el depurador se detiene sólo si la condición es cierta.

14.1.2. Depurar un programa en C



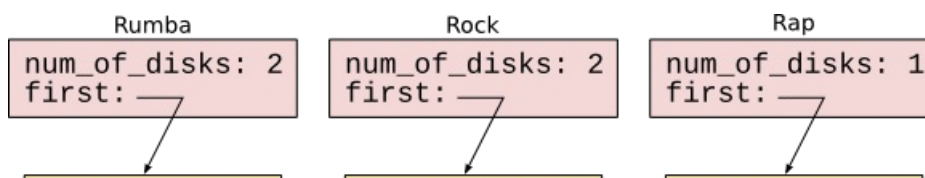
Recursos

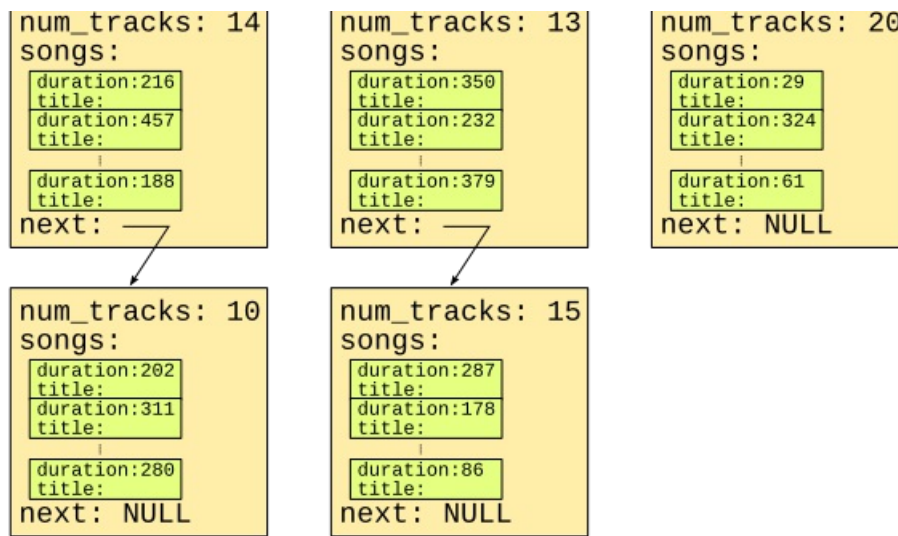
- Carpeta `Music_data` en tu espacio compartido de Subversion.



Plan de trabajo

El programa `my_music.c` almacena información sobre una colección de discos tal y como se muestra en la siguiente figura.





Las variables globales `rap`, `rock` y `rumba` son estructuras de tipo `Category` con el número de discos y una lista encadenada de ellos. Cada disco es una estructura con el número de canciones, un array de información sobre cada canción y el puntero al siguiente disco. La información de una canción es su duración y un puntero al título.

El programa (ver función `main`) primero imprime el número de discos para cada categoría, luego la duración de todas las canciones de una categoría (mediante la función `Duration`), y finalmente el título de la última canción de cada disco en cada categoría mediante la función `print_title_last_track_in_each_disk`.

Puedes ignorar las funciones estáticas del fichero. Son correctas y lo único que hacen es cargar y descargar los datos en la estructura.

1. Compila y ejecuta el programa `my_music.c` (recuerda compilarlo con la opción `-g`). Contiene tres errores. Debido a ellos, verás que los mensajes que imprime por pantalla son incorrectos. Por ejemplo, la duración de los discos de la categoría "rap" es incorrecta, y los mensajes que muestran el título de la última pista de cada disco son incorrectos.
2. Utilizando el depurador, analiza la ejecución y arréglos (igual es conveniente hacerte una copia del programa, para volver al fichero original si lo necesitas). Para ello, puedes localizar las tres variables globales y las dos funciones `Duration` y `print_title_last_track_in_each_disk`. Con el depurador vete hasta que se imprime la duración de la colección de rap y ejecuta paso a paso cómo se calcula. Repite el procedimiento para la primera llamada a la función `print_title_last_track_in_each_disk`.

Recuerda que las funciones estáticas del fichero son correctas y lo único que hacen es cargar y descargar los datos en la estructura.

3. La forma de saber si la ejecución es la correcta es porque la salida del programa debe ser (presta atención a la última línea):

```

--- Welcome to the Music Statistics program ---
    Analyzing your music collection...

2 disks in the rumba category
2 disks in the rock category
1 disks in the rap category

Duration of rumba songs: 6019
Duration of rock songs: 7044
Duration of rap songs: 4639
Duration of all songs: 17702

```

```
Title of last track in rumba disks:
  Disk 1: vestido de flores
  Disk 2: nina

Title of last track in rock disks:
  Disk 1: god save the queen
  Disk 2: now im here

Title of last track in rap disks:
  Disk 1: curtains close skit

--- PROGRAM IS FIXED! COMMIT YOUR CHANGES ---
```



Evaluación

Cuando termines, sube el código al SVN y enséñaselo al profesor. Es conveniente que vayas apuntando los pasos para explicarle al profesor cómo has solucionado los errores y de qué manera te ha ayudado a hacerlo el depurador. El profesor además puede comprobar que has entendido el manejo del depurador con algunas preguntas.

14.1.3. Detección y corrección de errores en programas con el depurador



Recursos

- Documento [El programa depurador o debugger](#)
- Fichero `strings_in_table.c` en la subcarpeta `Debugger` de tu carpeta de trabajo compartida.



Plan de trabajo

1. El programa [strings_in_table.c](#) inserta N strings en una tabla de M entradas tal que $N \gg M$. La forma de hacerlo consiste en mantener una lista encadenada de strings en cada elemento de la tabla. La función `main` primero inserta N strings, luego borra algunos de ellos, y finalmente imprime el contenido de la tabla.

El programa contiene cuatro errores. Uno de ellos impide su terminación. Dos de ellos permiten terminar al programa pero con resultado manifiestamente incorrecto. Finalmente, el cuarto error hace que el programa utilice más memoria de la imprescindible (580 bytes) y produce fugas de memoria.

Utilizando el depurador, obtener una versión **correcta** de este programa (que termine, imprimiendo el contenido de la tabla de forma correcta, sin fugas de memoria y que utilice únicamente 580 bytes de memoria) manteniendo su funcionalidad.

2. En la sesión de depuración del programa anterior ver el efecto que produce cuando se ejecuta el comando **print** y su argumento es una llamada a la función **print()**:

```
(gdb) p print(table, 5)
```

3. En la sesión de depuración del programa anterior, cuando la tabla ya tiene todos los elementos

insertados, seleccionar una de sus posiciones, y mediante el comando **print** mostrar por pantalla todos los strings almacenados en una lista encadenada.

14.1.4. Detección y corrección de errores en programas con el depurador



Recursos

- Documento [El programa depurador o debugger](#)
- Fichero `tablealloc.c` en la subcarpeta `Debugger` de tu carpeta de trabajo compartida.



Plan de trabajo

El programa contenido en el fichero `tablealloc.c` crea dos tablas de `N` elementos. Una de ellas almacena estructura de datos `name_and_number`, y la otra enteros. La función `main` rellena ambas tablas con datos y luego los imprime por pantalla. Si se compila y ejecuta, se puede comprobar que los resultados que imprime son incorrectos. Mediante la utilización del depurador, encontrar el error y modificar el código para que imprima los valores correctos.

14.1.5. Emergencia: "Necesitamos la clave para entrar"



Recursos

- Carpeta `Crack_passwd` en tu espacio compartido con Subversion.



Plan de trabajo

Llega una llamada a SAUCEM S.L. de un cliente en Singapur que tiene una aplicación instalada en los móviles corporativos (unos 100). Esta aplicación requiere un identificador de usuario con 9 dígitos y una clave para entrar. Las claves son fijas porque las genera el propio programa en base a los 9 dígitos de entrada. El cliente ha extraviado el documento con la correspondencia entre identificadores y claves, por lo que ahora no tiene forma de saber cómo entrar en esa aplicación en ninguno de sus 100 móviles.

Por suerte, dispones del fichero `checkentry.c` en la carpeta `Crack_passwd` que contiene la porción de código encargada de leer el identificador, leer una clave y compararla con la generada. El plan es el siguiente:

- Compila y ejecuta el programa `checkentry.c` en tu carpeta `Crack_passwd`. Utiliza tu NIA para estas pruebas.
- No se puede modificar el código. Repito, sin modificar el código.
- Utiliza el depurador para analizar lo que hace el programa y obtener el conjunto de letras de la clave generada.
- Te vendría bien modificar el valor de alguna variable mientras el programa está ejecutando utilizando el depurador. Averigüa cómo se hace tal cosa.
- Una vez que tengas la clave para entrar con tu NIA, demuéstrale la ejecución al profesor.



Evaluación



Subir un fichero con nombre `claves.txt` en la carpeta `Crack_passwd` de tu espacio de trabajo que contenga las claves que has adivinado.