Herramientas de programación en C

José Galaviz

Análisis estático de código

- Es realizado sin ejecutar el código.
- Existen diferentes niveles de sofisticación.
  - Desde señalar algunos posibles errores en el código.
  - Hasta probar, usando métodos formales, algunas propiedades del código.
  - Desde analizar unidades pequeñas de código.
  - Hasta analizar interacciones complejas entre diferentes componentes.

cppcheck

- Herramienta para hacer análisis estático de código en diferentes dialectos de C y C++.
- Se pretende encontrar errores:
  - Verificación de cotas.
  - Verificación de fugas de memoria dinámicamente solicitada.
  - Referencias de memoria nulas.
  - Variables no inicializadas antes de usarlas.
  - Verificar la seguridad en el uso de funciones.
  - Y muchos más: `http://sourceforge.net/p/cppcheck/wiki/ListOfChecks/`

Ejemplo 1

```
cppcheck -v --enable=all -I include . 2> Err.txt

[matrices.c:89]: (error) Memory leak: a
[matrices.c:89]: (error) Memory leak: b
[matrices.c:89]: (error) Memory leak: c
```

Ejemplo 2

```
void f()
{
    char *p;
    *p = 0;
}
```

```
[test.c:3]: (style) Variable 'p' is not assigned a value.
[test.c:4]: (error) Uninitialized variable: p
```

Ejemplo 3

```
void f1(struct fred_t *p) {
    // dereference p and then check if it's NULL
    int x = p->x;
    if (p)
        do_something(x);
}
void f2() {
    const char *p = NULL;
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == ' ') {
            p = str + i;
            break;
        }
    }
    // p is NULL if str doesn't have a space. If str always has a
    // a space then the condition (str[i] != '\0') would be redundant
    return p[1];
}

void f3(int a) {
    struct fred_t *p = NULL;
    if (a == 1)
        p = fred1;

    // if a is not 1 then p is NULL
    p->x = 0;
}
```

Ejemplo 3

```
[test.c:4] -> [test.c:5]: (warning) Possible null pointer
dereference: p - otherwise it is redundant to check it
against null.
[test.c:23]: (error) Possible null pointer dereference: p
[test.c:33]: (error) Possible null pointer dereference: p
```

En Java

- Checkstyle (`http://checkstyle.sourceforge.net/`)
- FindBugs (`http://findbugs.sourceforge.net/`)
- PMD (`http://pmd.github.io/`)

Análisis dinámico

- Se lleva a cabo cuando el programa está en ejecución.
- Se pueden hacer varias cosas:
  - Detección de errores en uso de memoria.
  - Detección de errores de sincronización de threads.
  - Elaborar perfiles de predicción de salto.
  - Perfiles de llamadas a subrutinas y estado del stack.
  - Etc
- No es trivial porque se debe envolver el programa en un ambiente controlado de ejecución

Valgrind

- Es un framework para construir herramientas de análisis dinámico.
- La distribución de valgrind ya incluye seis de ellas.
- Ahorra tiempo de depuración.
- Contribuye a lograr programas eficientes.

Ejemplo 1

```
#include <stdlib.h>

  void f(void)
  {
     int* x = malloc(10 * sizeof(int));
     x[10] = 0;  // problem 1: heap block overrun
  }              // problem 2: memory leak -- x not freed

  int main(void)
  {
     f();
     return 0;
  }
```

Ejemplo 1

```
==19182== Invalid write of size 4
  ==19182==    at 0x804838F: f (example.c:6)
  ==19182==    by 0x80483AB: main (example.c:11)
  ==19182==  Address 0x1BA45050 is 0 bytes after a block of
size 40 alloc'd
  ==19182==    at 0x1B8FF5CD: malloc (vg_replace_malloc.c:
130)
  ==19182==    by 0x8048385: f (example.c:5)
  ==19182==    by 0x80483AB: main (example.c:11)
==19182== 40 bytes in 1 blocks are definitely lost in loss
record 1 of 1
  ==19182==    at 0x1B8FF5CD: malloc (vg_replace_malloc.c:
130)
  ==19182==    by 0x8048385: f (a.c:5)
  ==19182==    by 0x80483AB: main (a.c:11)
```

Características

- Uso de variables no inicializadas.
- Lecturas/escrituras de memoria previamente liberada.
- Lecturas/escrituras fuera de cota.
- Superposiciones de memoria.
- Y otros

Análisis de flujo de control

- Sirve para determinar todas las posible rutas que el flujo de control puede seguir en la ejecución del programa.
- Lo que es potencialmente útil para diseñar pruebas.

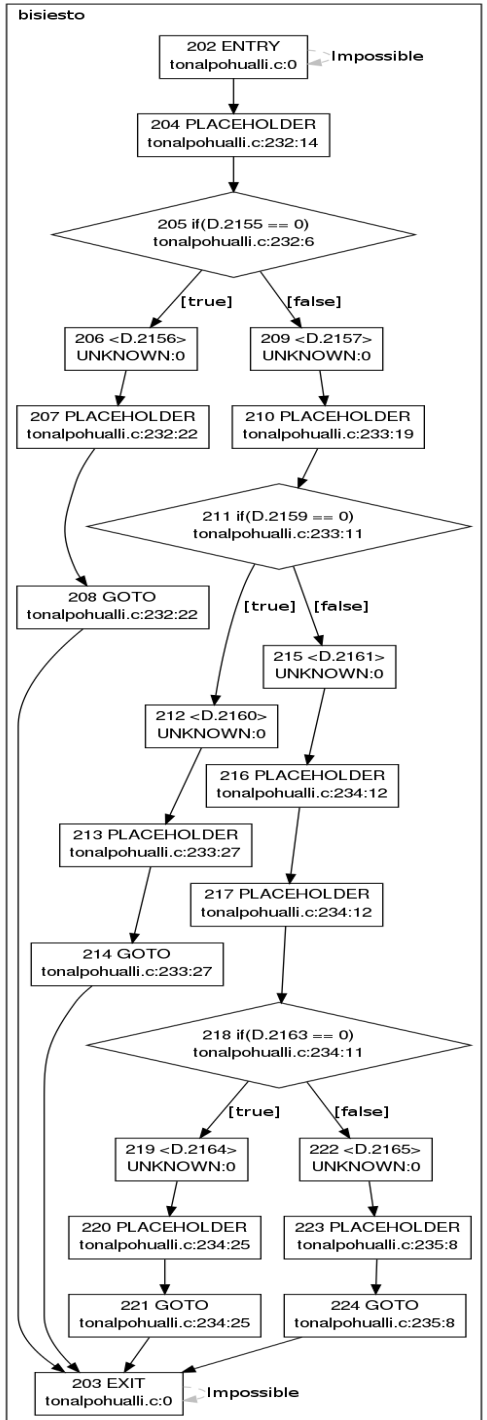
CoFlo

- Herramienta de software libre para analizar el flujo de control.
- Genera una salida que puede transformarse, literalmente, en un gráfica que muestra las rutas del flujo de control.
- `http://coflo.sourceforge.net/wordpress/`

Ejemplo

```
coflo tonalpohualli.c -O Analysis-html


int
bisiesto(int nano) {
   if (!(nano % 400))
      return 1;
   else if (!(nano % 100))
      return 0;
   else if (!(nano % 4))
      return 1;
   else
      return 0;
}
```


bisiesto
202 ENTRY
tonalpohualli.c:0
Impossible
204 PLACEHOLDER
tonalpohualli.c:232:14
205 if(D.2155 == 0)
tonalpohualli.c:232:6
[true]
[false]
206 <D.2156>
UNKNOWN:0
209 <D.2157>
UNKNOWN:0
207 PLACEHOLDER
tonalpohualli.c:232:22
210 PLACEHOLDER
tonalpohualli.c:233:19
211 if(D.2159 == 0)
tonalpohualli.c:233:11
208 GOTO
tonalpohualli.c:232:22
[true]
[false]
215 <D.2161>
UNKNOWN:0
212 <D.2160>
UNKNOWN:0
216 PLACEHOLDER
tonalpohualli.c:234:12
213 PLACEHOLDER
tonalpohualli.c:233:27
217 PLACEHOLDER
tonalpohualli.c:234:12
214 GOTO
tonalpohualli.c:233:27
218 if(D.2163 == 0)
tonalpohualli.c:234:11
[true]
[false]
219 <D.2164>
UNKNOWN:0
222 <D.2165>
UNKNOWN:0
220 PLACEHOLDER
tonalpohualli.c:234:25
223 PLACEHOLDER
tonalpohualli.c:235:8
221 GOTO
tonalpohualli.c:234:25
224 GOTO
tonalpohualli.c:235:8
203 EXIT
tonalpohualli.c:0
Impossible

En Java

- Cubertura: (`http://cobertura.github.io/cobertura/`)
  - Muestra el porcentaje de código cubierto por las pruebas.
- SonarQube: (`http://www.sonarqube.org/`)
- Plugin de métricas de Eclipse.

Doxygen

- Procesador de documentación al estilo de javadoc para C y C++ (de hecho para cualquier lenguaje).
- Al igual que javadoc usa etiquetas para indicar la semántica del comentario.

Ejemplo

```
/**
 * \brief Writes an image to a JPG file.
 * Given a memory buffer, which contains the pixel...
 * ...writes a file in JPG format
 * @param[in] img is the image row array.
 * @param[in] fname is the name of the file where the
 * image will be writen.
 * @param[in] width is the image width.
 * @param[in] height is the image height
 * \return an integer which represents the result.
 * \pre The image buffer must be non NULL
 * pixel encoded in ARGB format.
 * \post new file, whose name is provided, in PNG format.
 */
int writeJPGImage(unsigned int **img, char *fname,
                  int width, int height);
```

`doxygen -g`

(genera un archivo de configuración)

`doxygen Doxyfile`

```
int writeJPGImage ( unsigned int **  img,
                    char *           fname,
                    int              width,
                    int              height
                  )
```

Writes an image to a JPG file.

Given a memory buffer, which contains the pixel information of some image in the format ARGB (packed in unsigned int's), this functions writes a file in JPG format with such image, excluding the alpha channel.

Parameters

- `[in]` **img** is the image row array.
- `[in]` **fname** is the name of the file where tha image will be writen.
- `[in]` **width** is the image width.
- `[in]` **height** is the image height

Returns

an integer which represents the result of the operation: 1 if success, a negative number otherwise.

Precondition

The image buffer must be non NULL and must contain the each pixel encoded in ARGB format.

Postcondition

new file, whose name is provided, in PNG format and an integer with a result code. 1 if the operation was sucessful or a negative number otherwise.