

Programación I

Depuración de código

Pablo Garaizar Sagarminaga
Borja Sanz Urquijo

Facultad de Ingeniería

Índice

- Introducción.
- Depuración troglodita.
- Depuración con fundamento.

9/9

Introducción

0800

Antan started

1000

" stopped - antan ✓

1300 (033) MP - MC

~~1.982647000~~
~~2.130476415~~

9.037 847 025

9.037 846 995 correct

4.615925059(-2)

(033) PRO 2

2.130476415

correct

2.130676415

Relays 6-2 in 033 failed special speed test
in relay " 11.000 test.

Relays changed

1100

Started Cosine Tape (Sine check)

1525

Started Multi-Adder Test.

1545



Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

~~1630~~ 1630

Antan started.

1700

closed down.

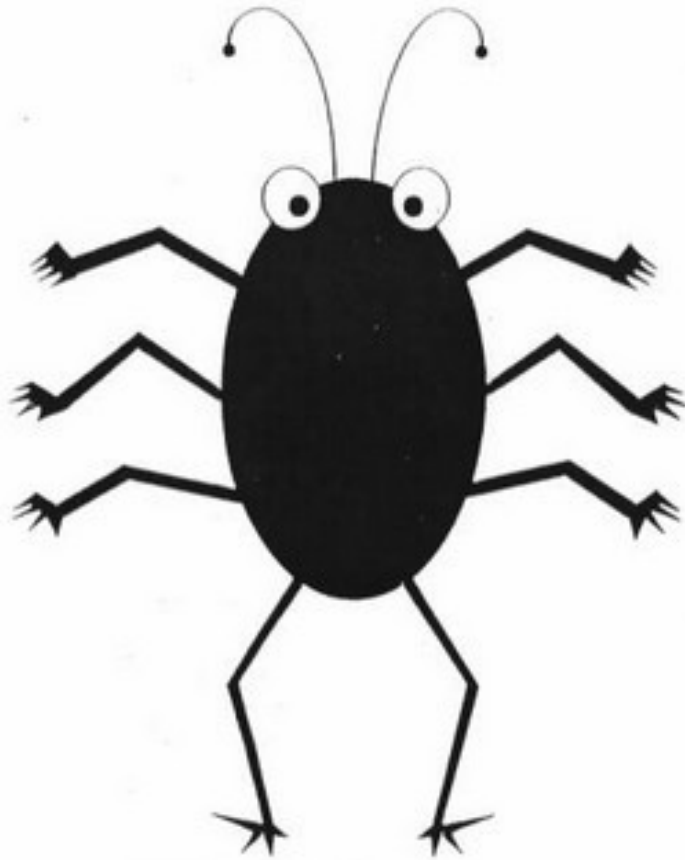
Relay
2145
Relay 3370

Historia

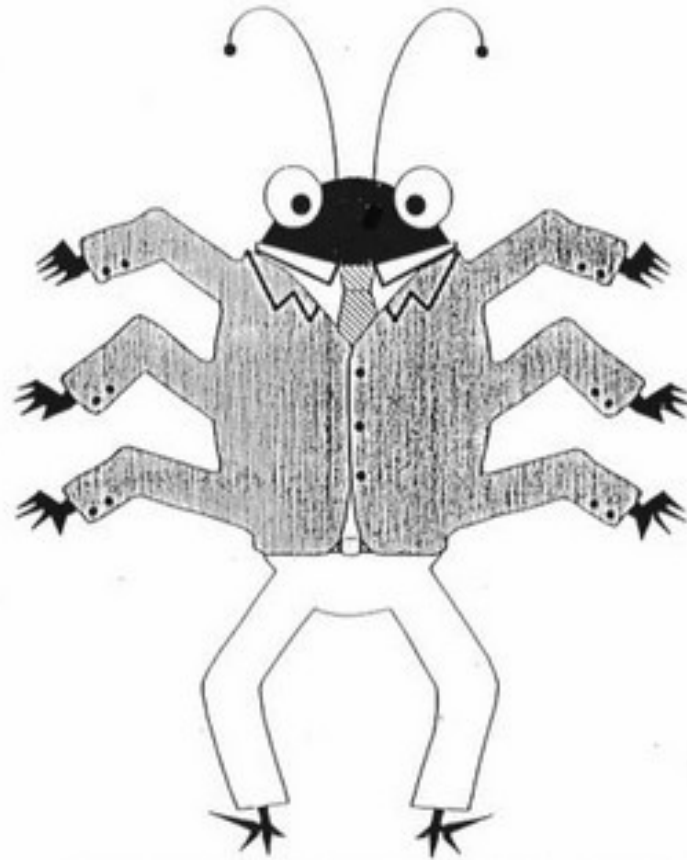
- **Bug** (“bicho”) se usa como sinónimo de error desde el s. XIX.
 - La primera en usarlo como error en informática fue Grace Hopper en los años 1940.
 - Desde esa época había métodos para encontrar y corregir errores de programación.
 - Los programas **interactivos** para depurar errores de otros programas vienen usándose desde 1970 (ej: ODT: On-line Debugging Tool).

Las 6 fases de la depuración

- 1) Eso **no puede** ocurrir.
- 2) Eso **en mi ordenador** no ocurre.
- 3) Eso **no debería** ocurrir.
- 4) ¿**Por qué** ocurre eso?
- 5) Oh, **ya veo...**
- 6) ¿**Cómo** ha podido funcionar eso **antes**?



BUG



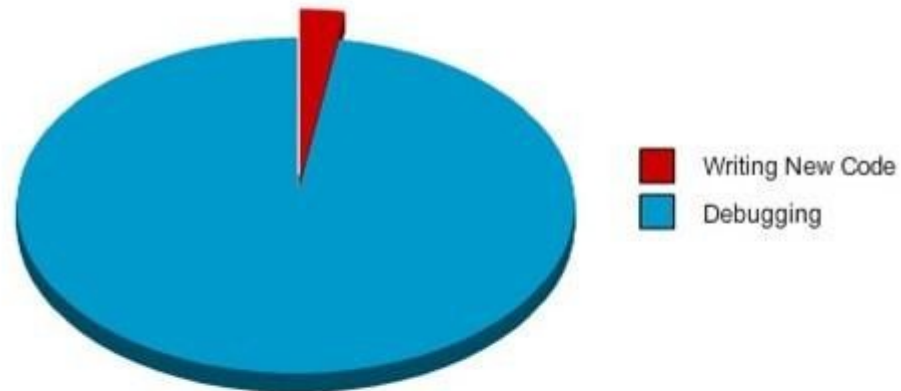
FEATURE



Depuración

- Es importante porque nos pasamos más tiempo **arreglando fallos** que programando **nuevas funcionalidades**:

Programming



Depuración troglodita



Depuración troglodita

- A pesar de que no es **nada recomendable**, es mejor que no depurar.
- ¿Cómo?
 - Llenando el código de **mensajes informativos**:

Ejemplo:

```
for i in range(10):  
    print("Voy por la vuelta" + i) # Debug  
    hacerCosas(i)
```

Depuración troglodita

- Otro truco clásico es poner una **pausa** en el programa antes del momento crítico.
- ¿Cómo?
 - En Python podemos usar **`input()`**.

Depuración troglodita avanzada

- Podemos mejorar un poco las cosas si usamos un **booleano** para **decidir si mostrar** o no los mensajes de depuración:

```
debug = True
```

```
for i in range(10):  
    if (debug):  
        print("Voy por la vuelta" + i) # debug  
    hacerCosas(i)
```

- Así, si queremos **deshabilitarlos**, basta con poner debug a False.

A photograph of a long, perspective view of a classical colonnade. The columns are made of light-colored stone and feature ornate capitals. The perspective leads the eye from the foreground columns towards the background, where the columns become smaller and eventually blend into the distance. The lighting is warm, suggesting late afternoon or early morning, with long shadows cast across the ground.

Depuración con fundamento

Depuración con fundamento

- Casi todos los entornos de desarrollo cuentan con un **depurador integrado**.
- En Python, la librería estándar incorpora el pdb (Python debugger).



JORGE CHAM © 2005

www.phdcomics.com

Depuración

- Conceptos importantes:
 - **Breakpoint**: punto de ruptura de la ejecución.
 - Pondremos un breakpoint antes de la parte que creemos problemática.
 - Una vez detenido el programa, podremos ir avanzando paso a paso.
 - **Post-mortem**: análisis del programa una vez ya ha fallado.

Depuración

- ***Breakpoint***: `pdb.set_trace()`

```
import pdb
def div(a, b):
    """Return a / b."""
    pdb.set_trace()
    return a / b
>>> print(div(5, 0))
-> return a / b
(Pdb)
```


Depuración

- Una vez pdb ha parado, podemos hacer muchas cosas:

`(Pdb) p a # imprime el valor actual de a`

`(Pdb) !b = 1 # cambia el valor de b`

`(Pdb) next # avanza un paso la ejecución`

`(Pdb) step # como next pero entrando en funciones`

`(Pdb) until 12 # continúa la ejecución hasta la línea 12`

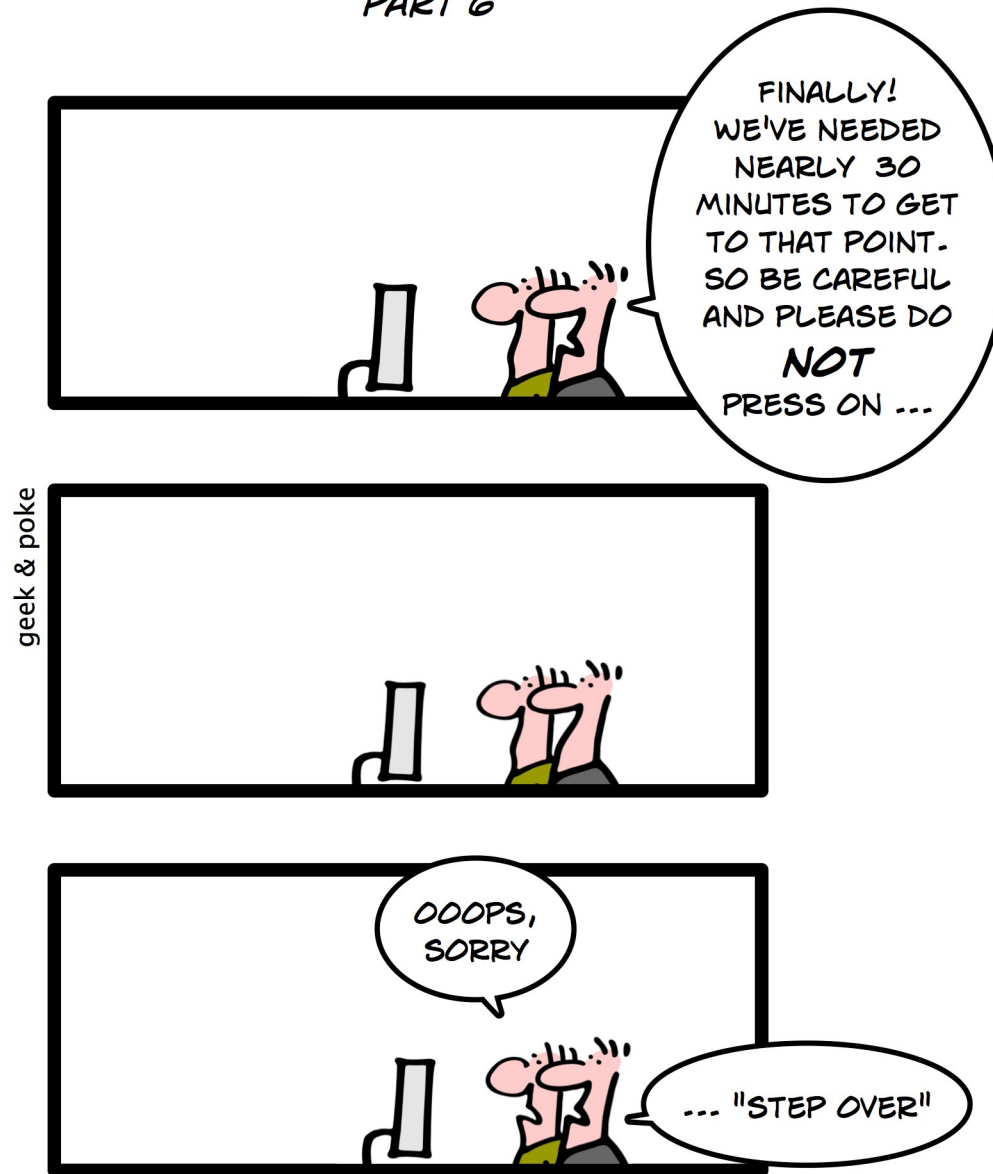
`(Pdb) continue # continúa la ejecución`

`(Pdb) h # ayuda`

`(Pdb) interact # consola de Python con todas las variables`

ONE DAY IN THE LIFE OF A CODER

PART 6



0230 PM: DEBUGGING

Depuración

- ***Post-mortem***: lanza tu script con:
`python -m pdb script.py`
- Una vez se produzca el error, pdb habrá parado y podremos depurar el código.

Referencias

- Cómo Pensar como un Informático. El aprender con Python 2º Edición por Jeffrey Elkner, Allen B. Downey, y Chris Meyers (trad. Gregorio Inda)
- recursospython.com
- Wikipedia.

Referencias

- Imágenes:
 - Wikipedia
 - [Geek & poke.](#)
 - [PhDcomics.com](#)
 - <https://www.flickr.com/photos/thaqela/6774236608>
 - <https://www.flickr.com/photos/jechstra/21828418272>