

Alice nel pozzo: c'è davvero caos dentro il PC?

Claudio Ferretti

**Università degli Studi di Milano - Bicocca
Dip. di Informatica, Sistemistica e Comunicazione
(DISCo)**

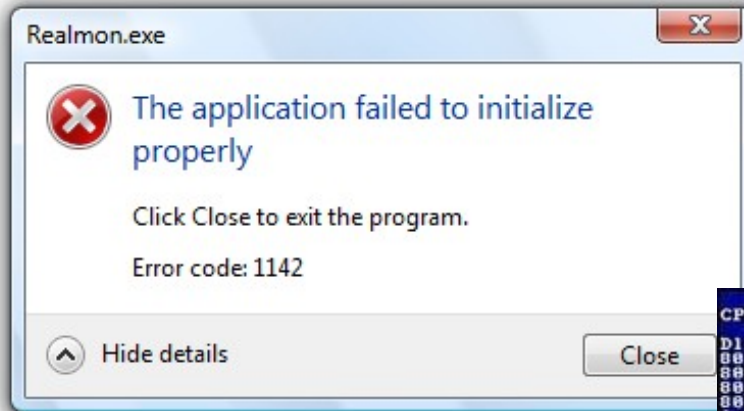
Viale Sarca 336/14 - Milano - Italy

- Cosa si vuole presentare:
 - Il computer a volte sembra un caos che sopravvive a se stesso
 - Guardiamo dietro lo schermo...
 - ...in effetti è complicato, ma
 - Tecnologie che lo tengono 'in ordine'
 - Quali tecnologie possono aiutarci a leggere il suo 'caos' come 'ordine'
 - Cosa potrebbe migliorare nel futuro

Incontro che sorprende!



Incontro che sorprende!



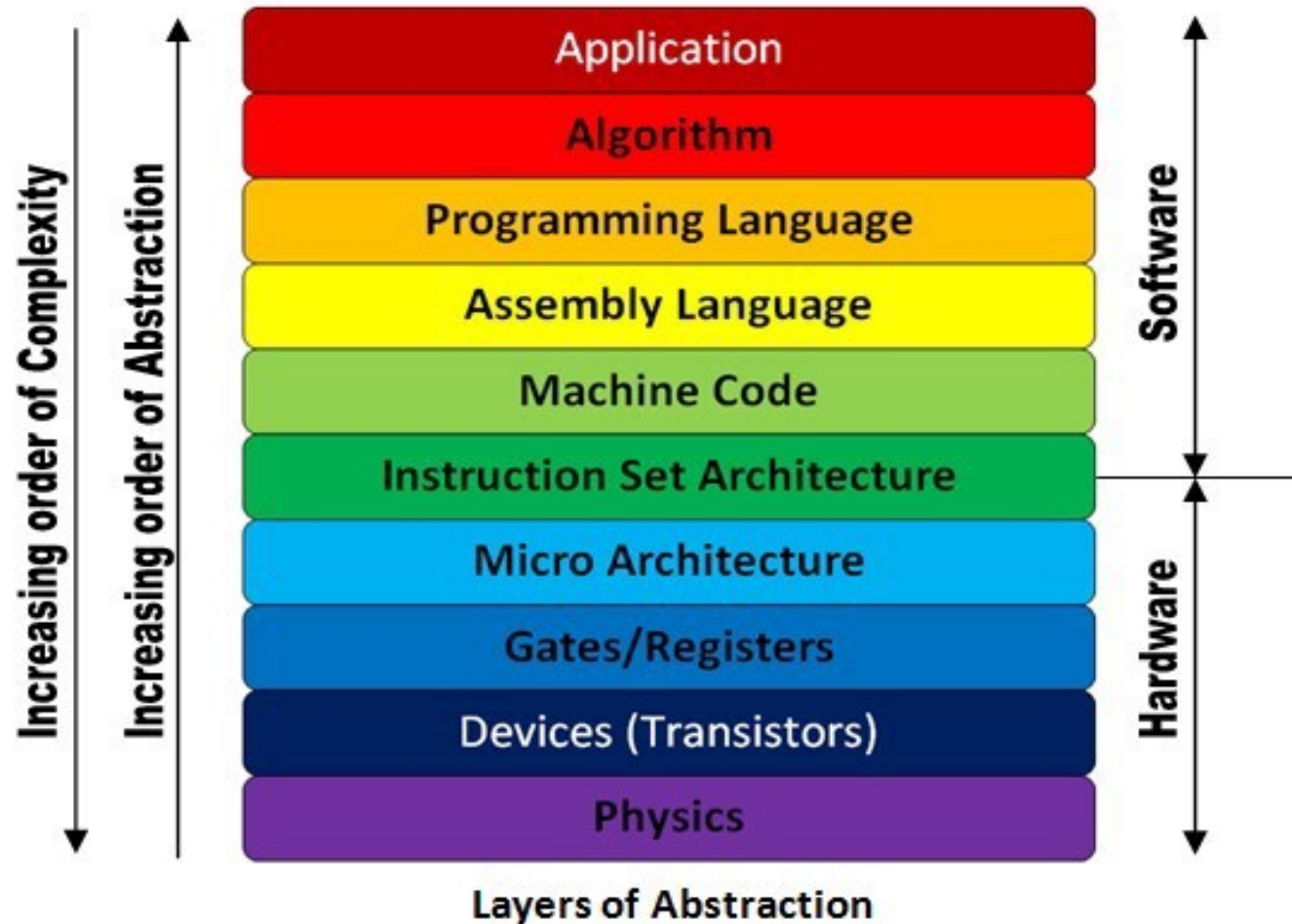
Incontro che sorprende!



OK, vediamo da vicino ...

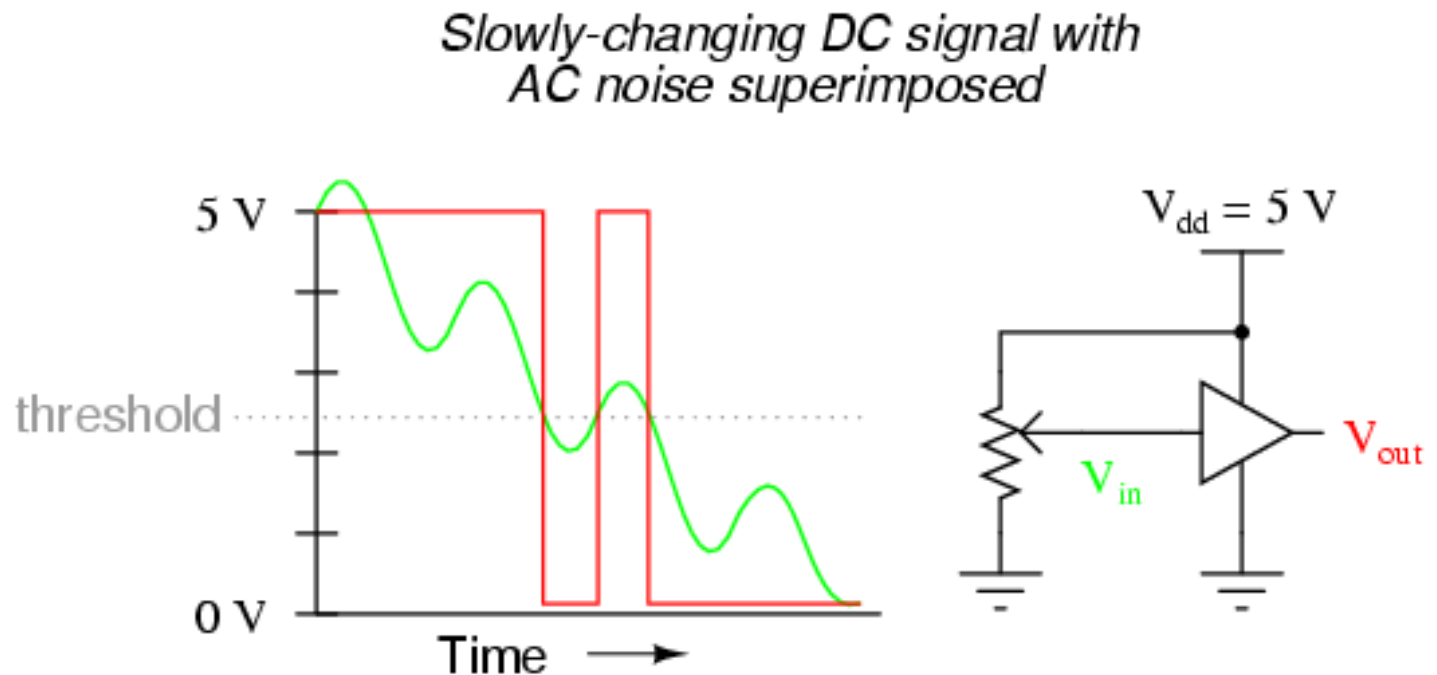


Il pozzo è profondo:



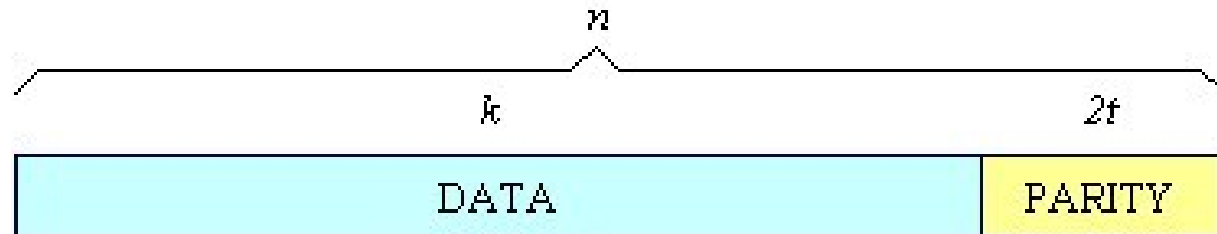
Caos/rumore, in hardware?

- L'hardware, elettronico, è basato su fenomeni fisici analogici e imprecisi
- Ma l'elettronica è costruita per nascondere il rumore:



Vogliamo 'sprecare' memoria?

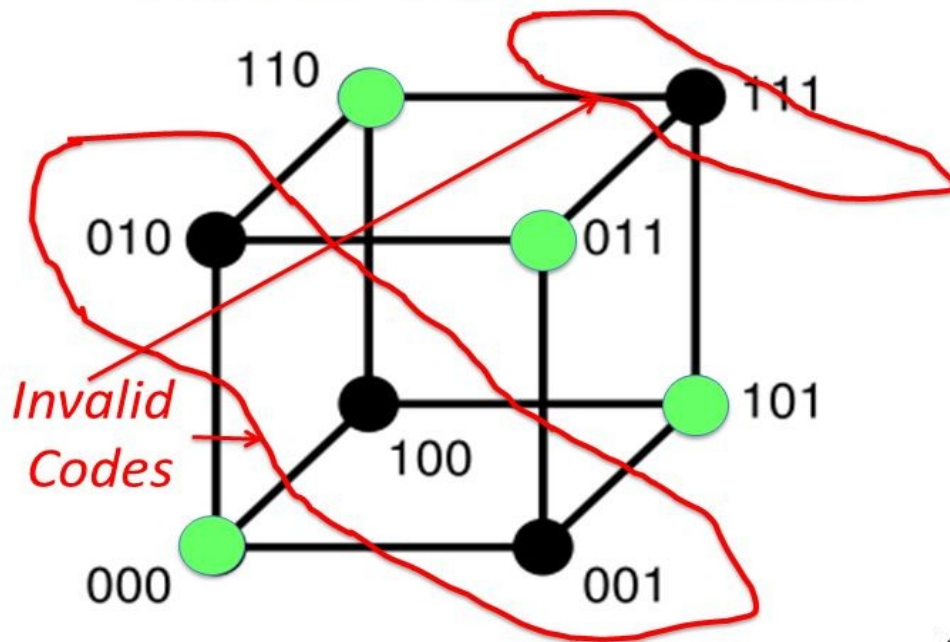
- A volte l'incertezza sulla prestazione dell'hardware mette in pericolo i dati:
- la memoria (RAM) del computer può fare errori,
- ma se ne usiamo più di quanta serve per i dati possiamo proteggerci:



Proteggiamo i numeri dal rumore!

Hamming Distance 2: Detection

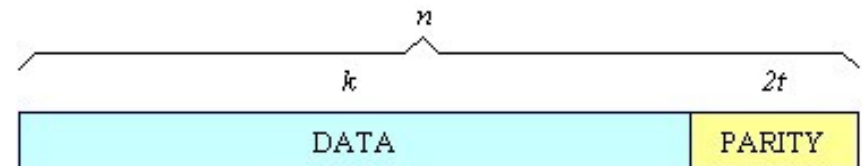
Detect Single Bit Errors



- No 1 bit error goes to another valid code
- $\frac{1}{2}$ codes are valid

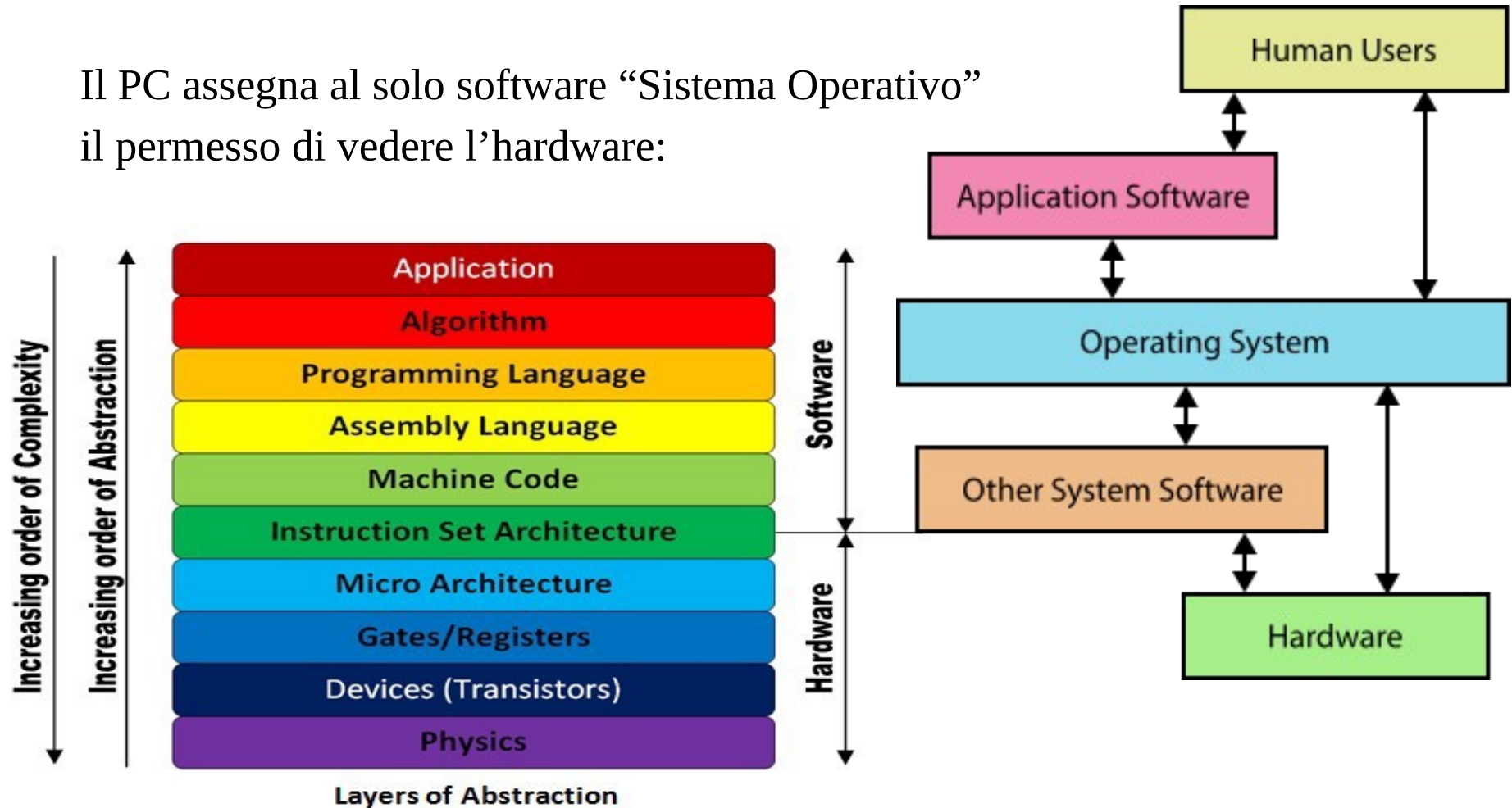
Esempio: memorizziamo 2 +1 bit
 La memoria “in più” ci aiuta a
 correggere errori (che emergono da
 problemi nell’hardware):

I bit a 1 devono essere in numero pari



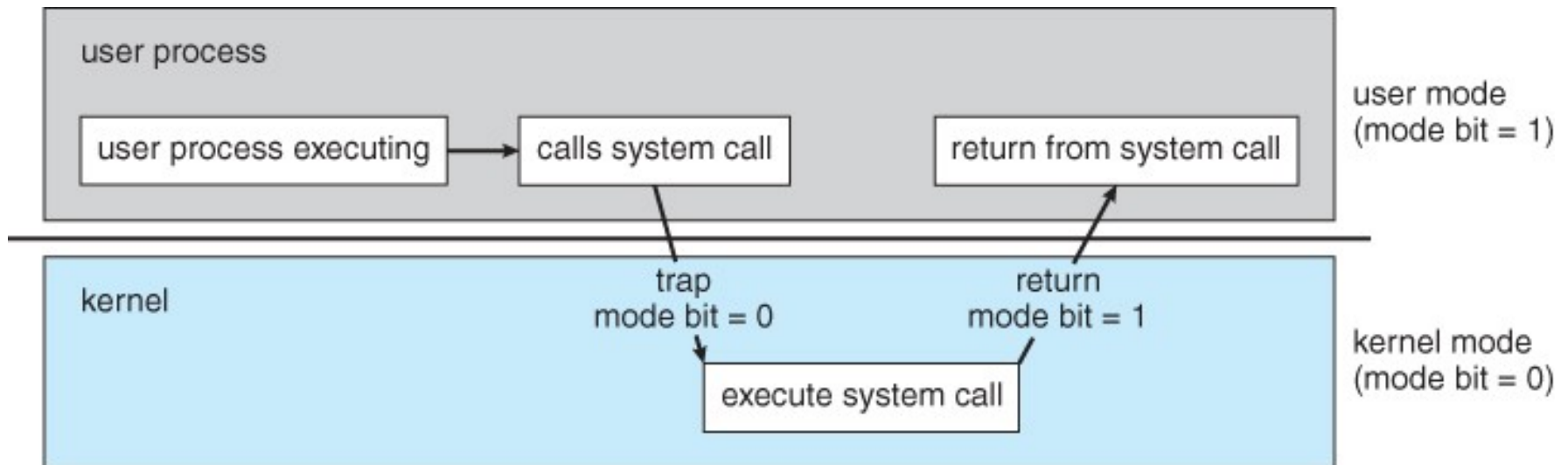
“Nascondiamo” l’hardware

Il PC assegna al solo software “Sistema Operativo” il permesso di vedere l’hardware:



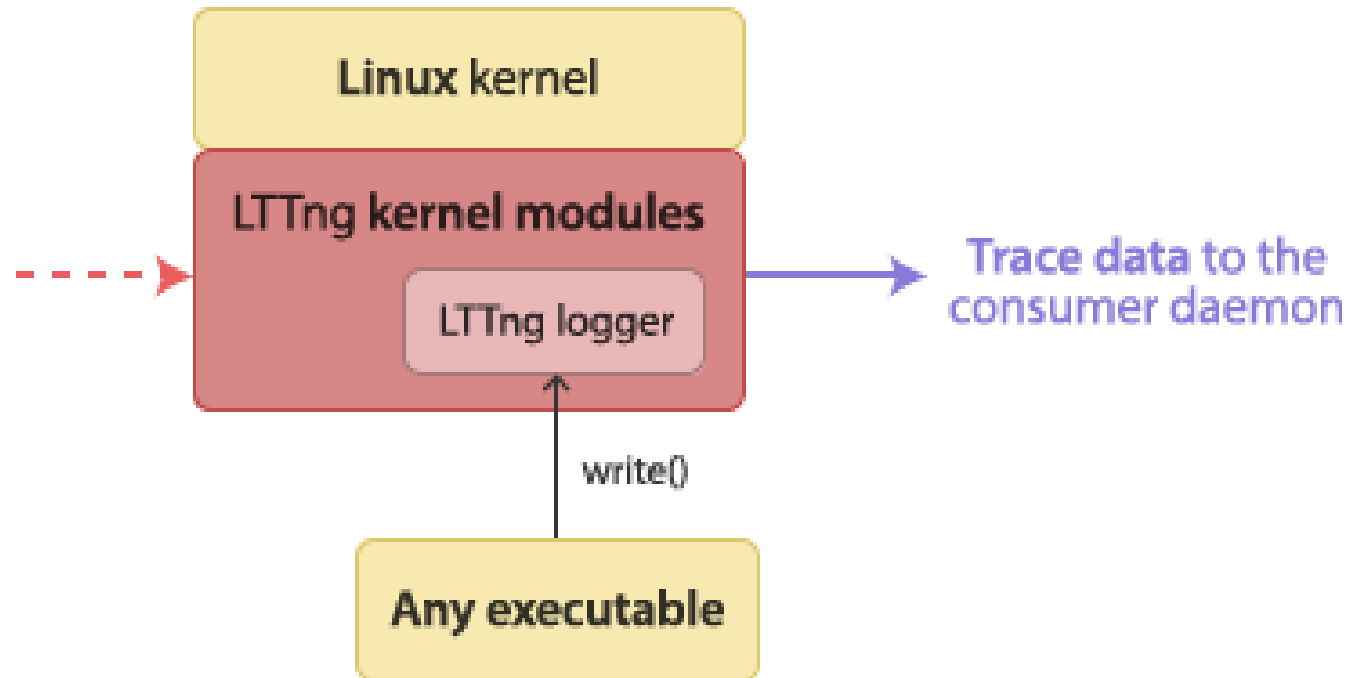
Separiamo i due mondi software

L'utente (software in 'user mode') chiede servizi al [sistema operativo/kernel](#):



Ma se vogliamo guardare dentro...

Usiamo tecnologie di [tracciatura](#), esempio in Linux:



Ma se vogliamo guardare dentro...

Ecco quali servizi chiede un programma test in Windows:



Vogliamo invece capire le applicazioni

Diamo senso al codice sorgente: ricerche per la “Program Comprehension”

```
public static void main(String [] args) {  
    int [] data = {1, 2, 4, 5, 6, 10};  
    computeMedian(data);  
}  
  
public int computeMedian(int[] data) {  
    int[] data = data;  
    sort(data);  
    float b;  
  
    if (data.length % 2 == 1)  
        b = data[data.length / 2];  
    else  
        b = ((float) data[data.length / 2 - 1] +  
             (float) data[data.length / 2]) / 2;  
  
    return (b);  
}
```

That's the list of numbers...
b should be 4.5

It should compute the
median

Case for the odd array

Case for the even array

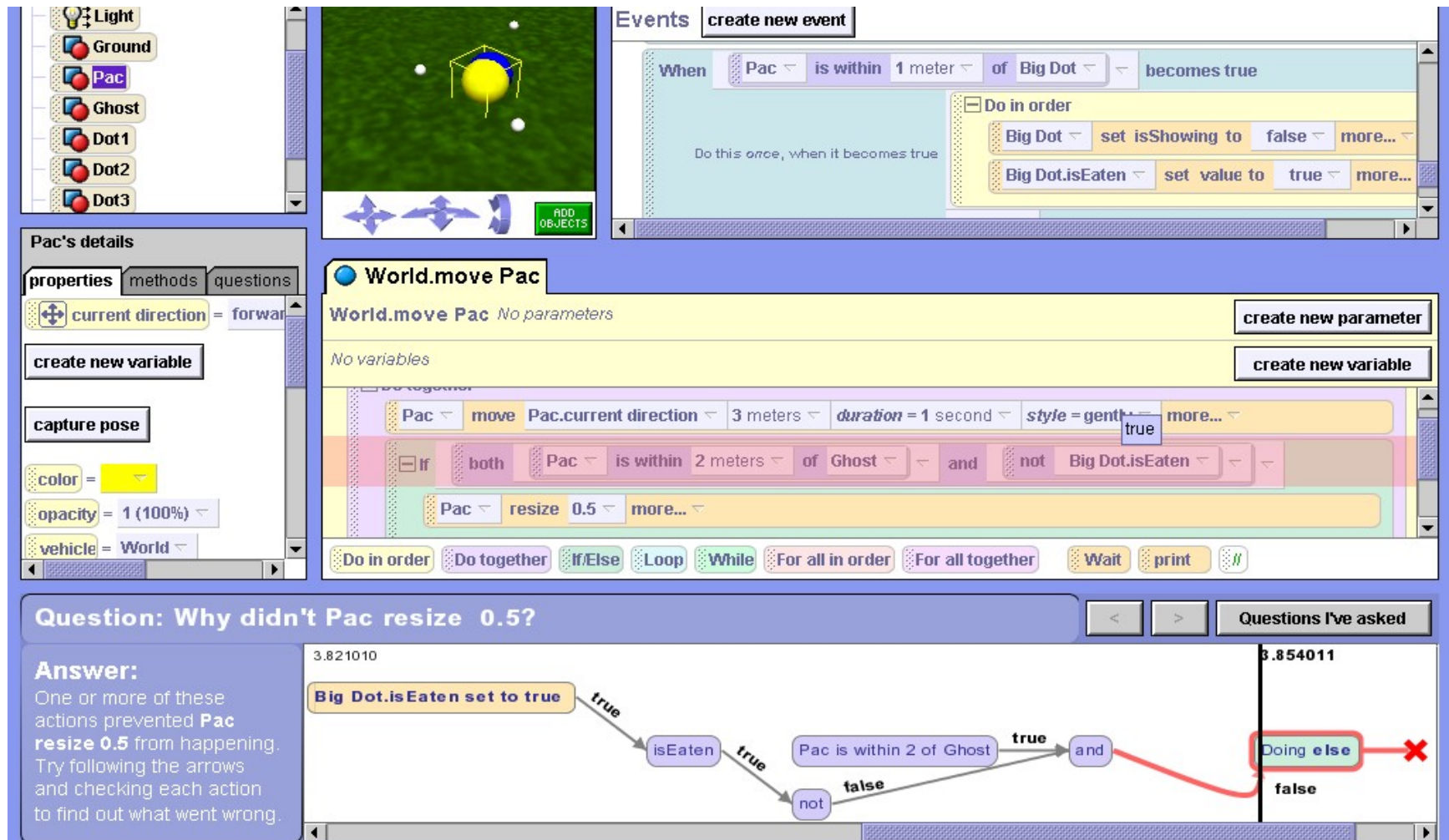
Vogliamo invece capire le applicazioni

Ad esempio: strumenti per fare [Slicing](#)

mid() {		
int x,y,z,m;		
1: read("Enter 3 numbers:",x,y,z);		• 3,3,5
2: m = z;		•
3: if (y<z)		•
4: if (x<y)		•
5: m = y;		
6: else if (x<z)		•
7: m = y; // bug		•
8: else		
9: if (x>y)		
10: m = y;		
11: else if (x>z)		
12: m = x;		
13: print("Middle number is:", m);		•
}	Pass/Fail	P

Vogliamo invece capire le applicazioni

Altro esempio, [Whyline](#) chiede al programma perchè ha fatto un dato calcolo:



The screenshot displays the Whyline interface for a Pac-Man simulation. The top left shows a list of objects: Light, Ground, Pac, Ghost, Dot1, Dot2, and Dot3. The top center is a 3D visualization of the game world with a yellow Pac-Man character and a large yellow dot. The top right shows the 'Events' panel with a trigger 'When Pac is within 1 meter of Big Dot becomes true' and actions 'Big Dot set isShowing to false' and 'Big Dot.isEaten set value to true'. The bottom left shows 'Pac's details' including 'current direction = forward', 'create new variable', 'capture pose', 'color', 'opacity = 1 (100%)', and 'vehicle = World'. The bottom center shows the 'World.move Pac' block with parameters 'Pac', 'Pac.current direction', '3 meters', 'duration = 1 second', and 'style = gentle'. The bottom right shows a 'Question: Why didn't Pac resize 0.5?' with an 'Answer' section. The answer states: 'One or more of these actions prevented Pac resize 0.5 from happening. Try following the arrows and checking each action to find out what went wrong.' Below the answer is a flowchart showing the execution path: 'Big Dot.isEaten set to true' (true) leads to 'isEaten' (true), which leads to 'Pac is within 2 of Ghost' (true), which leads to 'and' (true), which leads to 'Doing else' (false) with a red 'X' mark. The flowchart also shows a 'not' path from 'isEaten' (true) to 'Pac is within 2 of Ghost' (false), which leads to 'and' (false).

Question: Why didn't Pac resize 0.5?

Answer:
One or more of these actions prevented **Pac** **resize 0.5** from happening. Try following the arrows and checking each action to find out what went wrong.

3.821010

Big Dot.isEaten set to true

isEaten true

Pac is within 2 of Ghost true

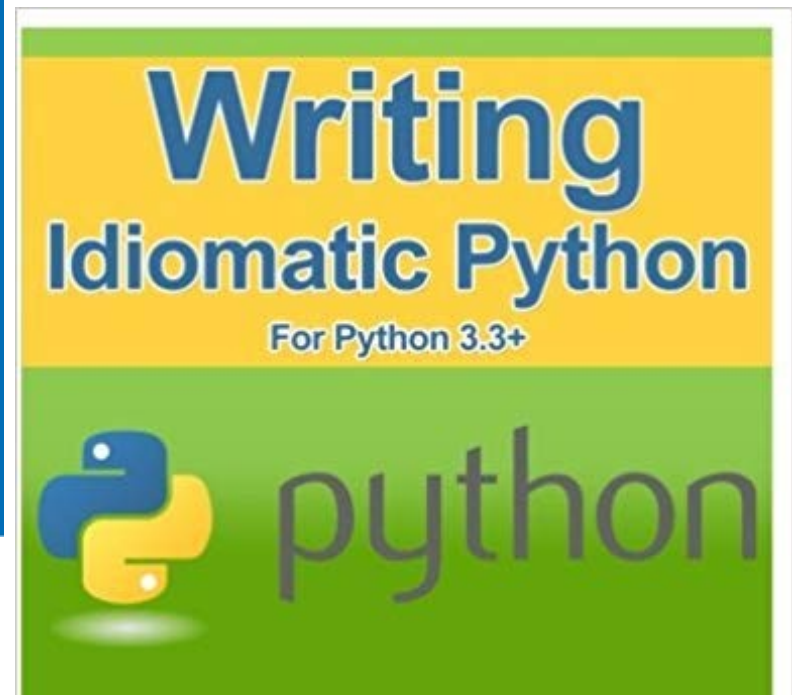
and

Doing else false

3.854011

Come un programmatore legge/capisce il codice?

Idiomi e schemi usuali!



Dobbiamo addestrare la mente:

Il codice idiomatico è familiare solo a chi l'ha usato a lungo:

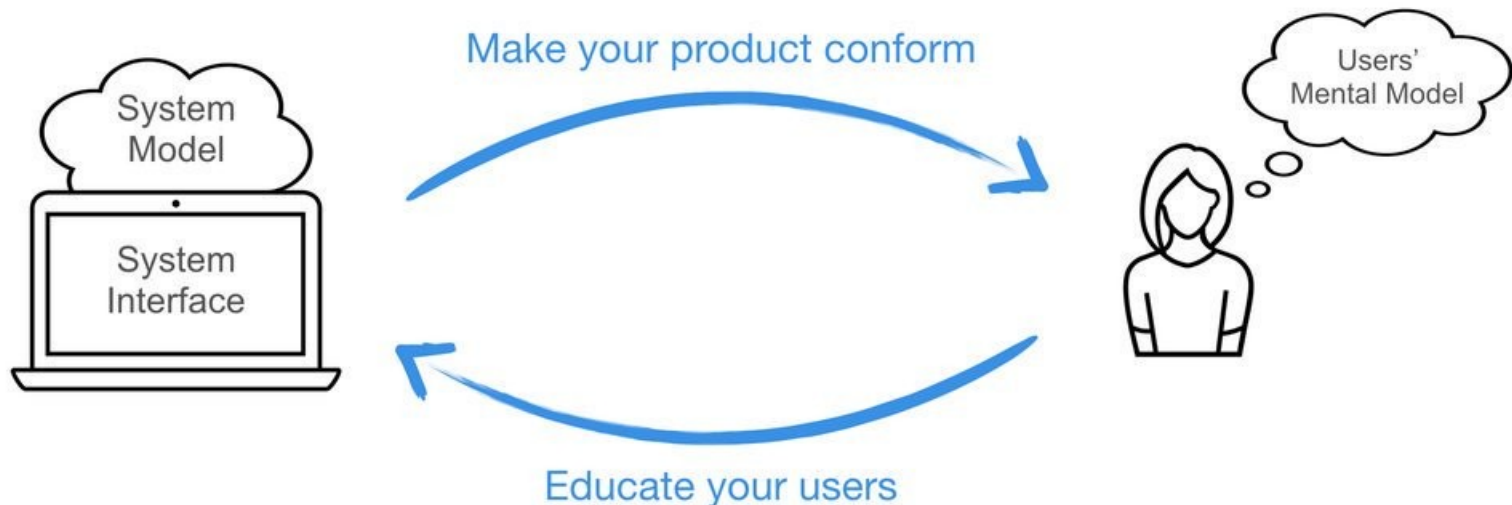
```
res = []  
for i in [1,2,3,4,5]:  
    if i < 4:  
        res.append(i)  
    else:  
        res.append(0)
```



```
res = [i if i < 4 else 0 for i in range(1,6)]
```


Andiamo verso il futuro...usiamo ciò che abbiamo già nella mente?

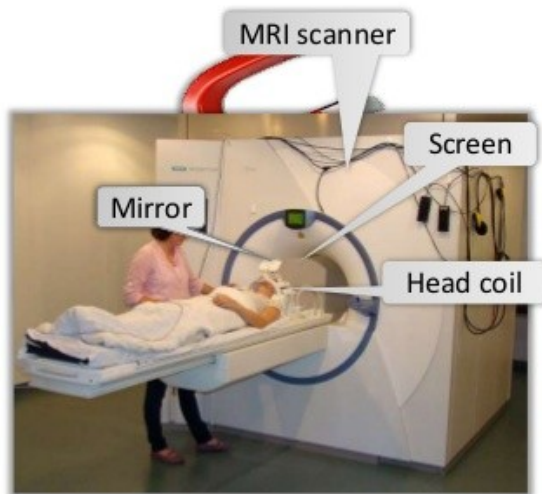
La macchina ci deve presentare i suoi dettagli in una forma
che ci sia familiare:



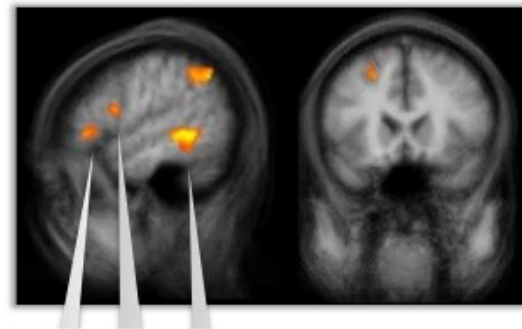
Ma...cosa usano i programmatori?

- Parliamo proprio di mente in termini di neuroni:

Functional Magnetic Resonance Imaging

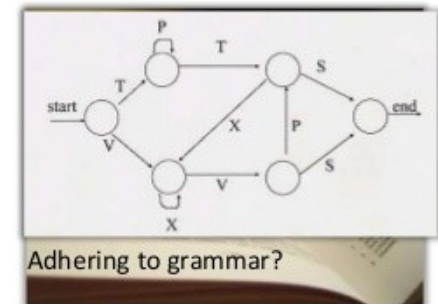


Results



Language processing:

- Natural and artificial
- Semantic processing at word level
- Combinatorial aspects



Adhering to grammar?

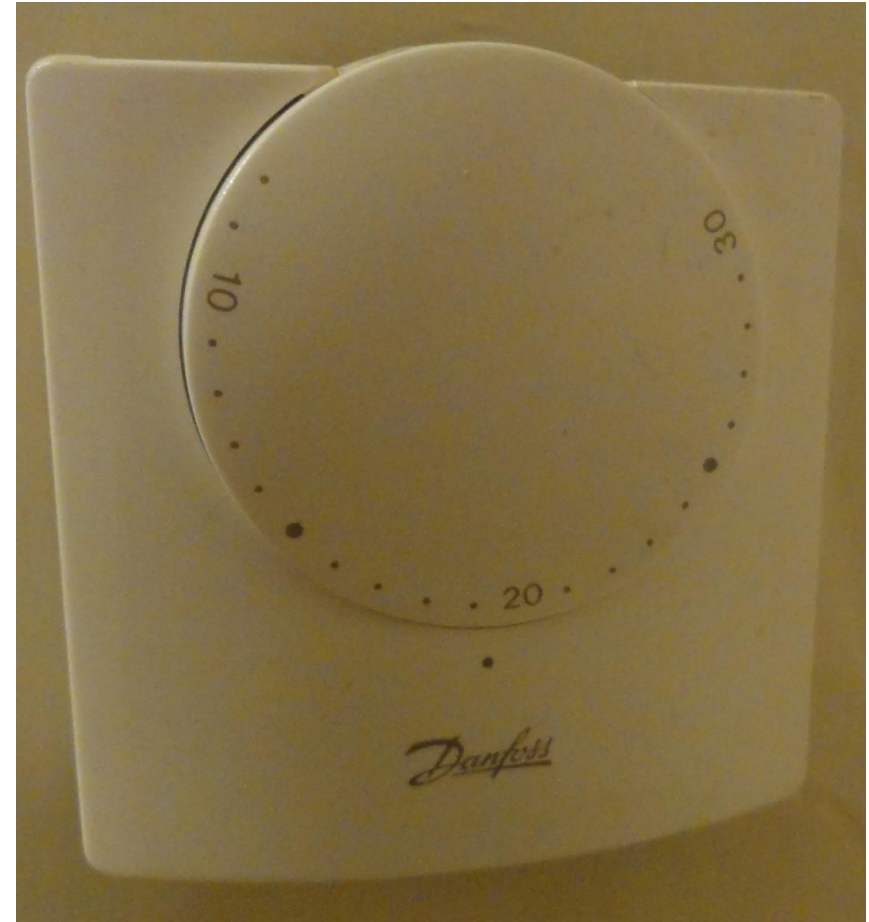
VXVS
TPTXVS
VXXXS
TPTPS

E noi che modelli mentali possiamo usare al computer?

Se la stanza è a 15° e fa freddo:

Lasciamo il termostato su 20° ?

O lo spostiamo su 22° per far prima ?



Potenza dei modelli mentali: li usiamo anche in automatico

I ricercatori arrivano a definirli
“zombie” nella nostra mente
(se volete sono una parte sonnambula di noi)

Ma ci aiutano a svolgere bene molti compiti...

(Lady Macbeth
sonnambula)



- Allo scopo di rendere più comprensibile il “caos” di ciò che accade nel PC:
- Vogliamo che il PC lasci vedere di più cosa fa, ma...
- Deve farlo in modo vicino al nostro pensare!

La ricerca lavora in nuove direzioni:

- Identificare nostri modelli mentali che assomiglino a fenomeni del PC
- *Tradurre in automatico dettagli interni al PC in rappresentazioni familiari
- Esplicitare i modelli mentali appresi dagli esperti
- e trovare il modo di addestrare su di essi con minimo sforzo in non esperti

*Tradurre in automatico dettagli interni in rappresentazioni note

Esempi (prodotti da reti neurali appositamente addestrate):

```
hi--;  
while (lo < hi) {  
    Object t = a[lo];  
    a[lo++] = a[hi];  
    a[hi--] = t;  
}
```

Suggestions: ► reverse, range (22.2%)

JsonWriter pop()

```
if (named) throw  
    new IllegalStateException  
stack.pop().close();  
current = stack.size == 0 ?  
    null : stack.peek()  
return this;
```

Suggestions: ► close (21.4%) ►

Usciamo dal pozzo! ...

Incubo?



...senza caos! ;-)



*Grazie per
l'attenzione !*