

Inyección de errores para evaluación de aspectos no funcionales en sistemas

Ariel Sabiguero Andrés Aguirre
{asabigue, aaguirre}@fing.edu.uy

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, Uruguay

X Jornadas de Informática e Investigación Operativa

Agenda

- 1 Motivación
 - Errores en el Hardware y el Software
 - Desafíos para el testing tradicional
 - Inyección de defectos
- 2 Trabajo Realizado
 - Construcción de herramienta para inyección de defectos
 - Utilizando la herramienta construida
- 3 Resumen

Tipos de errores en el hardware

- El hardware al igual que el software puede fallar
- Los errores pueden ser categorizados por su duración:
 - Errores permanentes
 - Problemas de fábrica o mal uso
 - Errores transitorios
 - Variaciones de voltaje.
 - Interferencia electromagnética o radiación.

Algunos tipos de errores transitorios en el hardware

http://news.bbc.co.uk/hi/spanish/science/newsid_7337000/7337857.stm

BBC

Home News Sport Radio TV Weather Languages

Buscar

BBC MUNDO.com

Menos gráficos | Sólo texto | Escribanos | Ayuda

Miércoles, 9 de abril de 2008 - 22:06 GMT

Envíe esta nota por e-mail Imprima esta nota

Intel contra los rayos cósmicos

Redacción BBC Mundo

Intel, el mayor fabricante de microprocesadores del mundo, dio a conocer los detalles de la patente de un nuevo aparato para proteger las futuras generaciones de computadoras de la creciente amenaza que representan los rayos cósmicos.



De acuerdo con una investigación de Intel, los problemas se incrementarán a medida que aumentan los chips.

Debido a que las operaciones de los ordenadores se llevan a cabo a través de partículas cargadas, los impredecibles efectos de los rayos

NOTAS RELACIONADAS

- ▶ La nueva ola del Valle del Silicio 06 08 07 | Economía
- ▶ La computadora de hace 200 años 24 07 07 | Ciencia y Tecnología
- ▶ Alistan computadora de US\$100 23 07 07 | Ciencia y Tecnología
- ▶ La computadora más rápida del mundo 27 06 07 | Ciencia y Tecnología
- ▶ Una computadora por "guri", a la uruguaya 20 06 07 | Cultura y Sociedad

VÍNCULOS

- ▶ Intel

El contenido de las páginas externas sugeridas no es responsabilidad de la BBC.

EN ESTA SECCIÓN

- ▶ Tumba propia de un rey

OTROS IDIOMAS
ENGLISH

Portada
Internacional
América Latina
Mundo USA
Economía
Ciencia y Tecnología
Cultura y Sociedad

Participe
A fondo
Video
Aprenda inglés
Tiempo

Quiénes somos
Nuestros socios
SERVICIOS
¿Qué es RSS? | RSS

Ariel Sabiguero, Andrés Aguirre

Inyección de errores

¿Como responde el sistema ante fallas en el hardware?

- El hardware falla y puede conducir al software a estados no considerados.
- Un defecto en la memoria puede producir diferentes comportamientos en el sistema.
 - Pasar por desapercibido
 - Cambio en variable
 - Cambio en el código

Clasificación de errores en el software

Se clasifican según su facilidad de reproducción

- Errores en el diseño:
 - Facilmente identificables.
 - Corregibles durante la fase de pruebas.
- Errores internos intermitentes
 - Su condición de activación ocurre rara vez.
 - Difícilmente reproducibles.
 - Dependencia temporal entre varios eventos.
- Errores ocasionados por el envejecimiento de software
 - Tipicamente por agotamiento en los recursos del sistema operativo.

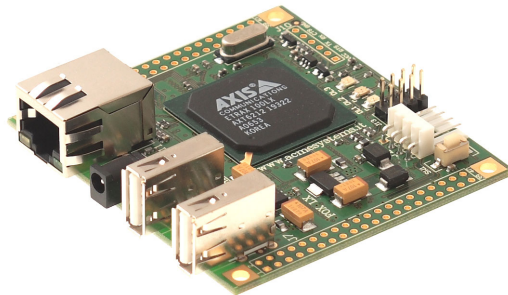
Catástrofes conocidas

Casos de defectos en el software que podrían haberse evitado

- Ariane 5
 - Error de en conversión.
 - Resultado de ángulo interpretado incorrectamente por sistema de navegación.
- Therac-25
 - Reemplazo del Therac-20 (implementado totalmente en hardware)
 - Therac-20 implementaba en hardware controles (interlocks) para evitar suministrar al paciente una dosis peligrosas.
 - Therac-25 no implementaba esos controles en hardware ni en software.
 - Condiciones en las entradas dadas por el operador en ciertos intervalos de tiempo producían una dosis incorrecta

Nuevas funcionalidades en dispositivos embebidos

- Aumento en funcionalidades de dispositivos embebidos.
- Utilización de sistemas operativos estándar para afrontar nuevas funcionalidades.
- GNU/Linux una opción frecuente.
 - Extensiones de tiempo real incluidas en el Kernel



¿Como podemos testear en estos escenarios?

- Las características de los sistemas embebidos nos imponen un bajo el nivel de intrusividad.
- Existen errores tanto en el hardware como en el software muy difíciles de reproducir.
- Técnicas de testing tradicional no aportan soluciones.
- ¿Que ocurre si no dispongo del código fuente?

¿Que es?

Definición:

Inyección de defectos: Es la inserción deliberada de defectos en un sistema informático o una simulación de defectos en el entorno del mismo, con el fin de conocer los fallos que pueden ocurrir cuando el sistema esté en explotación.

¿Que aporta la Inyección de defectos?

- Anticipar el comportamiento del sistema ante la presencia de defectos.
- Medir la tolerancia a fallas en el sistema.
- Util para testear los mecanismos de excepción y tratamiento de fallas.
- Técnica aplicable para validación de software de terceras partes, sin necesidad de tener acceso al código fuente.
- Medida de calidad.

¿Que vamos a predecir?

La simulacion de defectos mediante inyección, busca dar respuesta a preguntas del tipo:

Ante un defecto en el hardware/software:

- ¿Puedo confiar en el sistema de frenos del auto?
- Sabiendo que mi servidor web es vulnerable a ciertos defectos.
¿Con que probabilidad se van a responder a mis pedidos?
- ¿Puede el sistema de control de ascensores llegar a lastimar a alguien?.

Marco de trabajo

- Trabajo en conjunto InCo-CES.
- Colaboración con empresa española (MTP).
- Proyecto Iberoeka.

Requerimientos funcionales

- Inyección de defectos sobre:
 - Registros del CPU.
 - Memoria RAM.
 - Variables.
 - Funciones del programa.
 - Argumentos.
 - Valores de retorno.

Requerimientos funcionales

- Inyección de defectos sobre:
 - Llamadas al sistema operativo
 - Open
 - Read
 - Write
- Monitorizaciones
- Temporizaciones

Requerimientos no funcionales

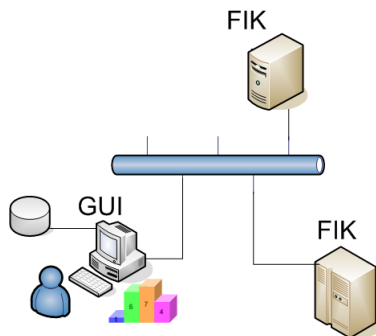
- Comunicación mediante protocolo TCP/IP y serie (rs-232c).
- Bajo nivel de intrusividad.
- Tamaño reducido.

Decisiones tomadas

- Sistema operativo GNU/Linux
- Implementación en C/C++
- Diseñado utilizando patrones de diseño
 - Fuerte uso del patrón observer para manejo de eventos asincrónicos.
 - Utilización del patrón state para manejar los cambios de comportamiento.
- Varios hilos de ejecución para permitir un alto grado de asincronismo.

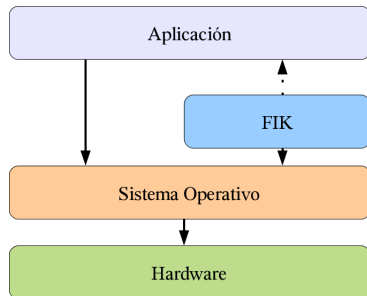
Arquitectura

- Dos componentes fundamentales
 - GUI
 - FIK



Funcionamiento General

- La aplicación accede normalmente a los recursos del sistema mediante los servicios del Sistema Operativo.
- El FIK accede de forma independiente de la aplicación a las abstracciones del hardware que el Sistema Operativo provee.
- Se modifica el comportamiento del sistema bajo prueba sin modificar la aplicación a probar.



Protocolo implementado

- El protocolo permite especificar:
 - Aplicación a probar.
 - Tipo de defecto a inyectar.
 - Momento en el que se debe producir el defecto.
 - Donde se debe inyectar el defecto.
 - Cuanto tiempo permanece activo el defecto.

```
<ack>
  <message>SystemUP</message>
</ack>
<process-id>
  <pathname>/home/Andres/testfik/asm_loop</pathname>
</process-id>
<injection>
  <fault class="binary">
    <type>Value</type>
    <register>ecx</register>
    <mask>0xffffffff</mask>
    <timeTrigger>now</timeTrigger>
  </fault>
</injection>
<injection>
  <fault>
    <type>FunctionInterception</type>
    <address>0x08048452</address>
    <timeTrigger>forever</timeTrigger>
    <parameter>
      <type>Value</type>
      <num>1</num>
      <mask>10</mask>
    </parameter>
  </fault>
</injection>
```

Situación actual

- Construcción de un prototipo capaz de inyectar fallas en hardware. y software con baja intrusividad.
- Utilización de la herramienta para medir la tolerancia a fallas en sistemas.

Conclusiones

- En general las aplicaciones no toleran las fallas originadas por defectos en registros del CPU.
- Con las inyecciones a memoria los resultados no son tan directos.
 - No todos los defectos generan errores.
 - Realizando pruebas de inyección aleatoria, luego de un tiempo de ejecución las fallas generadas no son toleadas por las aplicaciones.

Trabajo a Futuro

- Utilización de modelos estadísticos para simular los defectos.
- Integrar fallas a llamadas al sistema operativo.
- Portar a otras arquitecturas.

¿Preguntas?



Inyección de errores para evaluación de aspectos no funcionales en sistemas

Ariel Sabiguero Andrés Aguirre
{asabigue, aaguirre}@fing.edu.uy

Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, Uruguay

X Jornadas de Informática e Investigación Operativa

Parámetros de calidad

- Reliability(confiabilidad): $R(t)$
 - Probabilidad que un sistema cumpla sus requerimientos hasta un tiempo t cuando opera bajo sus condiciones establecidas de funcionamiento.
- Availability(disponibilidad): $A(t)$
 - Probabilidad que un sistema esté operando correctamente en el instante t de tiempo.

Parámetros de calidad

- Safety(seguridad): $S(t)$
 - Es la probabilidad que condiciones que pueden derivar en una situación adversa, no ocurran, independientemente de si el sistema cumple o no con su misión. Especialmente nos referimos a daños a propiedad, medio ambiente e incluso, vidas humanas.

Arquitectura

Interfaz Gráfica

- Gestiona y controla la ejecución de los experimentos.
- Visualización de resultados.
- Fue realizada por empresa española (MTP).

Arquitectura

Fault Injection Kernel

- Inyecta los defectos especificados por la interfaz gráfica.
- Componente de tamaño reducido.
- Muy baja intrusión sobre el sistema bajo prueba.
- Implementa monitorizaciones.
- Maneja asignación de tiempos a las inyecciones.