



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

Un enfoque para la enseñanza de la depuración de errores en las asignaturas de programación

Sergio Luján-Mora

Un enfoque para la enseñanza de la depuración de errores en las
asignaturas de programación

Contenido

- Introducción
- Cómo se enseña a depurar errores en los libros
- Características de la asignatura
- Carencias de los alumnos
- Aprender a depurar
- Conclusiones

Contenido

- **Introducción**
- Cómo se enseña a depurar errores en los libros
- Características de la asignatura
- Carencias de los alumnos
- Aprender a depurar
- Conclusiones

3

Introducción

- Depuración y corrección de errores: los grandes olvidados
 - Falta de tiempo
 - Complejidad del proceso
 - ...
- “Errar es de humanos”
- Parece inherente (y se asume) la existencia de errores

4

Introducción

- Algunos datos:
 - Software “normal” → 15 errores / 1.000 líneas
 - MS Windows 2000 → 63.000 fallos (40 millones de líneas)
 - NIST (2002) → Coste anual 60.000 millones \$
 - 60% desarrolladores involucrados en resolver errores evitables
 - 47 días (desarrollar), 150 días (verificar y depurar)

5

Introducción

- Falta:
 - Prestar más atención a la depuración y corrección de errores
 - Incidir en obtener software de calidad
- En este artículo:
 - Enfoque empleado en “Programación y estructuras de datos” (PED)

6

Contenido

- Introducción
- **Cómo se enseña a depurar errores en los libros**
- Características de la asignatura
- Carencias de los alumnos
- Aprender a depurar
- Conclusiones

7

Cómo se enseña a depurar errores en los libros

- No se enseña:
 - Bibliografía sobre C/C++
 - Un par de páginas dedicadas a los tipos de errores (compilación, enlazado, etc.)
 - Se remite al lector al manual del compilador

8

Contenido

- Introducción
- Cómo se enseña a depurar errores en los libros
- **Características de la asignatura**
- Carencias de los alumnos
- Aprender a depurar
- Conclusiones

Características de la asignatura

- Programación y estructuras de datos (PED):
 - Troncal, 2º curso, II, ITIG, ITIS
 - 9 créditos anuales: 4,5 + 4,5
 - Prerrequisitos:
 - Fundamentos de Programación I (FPI)
 - Fundamentos de Programación II (FPII)
 - Descriptor:
 - Estructuras de datos y algoritmos de manipulación
 - Tipos abstractos de datos
 - Diseño recursivo

Características de la asignatura

- FPI: principios de algoritmia, estructuras de programación y representación de datos
- FPII: C++ sin orientación a objetos
- PED:
 - Estructuras de datos básicas (pila, lista, árbol binario, etc.) y avanzadas (árbol AVL, árbol 2-3-4, grafo, etc.)
 - Recursividad
 - Prácticas con C++

11

Contenido

- Introducción
- Cómo se enseña a depurar errores en los libros
- Características de la asignatura
- **Carencias de los alumnos**
- Aprender a depurar
- Conclusiones

12

Carencias de los alumnos

- Carencias o falta de habilidades:
 - No comprenden las herramientas de desarrollo (compilador, enlazador, etc.)
 - No distinguen los distintos tipos de errores (de diseño, lógicos, sintácticos, etc.)
 - No saben cómo emplear un depurador
 - No son sistemáticos
 - No tienen paciencia

13

Carencias de los alumnos

- Comportamiento frente a los errores en el código:
 - No le dan importancia
 - No leen los mensajes de error
 - Profesor = depurador
 - “Echan balones fuera”

14

Contenido

- Introducción
- Cómo se enseña a depurar errores en los libros
- Características de la asignatura
- Carencias de los alumnos
- **Aprender a depurar**
- Conclusiones

15

Aprender a depurar

- Recetas y consejos
- Objetivos:
 - Sea consciente de la importancia de los errores
 - Aprenda técnicas y habilidades relacionadas con la depuración de errores

16

Aprender a depurar

1. No todos los errores son iguales: “Primero conoce a tu enemigo”

- Errores de diseño
- Errores lógicos
- Errores sintácticos
- Avisos de compilación
- Errores de enlazado
- Errores de ejecución

17

Aprender a depurar

2. Mentalízate: “Seguro que vas a tener errores, prepárate con antelación”

- Los errores son MUY IMPORTANTES:
 - Máquina de radioterapia (1985-1987)
 - Ariane 5 (1996)
 - Mars Climate Orbiter (1999)

18

Aprender a depurar

3. Inténtalo tú mismo: “Pide ayuda cuando lo hayas intentado todo”

- Programar requiere ciertas habilidades que sólo se pueden lograr con la práctica

19

Aprender a depurar

4. Estado mental: “Tómate tu tiempo”

- Depurar es una actividad mental que requiere una gran concentración
- La fatiga y la frustración impiden localizar un error obvio → Es mejor posponer la búsqueda

20

Aprender a depurar

5. Localiza el error: “¿Dónde está el problema?”

- Mensajes “por aquí pasa” o “a = ”
- Compilación condicional → Versión de depuración y versión final

21

Aprender a depurar

6. Prueba y error: “¡¡Peligro!!”

- Una de las técnicas más usadas: modificar, compilar y probar (ejecutar)
- Mucho tiempo
- ¿Seguro que los has resuelto?
- Nuevos errores

22

Aprender a depurar

7. Cuidado con los mensajes de error:

“Escucha al compilador”

- El primer mensaje es un error real, el resto puede que no
- El origen de un error se puede encontrar en cualquier línea por encima de la línea señalada

23

Aprender a depurar

8. Tu amigo el depurador: “No te va a morder”

- Desconocimiento: conocen su manejo, pero no cómo aprovecharlo
- “Sin embargo, con frecuencia los depuradores son difíciles de utilizar y de comprender, por lo que son rara vez utilizados por los estudiantes de un primer curso de programación” (Deitel y Deitel)

24

Aprender a depurar

9. Conjunto de ficheros de prueba:

“¿Cuántas veces lo has probado?”

- Verificar automáticamente y de forma exhaustiva el correcto funcionamiento

25

Aprender a depurar

10. Contempla todos los posibles

casos: “¿Y si ...?”

- Suponer que una situación es improbable (o imposible) es fuente de muchos errores

26

Aprender a depurar

11. Los errores son evitables: “Si no haces nada, fallará”

- Industria software: verificar un programa cuando está terminado
 - Costoso: consume mucho tiempo y recursos
 - Ineficiente: no asegura que esté libre de errores
- Abordar el problema desde el principio

27

Aprender a depurar

12. Errores más comunes: “Ya sé quién eres”

- Errores típicos cuando se comienza a programar con C++:
 - Sobre el fichero makefile y la compilación
 - Sobre las directivas de inclusión
 - Sobre las clases
 - Sobre la sobrecarga de los operadores
 - Sobre la memoria
 - Sobre las cadenas
 - Varios

<http://www.dlsi.ua.es/~slujan/files/errores.pdf>

28

Un enfoque para la enseñanza de la depuración de errores en las asignaturas de programación

Confundir la declaración de una función amiga (friend) con los modificadores de visibilidad. Ejemplo:

```
class UnaClase {  
    friend:  
        int funcionAmiga(void);  
    public:  
        UnaClase();  
        ~UnaClase();  
        ...  
};
```

Mensaje de error:

```
In file included from unaclass.cc:1:  
unaclass.h:9: parse error before '('
```

Solución: El modificador friend se tiene que poner a cada función que sea amiga.

```
class UnaClase {  
    friend int funcionAmiga(void);  
    public:  
        UnaClase();  
        ~UnaClase();  
        ...  
};
```

29

Un enfoque para la enseñanza de la depuración de errores en las asignaturas de programación

Contenido

- Introducción
- Cómo se enseña a depurar errores en los libros
- Características de la asignatura
- Carencias de los alumnos
- Aprender a depurar
- **Conclusiones**

30

Conclusiones

- La depuración y corrección de errores no recibe la atención que debiera
- Recetas y consejos empleados en PED
- Trabajos futuros:
 - Estudiar el rendimiento académico de los alumnos
 - Desarrollar una metodología de depuración

31



Un enfoque para la enseñanza
de la depuración de errores en
las asignaturas de programación

Sergio Luján-Mora

32