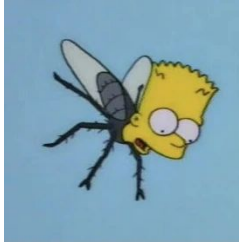
Debugging



¿ Qué es un bug ?

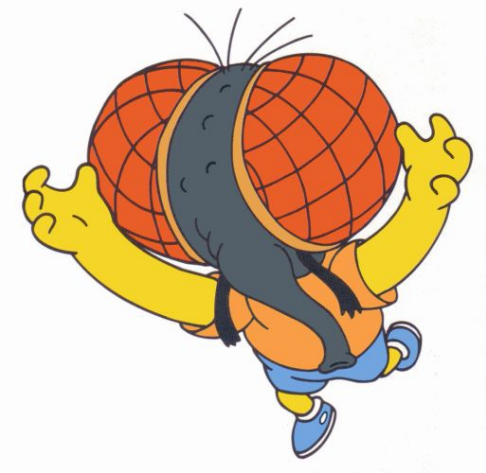


Un bug es un error en nuestro código que provoca comportamientos indeseados.

Generalmente tienen dos causas:

- Error humano
- Factores externos:
  - Diferencias entre SO.
  - Comportamiento no determinado: Es una situación en la que el estándar del lenguaje **no define qué debería pasar**, por lo tanto, **el compilador puede hacer cualquier cosa.**
    - Acceder a memoria no inicializada.
    - Acceder fuera del rango de vectores.
    - Dividir por cero.

Tipos de errores



- Errores de sintaxis
  - Ejemplo: falta de punto y coma, paréntesis mal cerrados
- Errores de compilación
  - Tipos de datos incompatibles, funciones no declaradas
- Errores de lógica
  - El programa compila, pero no hace lo correcto (p. ej., usar <= en lugar de <)
- Errores de ejecución (runtime)
  - División por cero, acceso a memoria inválida, segmentation fault

Debugging



El proceso de debugging (también llamado depurar o debbugear) es el acto de eliminar bugs de nuestro código.

Hay dos formas de debuggear nuestro programa:

- Usando prints:
  - Es la forma más simple, seguro es la que usaron hasta ahora.
  - Tiene varios problemas. Es muy limitado y nos obliga a cambiar nuestro código.
- Usando un debugger:
  - Son herramientas dedicadas y construidas específicamente para debuggear, en clase usaremos una llamada GDB.
  - Permiten hacer mucho más que imprimir variables. Además, no hace falta editar el código.



GDB



¿ Qué es GDB ?

GDB es una herramienta esencial para depurar programas en C.

Permite:

- Detener la ejecución en puntos específicos (breakpoints).
- Ver el contenido de variables.
- Analizar la pila de llamadas.
- Observar cambios en variables.

¿ Cómo se usa ?

Primero debemos compilar nuestro programa y agregar el flag -g:

gcc -g archivo.c -o programa

“To tell GCC to emit extra information for use by a debugger, in almost all cases you need only to add -g to your other options.”

Luego ejecutamos nuestro programa anteponiendo el comando “gdb”:

gdb ./programa

Comandos...

break <nombre_funcion>	# Coloca break point. O break <arch>:<linea>
run	# Ejecuta el programa hasta encontrar un break
start	# Ejecuta el programa deteniéndose en el inicio
next	# Ejecuta la siguiente línea
step	# Ingresa a una función
print <var>	# Muestra el valor de la variable
backtrace	# Muestra la pila de llamadas
watch <var>	# Vigila un cambio en una variable
display <var>	# Muestra el valor de la variable por cada next
continue	# Sigue la ejecución
finish	# Termina la función actual
quit/exit	# Sale de GDB

TUI (Text User Interface)

Es una interfaz de usuario textual que te permite ver el código fuente mientras debugueás, todo desde la consola.

Op1:

Ejecutar ***“gdb ./programa”***
Luego ***“Ctrl + x”*** y luego ***“A”***

Op2:

gdb -tui ./programa