

Video3 (1)

Uso mínimo del GDB

Conceptos Principales

1. a
2. b
3. c

Notas

- **Propósito del video:** Introducir el uso básico de **GDB**, el debugger de C, para seguir la ejecución de un programa paso a paso y detectar errores, especialmente los relacionados con manejo de memoria.
- **Conceptos clave:**
 - **Debugger:** herramienta que permite ejecutar programas línea por línea, inspeccionar variables y entender el flujo del código.
 - Importancia en C: ayuda a detectar errores que no aparecen en tiempo de compilación; especialmente útil para errores de memoria como **segmentation fault**.
- **Preparación para usar GDB:**

- Compilar el programa con la opción **g**:

```
gcc -g archivo.c -o ejecutable
```

- Esta opción incluye información de depuración en el ejecutable.
- Sin **g** se puede ejecutar, pero GDB no puede mapear líneas de código ni variables.

- **Comandos básicos de GDB:**
 - **break <línea o función>**: define un **punto de quiebre** donde la ejecución se detendrá.
 - **run** o **r**: inicia la ejecución hasta el primer breakpoint.
 - **list** o **l**: muestra el código fuente alrededor de la línea actual.
 - **next** o **n**: avanza a la siguiente línea de código **sin entrar en funciones**.
 - **step** o **s**: avanza a la siguiente línea de código **entrando dentro de funciones**.
 - **print <variable>**: imprime el valor actual de una variable.
 - **display <variable>**: imprime automáticamente el valor de la variable cada vez que se ejecuta un comando siguiente.
- **Ejemplo de flujo:**
 1. Colocar un **break** en la función donde se sospecha el error.
 2. Ejecutar **run** para iniciar el programa hasta el breakpoint.
 3. Usar **next** para avanzar línea por línea y observar cambios en variables.
 4. Usar **print** o **display** para monitorear valores.

5. **step** se puede usar si se quiere ingresar dentro de una función.

- **Notas importantes:**

- GDB permite **visualizar cómo cambian los valores** durante la ejecución y facilita detectar errores lógicos o de memoria.
- **0** en C se interpreta como **false**, cualquier otro valor como **true**.
- Es recomendable practicar con los ejercicios para familiarizarse con comandos y flujo de depuración.

- **Recomendación final:**

- Este video muestra el uso mínimo de GDB; más adelante se profundizará en cómo utilizarlo para errores complejos y manejo de memoria.
- Mantener comentarios y documentación interna para entender mejor el flujo del programa durante la depuración.