Unidad 7

Depurador

Contenido

- Introducción
- Arquitectura de un Depurador
- Elementos de Hardware y Software
- Control de Ejecución
- Interrupciones y Ejecución Paso a Paso
- Inspección de Variables

Introducción

¿Qué es un Depurador?

- Los Depuradores o *Debuggers* son herramientas fundamentales en el desarrollo de software, un Depurador es una herramienta que permite ir dando seguimiento y remover algunos bugs o errores de un software
- Los Depuradores son aplicaciones que requieren algoritmos complejos y varias estructuras de datos para completar sus tareas además de necesitar una relación muy compleja con ciertas funciones del Sistema Operativo

Principios Básicos de un Depurador

- Se consideran cuatro principios básicos para el diseño y desarrollo de un depurador:

  - El principio de Heisemberg
  - La confianza en un Depurador
  - El despliegue de información
  - Uso adecuado de la tecnología

El Principio de Heisemberg

- Se refiere a que el Depurador no debe afectar en el proceso de depuración
- Esto significa que depurar una aplicación no debe afectar su funcionamiento, un Depurador debe ser No Intrusivo
- En el momento en que un Depurador es controlado por el mismo sistema operativo que controla la aplicación a depurar, éste principio se vuelve complicado de cumplir
- Por esta razón los diseñadores y desarrolladores de Sistemas Operativos y de Depuradores prestan gran atención al Principio de Heisemberg

Confianza en la Depuración

- Un depurador siempre debe proporcionar información confiable durante una depuración
- El depurador nunca debe proporcionar información incorrecta al usuario ya que lo podría guiar en información incorrecta
- Esta información se relaciona principalmente con el valor de las variables durante el proceso de depuración

Despliegue de Información

- Se refiere a la información más importante que requiere el programador durante el proceso de depuración que es la información del contexto del programa

- El contexto puede incluir varios tipos de información como el código fuente, información de las variables, de los hilos y otros procesos

¿Qué Información Interesa?

- Se desea que el depurador indique en qué línea de código en la que se está trabajando

- También es deseable que el depurador se detenga cuando detecte que se ha generado algún error indicando la línea en la que sucedió

- Se quiere mostrar una lista de las operaciones que se han realizado hasta llegar a esa línea

¿Qué información interesa?

- El valor de la variables locales y globales también representa una información crítica que debe ser desplegada

- También se desea tener información de las instrucciones que se van a ejecutar y la información en los registros, esto se conoce como una vista de CPU

Uso Adecuado de la Tecnología

- La tecnología de un depurador debe estar al mismo nivel que la tecnología utilizada para crear entornos de desarrollo y sistemas operativos

- El Sistema Operativo debe proporcionar la infraestructura necesaria para proporcionar las últimas tecnologías a los desarrolladores de Depuradores

Clasificación de los Depuradores

- Depurar es una actividad general, pero cada aplicación requiere un uso de un depurador especial para encontrar y eliminar algún posible bug

Depuradores a Nivel Código Fuente y Máquina

- Un depurador a nivel máquina es aquel que mapea las instrucciones máquina producidas por un compilador o ensamblador a su equivalente en instrucciones de código fuente

- De esta manera, el desarrollador tiene la impresión de que el depurador está ejecutando directamente el código fuente

- Un depurador debe proporcionar información sobre los registros y actividad en memoria (nivel máquina) además de la información sobre los valores de las variables (nivel código)

Depuradores Separados e Integrados

- Anteriormente los depuradores trabajaban de manera independiente a los entornos de desarrollo

- Actualmente la mayoría de los entornos de desarrollo integrados (IDE) contienen integrado su propio depurador, esto facilita la tarea del desarrollador

- En los IDE con un entorno gráfico, el Depurador también es tratado con una interfaz GUI, lo que permite un mejor manejo y entendimiento de la información

Depuradores en Lenguajes Interpretados

- Los lenguajes interpretados se utilizan en entornos donde se busca una gran productividad orientada al negocio con un gran uso de bases de datos

- El uso de un entorno interpretado proporciona grandes ventajas a un Depurador ya que puede proporcionar una retroalimentación casi inmediata sobre los efectos de los cambios realizados

- El intérprete proporciona un entorno seguro y protegido en donde la aplicación y el depurador pueden ejecutarse

Depurador de Kernel y de Aplicación

- La depuración del núcleo o *kernel* de un Sistema Operativo se utiliza durante el desarrollo de diversos componentes, por ejemplo manejadores o *drivers*

- Un depurador de kernel, contrario a una de aplicación, involucra dos máquinas: la máquina en la que se está desarrollando y la máquina para la que se está desarrollando

Arquitectura de un Depurador

Elementos de la Arquitectura

- Una arquitectura general para un Depurador está formada por los siguientes elementos:
  - Interfaz de Usuario
  - Núcleo del Depurador
  - Interfaz con el Sistema Operativo

Interfaz de Usuario

- Los componentes que se pueden encontrar de manera general en una interfaz de usuario de un Depurador son:
  - Vista del Código Fuente
  - Vista de los Procedimientos
  - Vista de los Puntos de Interrupción
  - Vista de CPU
  - Vista de las Variables

Vista del Código Fuente

- Es el punto de mayor interés para el desarrollador al momento de la depuración ya que da la impresión al usuario que es el Depurador el que ejecuta el programa
- Normalmente se remarcan las sentencias ejecutables además de indicar en donde se tienen puntos de interrupción
- Se indica en que parte del código se encuentra el proceso de depuración
- Es responsabilidad del Depurador el mapear la ejecución en la máquina a esta vista

Vista de los Procedimientos

- Proporciona una representación de la función o procedimiento
- Se forma una cola de *frames* que representan llamadas a una función incluyendo normalmente el nombre y parámetros
- Uno de sus usos más importantes es para determinar si se llegó o no a la función correcta o determinar en que función el programa pudo haberse detenido por completo

Vista de los Puntos de Interrupción

- Esta vista da un panorama general de los puntos de ruptura que el usuario ha colocado

- Los puntos de ruptura son herramientas críticas que permiten al programador controlar la ejecución de un programa indicando en que momento se detendrá

Vista de los Puntos de Interrupción

- Se presenta un estatus de activo, inactivo o no verificado: uno activo interrumpirá el programa al alcanzarlo, uno inactivo puede activarse en algún momento pero no detendrán el programa y los no verificados son los que aparecen en parte del código que no se ha cargado en el espacio de direcciones

Vista de CPU

- El principal objetivo de esta vista es mostrar al programador un panorama de lo que ocurre a nivel máquina
- Entre las vistas que puede contener se encuentran un panel en ensamblador, un panel con los valores en los registros, indicadores de memoria y banderas
- En algunos casos, los valores de los registros pueden ser modificados en esta vista

Vista de las Variables

- Se presenta a un nivel simbólico y permite conocer el valor con el que se están cargando las variables a lo largo de la ejecución
- El conocer el valor de las variables, es unas de las maneras más comunes de detectar algún error en algún momento de la ejecución

Núcleo del Depurador

- Se trata de la capa que da servicio a las capas de interfaz de usuario. En esta capa es donde se lleva a cabo el proceso de control, la aplicación que se quiere depurar es un proceso para el sistema operativo

- Está formado por los siguientes elementos: control del Proceso, Ejecución, Evaluador de Expresiones, Manejo de Tablas de Símbolos

Control de Proceso

- El proceso de depuración es para el Sistema Operativo un proceso, el Control de Proceso se encarga de iniciar el proceso o colocarlo junto a un proceso ya en ejecución.

- Al finalizar el proceso de depuración, el Control de Proceso le indica al Sistema Operativo que finalice el proceso

Manejo de Tablas de Símbolos

- La Tabla de Símbolos es parte del archivo que contiene el ejecutable y se utiliza para realizar un mapeo entre las instrucciones y las direcciones de las mismas

- También contiene información sobre las variables de tal manera que el Depurador pueda realizar un mapeo para localizarlas en memoria durante la depuración

Ejecución

- Esta sección puede controlar el proceso de depuración, basado en los mapeos de la tabla de símbolos, de tal manera que el usuario siente que está viendo el código fuente siendo ejecutado por el depurador

- Entre sus tareas se encuentran: realizar la depuración, depurar hasta el siguiente punto de ruptura, depurar las instrucciones paso a paso entre otras operaciones

Evaluador de Expresiones

- Es el proceso que utiliza el Depurador para evaluar variables, operaciones y llamar funciones según especificó el usuario

- Se comunica con la tabla de símbolos para buscar variables, lee la memoria del depurador para obtener los valores y utiliza la sección de Ejecución para ejecutar las funciones y presenta los resultados al usuario

Interfaz del Sistema Operativo

- La Principal función de la interfaz entre el Depurador y el Sistema Operativo es garantizar el principio de no intrusión de Heisemberg

- Una porción del Sistema Operativo proporciona las funciones básicas para crear un proceso de depuración

- El proceso no afecta a ninguna otra capa del Sistema Operativo o del hardware por lo que se satisface el principio de no intrusión

Elementos de Hardware y Software

Introducción

- Aunque parece un proceso que involucraría solamente al Sistema Operativo, hay ciertas funcionalidades que el hardware debe ofrecer para realizar un proceso de depuración
- Los elementos básicos que un Depurador necesita del hardware son:

1. Una manera de especificar un punto de interrupción
2. Un sistema de notificación que permita saber cuando algo importante ha sucedido
3. La capacidad de leer y escribir directamente a los registros de hardware cuando ocurra una interrupción

Mecanismos de Depuración

- El conjunto de elementos de soporte que el hardware debe ofrecer incluyen:
  - Soporte para Puntos de Interrupción
  - Soporte para Ejecución Paso a Paso
  - Detección de Errores
  - Soporte para Notificación

Soporte para Puntos de Interrupción

- Las interrupciones se implementan como una instrucción especial que provoca un cierto error en el sistema operativo, la cuál es manejada por un depurador
- La instrucción de interrupción depende de los registros de cada computadora, pero se coloca normalmente al final de un registro con una longitud lo más pequeña posible

Soporte para Puntos de Interrupción

- El depurador, gracias a ciertas rutinas que comparte con el Sistema Operativo, tiene la habilidad de leer y escribir en estos espacios

- Cuando se encuentra y ejecuta una de estas instrucciones de interrupción, el Sistema Operativo detiene el proceso y le informa al Depurador en dónde se detuvo, cuando se detuvo y por qué se detuvo

Soporte para Ejecución Paso a Paso

- Significa que el procesador ejecuta una sola instrucción máquina, la mayoría de los procesadores tienen un bit que controla ésta operación

- Este bit no causa que otros procesos se ejecuten paso a paso

- En algunos procesadores, se ha eliminado este bit, y la ejecución paso a paso se simula colocando puntos de interrupción después de cada instrucción

Detección de Errores

- Algunos errores son detectados por el procesador, como división entre cero, violaciones en el acceso a segmentos de memoria

- Debe ser posible especificar si se dejarán pasar algunos de los fallos que pudieran ocurrir

- Debe ser posible ver el efecto de un error antes de que realmente se produzcan

Soporte para Notificación

- Son notificaciones que se presentan cuando se modifican algunas direcciones en el espacio de memoria, esto sucede especialmente cuando se generan valores incorrectos o inesperados

- El hardware maneja estas notificaciones a través de registros muy específicos que contienen una dirección de inicio y longitud del programa

Cooperación entre S. O. y Depurador

- Para controlar un proceso de depuración, un Depurador necesita un mecanismo para notificar al sistema operativo sobre el programa que desea controlar
- El Depurador necesita ser capaz de modificar el código en memoria para alterar el flujo de instrucciones y colocar puntos de interrupción
- El Depurador necesita ser capaz de realizar sus funciones y ser notificado de que ha ocurrido alguna excepción
- El Depurador también debe conocer que direcciones de memoria debe leer para conocer los valores de las variables

Control de Ejecución

Inicializando el Programa Ejecutable

- Lo primero que realiza un depurador para controlar la ejecución es la creación del proceso de depuración ya sea creando uno nuevo o añadiéndolo a uno existente.
- Posteriormente se debe asegurar que se realizará la depuración bajo el control del Depurador a través de los siguientes pasos:
  - Creación de la Depuración
  - Unión con el Programa en Ejecución
  - Establecimiento de Puntos de Interrupción
  - Ejecutar la Depuración

Creación de la Depuración

- Lo primero que se realiza tras seleccionar el ejecutable a depurar es realizar llamadas al Sistema Operativo para iniciar la depuración
- El Sistema Operativo necesita crear el proceso de depuración o agregarlo a un proceso existente
- En cualquier caso el Sistema Operativo deberá controlar el proceso guiado por el Depurador
- El principal objetivo es modificar el comportamiento de las señales de excepción generadas por el proceso de depuración

Agregar a un Proceso en Ejecución

- Agregar la depuración a un proceso en ejecución es utilizado principalmente en depuración multiproceso en donde se debe depurar uno o todos los procesos que interactúan con el Depurador

Estableciendo Puntos de Interrupción

- Son instrucciones especiales que se colocan en una imagen ejecutable que provocan una excepción que detiene la ejecución inmediatamente

- Los Puntos de Interrupción son los mecanismos básicos utilizados por un depurador para controlar el proceso de depuración

Ejecutar la Depuración

- Una vez que comienza la ejecución, se presenta un intercambio en el contexto entre el Depurador y el Sistema Operativo
- El Depurador el el proceso activo y hace una llamada al Sistema Operativo para iniciar el proceso de depuración.
- El Sistema Operativo toma el control y coloca el proceso de depuración listo para ser ejecutado.
- El control regresa al Depurador y comienza la ejecución.
- El depurador puede estar intercambiando control de manera temporal según las necesidades del Sistema Operativo

Interrupciones y Ejecución Paso a Paso

Requisitos para Algoritmos de Puntos de Interrupción

- Un mecanismo para el manejo de Puntos de Interrupción necesita de los siguientes requerimientos:
  - El usuario debe insertar puntos de interrupción a nivel código
  - Pueden tenerse varios puntos de interrupción en cualquier parte del código
  - Los puntos de interrupción pueden tener asociadas condiciones para determinar si se ejecutan

Estructuras de Datos para Puntos de Interrupción

- Se requieren al menos dos niveles de representación de puntos de interrupción:
  - Puntos de Interrupción Lógicos
  - Puntos de Interrupción Físicos

- Los Puntos Lógicos son los establecidos por los usuarios mientras que los Puntos Físicos son aquellos relacionados a instrucciones ejecutables en la máquina

Puntos de Interrupción Lógicos

- Son responsables de representar un Punto de Interrupción resuelto (mapeado a una dirección física) o no resuelto (colocado en un módulo que no se ha cargado)

- Uno o más Puntos de Interrupción Lógicos pueden apuntar a un Punto de Interrupción Físico

Puntos de Interrupción Físicos

- Son los puntos encargados de contener la instrucción original (o una parte de ella) que será afectada por una interrupción.

- Si se elimina un punto de interrupción, esa parte de la dirección también es eliminada

Relación entre Puntos Físicos y Lógicos

- Uno o más puntos Lógicos apuntan a un Punto Físico, por esta razón es conveniente que se tenga una estructura para Puntos Físicos y una para Puntos Lógicos
- La comunicación entre Puntos Físicos y Lógicos es Bi direccional:
  - Los Puntos Lógicos se comunican con los Físicos cuando se establece, elimina o modifica un Punto en el código fuente
  - Los Puntos Físicos se comunican con los Lógicos en el momento que se ejecuta la depuración, se produce un mapeo entre direcciones físicas del SO con las direcciones físicas del Depurador y se buscan los Puntos Lógicos relacionados con esa dirección

Establecimiento y Activación

- El algoritmo básico para el establecimiento de Puntos de Interrupción es:
  - Entrada: nombre del archivo y línea o desplazamiento de un punto
  - Salida: Ubicación física o un error en la indicación
  - Método: Mapear a partir del nombre del archivo y el desplazamiento o línea usando una tabla de símbolos, un punto de interrupción lógica y un punto de ubicación físico

Algoritmo

- Paso 1:
  - Se realiza un mapeo dado el nombre del archivo y la línea solicitada en una dirección física
- Paso 2:
  - Crear un Punto de Interrupción Lógico con la información mapeada
- Paso 3:
  - Crear o incrementar la referencia si es que existía, una referencia física
- Paso 4:
  - El Punto de Interrupción ahora debe almacenar la instrucción original en esa localidad

Validación de Puntos de Interrupción

- La validación se refiere a asegurarse que todos los Puntos de Interrupción establecidos han sido mapeados a una dirección física en memoria

- Es necesario contar con un método de revisar que todos los Puntos de Interrupción Lógicos ya han sido mapeados a un Punto de Interrupción Físico

Ejecución Paso a Paso

- La Ejecución paso a paso es importante ya que los usuarios son capaces de observar la ejecución de un programa

- Esto permite analizar lo que sucede antes de que se llegue a un punto en donde se pudiera producir un error

- Para esto es necesario tener una serie de complejos mecanismos de control

Evaluación de Funciones

- Se pueden definir dos maneras de realizar la evaluación de una función, adentrarse en la función o simplemente saltarla

- La primera opción es conocida como saltar en (*step into*) mientras que la segunda solo se conoce como saltar (*step over*)

Step Into y Step Over

- En *Step Into*, la depuración salta a la primera instrucción ejecutable de una función

- En *Step Over*, la depuración salta a la primera instrucción ejecutable después de la llamada a función, una vez que ya la ejecutó

Ejemplo

```
int main() {
        int a,b;
        float c;
        printf("Introduce a y b:");
        scanf("%d , %d", &a,&b);
        c = calcular (a,b);
        printf("El resultado es: %f",c);
}

float calcular (int a, int b){
        c = pow (a,b);
        return c;
}
```

Inspección de Variables

Introducción

- Establecer los puntos de interrupción y la ejecución paso a paso es solamente la mitad del trabajo que debe realizar un depurador

- La otra parte de su función es analizar el contenido de las variables para analizar su comportamiento y poder detectar un posible error

Evaluación de Expresiones

- Un depurador puede desplegar los datos de las variables a través de una evaluación de expresiones que utiliza los mismos identificadores y sintaxis que se tienen en el programa fuente

- Esto hace que sea necesario contar con un analizador semántico

- Por esta razón, un compilador y un depurador trabajan de manera muy cercana e incluso comparten ciertas estructuras y funciones

Funciones y Evaluaciones

- Algunos lenguajes permiten el uso de una llamada a función como un operador, lo que dificulta su evaluación ya que una función no puede ser tratada de manera semántica

Evaluando una Función como Operando

- Para esto los Depuradores que permiten este tipo de evaluaciones realizan lo siguiente:
  - Especifican un punto de interrupción en donde se llama a la función
  - Se especifica un punto de interrupción en donde la función regresa el valor de retorno
  - Cuando se alcanza ese punto, se toma el valor y se regresa al punto en donde se invoca a la función ya con el valor regresado