

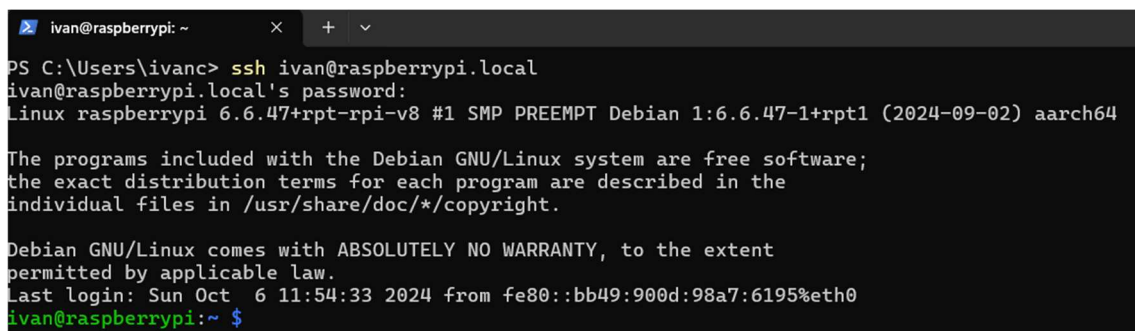
## Puzle 1 – “RFID Reader”

### Configuració de la Raspberry

Amb l'objectiu de controlar la RPi mitjançant el protocol SSH, s'ha de configurar la imatge del SO de tal manera que el servei SSH quedi habilitat. Això es fa entrant a la configuració de la imatge just abans de planxar-la a la SD mitjançant el programa “Raspberry Imager”. Aquesta és la única configuració que he necessitat dur a terme.

### Connexió amb la Raspberry

Com el que volem es controlar la RPi mitjançant SSH, vaig establir una connexió física a mitjançant Ethernet entre el portàtil i la Raspberry. Després, utilitzant la PowerShell de Windows iniciem sessió en la RPi de la següent manera:



```
PS C:\Users\ivanc> ssh ivan@raspberrypi.local
ivan@raspberrypi.local's password:
Linux raspberrypi 6.6.47+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.47-1+rpt1 (2024-09-02) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

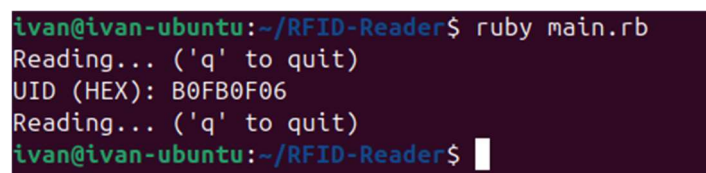
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Oct 6 11:54:33 2024 from fe80::bb49:900d:98a7:6195%eth0
ivan@raspberrypi:~ $
```

No obstant això, abans de dur a terme aquesta opció, vaig provar a fer servir SSH a través d'una màquina virtual amb Ubuntu, però sense tenir èxit. La complicació de fer-ho per una màquina virtual es que s'han de configurar les opcions de xarxa de aquesta d'una manera concreta (amb un segon adaptador de xarxa de tipus “Bridged Network”). De totes maneres, al iniciar la sessió SSH la màquina no trobava la Raspberry.

### Codi del “RFID-Reader”

El perifèric amb el que estic treballant és un lector que funciona exactament com si es tractés d'un teclat. De tal manera, la única llibreria que m'ha calgut importar és “io/console”, que ve per defecte amb Ruby i ens permetrà treballar amb la entrada del terminal i llegir les dades que ens dona el lector.

En quant al codi, el que se'ns demanava era mostrar el valor hexadecimal del UID que llegeix el lector. El codi de la solució que he proposat es troba a la següent pàgina. A continuació, adjunto una imatge del funcionament del programa:



```
ivan@ivan-ubuntu:~/RFID-Reader$ ruby main.rb
Reading... ('q' to quit)
UID (HEX): B0FB0F06
Reading... ('q' to quit)
ivan@ivan-ubuntu:~/RFID-Reader$
```

```
require 'io/console'

class RFID
  attr_accessor :uid

  # Initializes a new RFID instance with an optional UID.
  #
  # @param [String, nil] uid The UID of the RFID tag (optional).
  def initialize(uid = nil)
    self.uid = uid
  end

  # Converts the UID to a hexadecimal string.
  #
  # @return [String, nil] The UID as a hexadecimal string in uppercase, or `nil`
  #   if the conversion to integer fails.
  # @raise [ArgumentError] If the UID cannot be converted to an integer.
  def hex_uid
    begin
      Integer(uid).to_s(16).upcase
    rescue ArgumentError
      nil
    end
  end
end

if __FILE__ == $0

  rf = RFID.new

  # Main loop to continuously read input until the user quits.
  loop do
    print "Reading... ('q' to quit)\n"
    input = IO.console.noecho(&:gets).chomp

    # Exit the loop if the user enters 'q' or 'Q'.
    break if input.downcase == 'q'

    # Check if the input consists only of digits.
    if input.match?(/\A\d+\z/)
      rf.uid = input
      hex_value = rf.hex_uid

      # Display the hexadecimal value if the conversion is successful.
      if hex_value
        puts "UID (HEX): #{hex_value}"
      else
        puts "Error: UID can't be converted to hexadecimal."
      end
    else
      # Display an error message if the input contains non-digit characters.
      puts "Error: UID contains non-digit characters."
    end
  end
end
```