

# Đề thi giữa kỳ môn IoT

TS. Lê Trọng Nhân

Tháng 11 năm 2021

## 1 Giới thiệu

Trong bài tập này, các em sẽ tiến hành mô phỏng lại một bước trong quá trình xây dựng hệ thống IoT trong thực tế. Đó là, giải mã và xử lý các dữ liệu khi nhận về từ các thiết bị.

## 2 Đặc tả bài toán

Giả sử, các em đang phải xử lý dữ liệu là các thông tin được gửi một cách rời rạc, không liên tiếp từ các thiết bị trong hệ thống IoT. Bên cạnh đó, dữ liệu được gửi trong thực tế thường bị chen ngang bởi các dữ liệu nhiễu. Do đó, người xây dựng hệ thống IoT thường phải đặt ra các quy tắc về định dạng gửi dữ liệu để có thể dễ dàng hơn trong việc xác định được khi nào chuỗi dữ liệu bắt đầu và khi nào thì kết thúc chuỗi dữ liệu.

### 2.1 Định dạng dữ liệu

Quy ước: các em sẽ bắt đầu giải mã chuỗi dữ liệu khi gặp dấu ! và kết thúc bằng dấu #. Thông tin được gửi từ một thiết bị sẽ có định dạng:

***!ID:Number\_of\_Values:List\_of\_Values:CRC#***

Với định dạng các thông số được đặt tả theo quy ước như sau:

- **ID**: bắt đầu bằng các ký tự alphabet và kết thúc bằng các ký tự số.
- **Number\_of\_Values**: là số lượng các dữ liệu mà thiết bị phải gửi. Thông số này là như nhau đối với cùng 1 thiết bị.
- **List\_of\_Values**: bao gồm **Number\_of\_Values** các số nguyên, cách nhau bởi dấu ",".
- **CRC**: biến để kiểm tra tính đúng sai của dữ liệu. Nếu giá trị CRC bằng tổng của các giá trị ứng với thiết bị thì chuỗi dữ liệu đúng, ngược lại thì sai. Ngoài ra, còn một số lỗi khác cũng dẫn đến chuỗi dữ liệu không chính xác sẽ được trình bày ở phần 2.2

Chuỗi dữ liệu sau cùng phải giải mã sẽ bao gồm tổng hợp nhiều chuỗi dữ liệu được gửi từ các thiết bị.

Ví dụ:

**!device1:2:3,4:7#!device2:1:10,10#**

Tuy nhiên, bởi vì hệ thống phải xử lý 1 lượng lớn dữ liệu nên chuỗi dữ liệu cần giải mã có thể có nhiều chuỗi dữ liệu của các thiết bị được gửi tại các thời điểm khác nhau:

Ví dụ:

**!device1:2:3,4:7#!device2:1:10,10#!device1:2:4,5:9#**

Hơn nữa, hệ thống đang quản lý đã khá cũ nên đôi khi sẽ có những dữ liệu rác chen vào giữa các nhóm lệnh.

Ví dụ:

**112ToI!device1:1:5:5#!!!###abcd123device2:1:3#IoT211**

## 2.2 Xử lý dữ liệu

Do đó, yêu cầu của bài tập này là phải viết chương trình bằng Python để tách được các nhóm lệnh một cách hoàn chỉnh trong các trường hợp trên.

Sau khi đã được có được các nhóm lệnh hoàn chỉnh, chương trình cần phải lấy ra từng nhóm dữ liệu riêng biệt cho từng thiết bị. Và in ra màn hình câu lệnh theo định dạng sau:

**client.sent(ID,List\_of\_Values);**

Lưu ý: không có khoảng trắng giữa các dấu phẩy hay khoảng trắng cuối dòng.

Với từng thiết bị và giá tương ứng trong chuỗi dữ liệu mà các em vừa tách ra được theo thứ tự xuất hiện và các câu lệnh sẽ được in trên từng dòng, lệnh cuối cùng không có khoảng trắng hay xuống dòng.

Với chuỗi dữ liệu nhận được là là: **!device1:3:1,3,2:6#123!device2:1:2:2#** thì kết quả xuất ra màn hình sau quá trình giải mã sẽ là:

**client.sent(device1,1,3,2)  
client.sent(device2,2)**

Tuy nhiên, không phải lúc nào dữ liệu cũng được gửi chính xác mà đôi lúc vì các lỗi xảy ra trên đường truyền, hoặc do yếu tố chủ quan của người lập trình gửi dữ liệu, dẫn đến việc dữ liệu được tổng hợp không chính xác. Dữ liệu được xem là

không chính xác nếu như không thỏa mãn các ràng buộc được mô tả tại phần 2.1

Nếu tồn tại chuỗi dữ liệu ứng với một thiết bị nào đó trong chuỗi tổng hợp không theo định dạng trên. Chương trình sẽ xuất dòng **"ID" + "khoảng trắng" + "error"** ra màn hình. Với ID là tên thiết bị gửi dữ liệu lỗi. Còn đối với các chuỗi dữ liệu từ thiết bị nào đó không tồn tại ID thì chương trình chỉ xuất dòng **"error"**

Trường hợp chuỗi dữ liệu vừa tồn tại chuỗi đúng nhưng cũng vừa tồn tại chuỗi lỗi thì xuất ra kết quả tương ứng theo thứ tự xuất hiện của các dữ liệu trong chuỗi trên từng dòng.

Với chuỗi dữ liệu là: **!#ID002:2:10,50:60#** thì kết quả xuất ra màn hình sẽ là:

```
error
client.sent(ID002,10,50)
```

Trường hợp nếu một chuỗi dữ liệu được gửi, có ký tự bắt đầu là ! nhưng không có ký tự kết thúc # hoặc thiếu dữ liệu thì cũng xem như chuỗi lỗi.

Với chuỗi dữ liệu là: **!ID001:2!ID002:2:10,50:70#** thì kết quả xuất ra màn hình sẽ là:

```
ID001 error
ID002 error
```

### 3 Nộp bài

Các em sẽ làm bài trực tiếp tại phần **Mid-term Test** trên site BKeL môn học. Có một số testcase đơn giản được sử dụng để kiểm tra bài làm của các em nhằm đảm bảo rằng kết quả của em có thể biên dịch và chạy được. Các em có thể nộp bài bao nhiêu lần tùy ý nhưng chỉ có bài nộp cuối cùng được tính điểm. Vì hệ thống không thể chịu tải khi quá nhiều em nộp bài cùng một lúc, vì vậy em nên nộp bài càng sớm càng tốt. Các em sẽ tự chịu rủi ro nếu nộp bài sát hạn chót. Khi quá thời hạn nộp bài, hệ thống sẽ đóng nên các em sẽ không thể nộp nữa. Bài nộp qua email không được chấp nhận.

### 4 Gian lận

Các em phải tự mình hoàn thành bài tập này và phải ngăn không cho người khác đánh cắp kết quả của mình. Nếu không, các em sẽ bị xử lý theo quy định của trường vì gian lận.