|  |  |
| --- | --- |
| Học trực tuyến CNTT, học lập trình từ cơ bản đến nâng cao |  |

IOT301 – LẬP TRÌNH C CƠ BẢN

THUYẾT MINH ASSIGNMENT 2

XỬ LÝ BẢN TIN DỮ LIỆU CẢM BIẾN LƯU TRỮ TẠI BỘ ĐIỀU KHIỂN TRUNG TÂM

|  |  |
| --- | --- |
| Tên học viên:  Mã học viên:  Ngày báo cáo: | Hoàng Văn Trường  FX32766  08/03/2024 |

**HÀ NỘI, 03/2024**

**MỤC LỤC**

[1. Danh sách thư viện 1](#_Toc160831268)

[2. Danh sách các hàm 1](#_Toc160831269)

[3. Hàm fileToStr(char \*str) 2](#_Toc160831270)

[4. Hàm soBanTinGuiDi(char \*str) 2](#_Toc160831271)

[5. Hàm soBanTinGuiDenThietBi(char \*str) 3](#_Toc160831272)

[6. Hàm tachThuocTinh(char \*str) 3](#_Toc160831273)

[7. Hàm ktChuoiLap(char \*chuoi1, char \*chuoi2, char netEndp[18][2][25]) 4](#_Toc160831274)

[8. Hàm soCongTac() 4](#_Toc160831275)

[9. Hàm \*reqidBanTin(char \*chuoi, char banTin[27][50]) 5](#_Toc160831276)

[10. Hàm soBanTinGuiLoi() 5](#_Toc160831277)

[11. Hàm thoiGianTreLonNhat() 6](#_Toc160831278)

[12. Hàm thoiGianTreTrungBinh() 6](#_Toc160831279)

[13. Luồng xử lý code 8](#_Toc160831280)

**NỘI DUNG THUYẾT MINH**

# Danh sách thư viện

|  |  |  |
| --- | --- | --- |
| **STT** | **Thư viện** | **Mô tả** |
| 1 | stdio.h | Thư viện có các hàm nhập xuất  + printf: In dữ liệu ra màn hình  + scanf: Nhập liệu |
| 2 | string.h | Thư viện có các hàm làm việc với chuỗi kí tự  + strlen: Tính độ dài của một chuỗi  + strcpy: Dùng để sao chép một chuỗi vào một chuỗi khác  + strcmp: Dùng để so sánh hai chuỗi  + strstr: Tìm kiếm một chuỗi con trong một chuỗi  + … |
| 3 | stdbool.h | Thư viện định nghĩa kiểu dữ liệu logic  + true: 1  + false: 0 |
| 4 | stdlib.h | Thư viện cung cấp một loạt các hàm và định nghĩa macros để thực hiện các hoạt động như quản lý bộ nhớ, tạo số ngẫu nhiên, tạo và sắp xếp mảng, xử lý chuỗi, và nhiều công việc khác  + strtod: chuyển đổi một chuỗi thành một số kiểu double  + … |

# Danh sách các hàm

|  |  |  |
| --- | --- | --- |
| **STT** | **Danh sách hàm** | **Mô tả** |
|  | int main() | Hàm chính của chương trình |
|  | int fileToStr(char \*str) | Lưu trữ chuỗi str trong file log.txt vào mảng chuỗi fileStr |
|  | void soBanTinGuiDi(char \*str) | Hàm tính số bản tin gửi đi |
|  | void soBanTinGuiDenThietBi(char \*str) | Đếm bản tin gửi đi với mã network của thiết bị thực hiện truyền nhận dữ liệu |
|  | void tachThuocTinh(char \*str) | Tách các thuộc tính trong mỗi bản tin (Ví dụ: INFO, cmd, set, objects, type, ...) và lưu vào mảng thuocTinh |
|  | bool ktChuoiLap(char \*chuoi1, char \*chuoi2, char netEndp[18][2][25]) | Kiểm tra chuỗi lặp trong mảng netEndp |
|  | void soCongTac() | Thực hiện đếm số địa chỉ network và endpoint có trong danh sách |
|  | char \*reqidBanTin(char \*chuoi, char banTin[27][50]) | Lấy reqId của bản tin set hoặc status tùy thuộc vào chuỗi đưa vào là set hay status |
|  | void soBanTinGuiLoi() | Tính số bản tin lỗi |
|  | void thoiGianTreLonNhat() | Tính độ trễ lớn nhất giữa bản tin gửi đi và bản tin phản hồi |
|  | void thoiGianTreTrungBinh() | Tính thời gian trễ trung bình trong khoảng thời gian log |

# Hàm fileToStr(char \*str)

1. *Mô tả*

Hàm fileToStr(char \*str) sẽ copy toàn bộ nội dung của file log trong thư mục code vào biến tham số con trỏ str. Hàm này được bài cấp sẵn.

Tham khảo:

* [Hàm fread() trong C Thư viện C chuẩn (hoclaptrinh.vn)](https://hoclaptrinh.vn/posts/ham-fread-trong-c-thu-vien-c-chuan)
* [Đọc ghi File trong C - Hướng dẫn thao tác với File trong C (ironhackvietnam.edu.vn)](https://ironhackvietnam.edu.vn/cac-loai-tep-c/)

1. *Tham số truyền vào*

Con trỏ chuỗi \*str lưu trữ nội dung trong file log.txt.

1. *Giá trị trả về*

Hàm này trả về số lượng phần tử đã đọc thành công từ tệp tin với kiểu dữ liệu trả về là int.

# Hàm soBanTinGuiDi(char \*str)

1. *Mô tả*

Hàm soBanTinGuiDi(char \*str) sẽ in số bản tin gửi đi dựa vào chuỗi "cmd":"set" đươc tìm thấy bằng hàm strstr trên mỗi bản tin được phân tách bằng hàm strtok, hàm strtok phân tách bởi ký tự “\n”.

Tham khảo:

* [[C]. Hàm strtok() (28tech.com.vn)](https://blog.28tech.com.vn/c-ham-strtok)[Đọc ghi File trong C - Hướng dẫn thao tác với File trong C (ironhackvietnam.edu.vn)](https://ironhackvietnam.edu.vn/cac-loai-tep-c/)
* [cplusplus.com/reference/cstring/strtok/](https://cplusplus.com/reference/cstring/strtok/)
* [C cơ bản: Hàm strstr (deviot.vn)](https://deviot.vn/tutorials/c-co-ban.78025672/ham-strstr.09632275)

1. *Tham số truyền vào*

Con trỏ chuỗi \*str lưu trữ nội dung trong file log.txt.

1. *Giá trị trả về*

Hàm này trả về số lượng phần tử đã đọc thành công từ tệp tin với kiểu dữ liệu trả về là int.

# Hàm soBanTinGuiDenThietBi(char \*str)

1. *Mô tả*

Hàm soBanTinGuiDenThietBi(char \*str) sẽ đếm và in ra số bản tin gửi đi với mã network của thiết bị thực hiện truyền nhận dữ liệu được người dùng nhập vào.

Tham khảo:

* [[C]. Hàm strtok() (28tech.com.vn)](https://blog.28tech.com.vn/c-ham-strtok)[Đọc ghi File trong C - Hướng dẫn thao tác với File trong C (ironhackvietnam.edu.vn)](https://ironhackvietnam.edu.vn/cac-loai-tep-c/)
* [cplusplus.com/reference/cstring/strtok/](https://cplusplus.com/reference/cstring/strtok/)

1. *Tham số truyền vào*

Con trỏ chuỗi \*str lưu trữ nội dung trong file log.txt.

1. *Giá trị trả về*

Không có giá trị trả về. Hàm yêu cầu người dùng nhập vào mã network của thiết bị và in ra số bản tin đã gửi gi của thiết bị tương ứng.

# Hàm tachThuocTinh(char \*str)

1. *Mô tả*

Hàm tachThuocTinh(char \*str) tách các thuộc tính trong mỗi bản tin (Ví dụ: INFO, cmd, set, objects, type, ...) và lưu vào mảng thuocTinh. Số hàng của mảng thuộc tính tượng trưng cho số bản tin, số cột tương ứng với các thuộc tính, độ sâu của mảng thuocTinh thì lưu trữ các ký tự, chuỗi với kích thước nhất định.

Tham khảo:

* [[C]. Hàm strtok() (28tech.com.vn)](https://blog.28tech.com.vn/c-ham-strtok)[Đọc ghi File trong C - Hướng dẫn thao tác với File trong C (ironhackvietnam.edu.vn)](https://ironhackvietnam.edu.vn/cac-loai-tep-c/)
* [cplusplus.com/reference/cstring/strtok/](https://cplusplus.com/reference/cstring/strtok/)
* [Mảng đa chiều (tek4.vn)](https://tek4.vn/tutorial/cpp-tutorial/mang-da-chieu)

1. *Tham số truyền vào*

Con trỏ chuỗi \*str lưu trữ nội dung trong file log.txt.

1. *Giá trị trả về*

Không có giá trị trả về. Hàm lưu trữ thông tin các thuộc tính theo từng bản tin của mảng 3 chiều.

# Hàm ktChuoiLap(char \*chuoi1, char \*chuoi2, char netEndp[18][2][25])

1. *Mô tả*

Hàm ktChuoiLap(char \*chuoi1, char \*chuoi2, char netEndp[18][2][25]) sẽ kiểm tra chuỗi 1 và chuỗi 2 đã có trong mảng netEndp chưa. Nếu chưa thì trả về false, ngược lại thì trả bề true.

Tham khảo:

* [C cơ bản: Hàm strcmp (deviot.vn)](https://deviot.vn/tutorials/c-co-ban.78025672/ham-strcmp.06098724)

1. *Tham số truyền vào*

Con trỏ chuỗi \*chuoi1, \*chuoi2 chứa nội dung chuỗi muốn tìm trong mảng netEndp[18][2][25].

1. *Giá trị trả về*

Kiểu dữ liệu trả về là bool. Trong đó

* true : Tìm thấy chuỗi trong mảng netEndp
* false: Không tìm thấy chuỗi trong mảng netEndp

# Hàm soCongTac()

1. *Mô tả*

Hàm soCongTac() thực hiện phân tách các thuộc tính rồi đếm số địa chỉ network và endpoint có trong danh sách các bản tin qua hàm ktChuoiLap để tránh lặp lại thông tin các thiết bị.

Tham khảo:

* [C cơ bản: Hàm strcmp (deviot.vn)](https://deviot.vn/tutorials/c-co-ban.78025672/ham-strcmp.06098724)

1. *Tham số truyền vào*

Không có tham số truyền vào.

1. *Giá trị trả về*

Không có giá trị trả về. Hàm in ra số địa chỉ network và endpoint trong danh sách không trùng lặp.

# Hàm \*reqidBanTin(char \*chuoi, char banTin[27][50])

1. *Mô tả*

Hàm \*reqidBanTin(char \*chuoi, char banTin[27][50]) sẽ trả về reqId của bản tin set hoặc status tùy vào đối số chuỗi đưa vào là set hay status.

Tham khảo:

* [C cơ bản: Hàm strcmp (deviot.vn)](https://deviot.vn/tutorials/c-co-ban.78025672/ham-strcmp.06098724)

1. *Tham số truyền vào*

* Con trỏ chuỗi \*chuoi xác định bản tin là set hay status.
* char banTin[27][50]: Mảng chuỗi bản tin được dùng để tìm kiếm thông tin reqid trong từng bản tin. Mảng 2 chiều với 27 thuộc tính và độ dài chuỗi với mỗi thuộc tính là 50.

1. *Giá trị trả về*

Hàm này trả về reqId của bản tin set hoặc status.

# Hàm soBanTinGuiLoi()

1. *Mô tả*

Hàm soBanTinGuiLoi() sẽ in ra toàn bộ số bản tin lỗi dựa vào reqid của bản tin gửi đi và bản tin phản hồi. Nếu reqId của 2 bản tin bằng nhau thì biến đếm cộng thêm 1.

1. *Tham số truyền vào*

Không có tham số truyền vào.

1. *Giá trị trả về*

Không có giá trị trả về. Hàm này in ra số bản tin bị lỗi dựa vào reqId.

# Hàm thoiGianTreLonNhat()

1. *Mô tả*

Hàm thoiGianTreLonNhat() sẽ in ra độ trễ lớn nhất giữa bản tin gửi đi và bản tin phản hồi dựa trên hàm strtod chuyển đổi một chuỗi thành một số kiểu double đối với các bản tin không lỗi.

Tham khảo:

* [[C++]. Chuyển Đổi Giữa Số Và String Trong C++ (28tech.com.vn)](https://blog.28tech.com.vn/c-chuyen-doi-giua-so-va-string-trong-c)

1. *Tham số truyền vào*

Không có tham số truyền vào.

1. *Giá trị trả về*

Không có giá trị trả về. Hàm này in ra thời gian trễ lớn nhất giữa bản tin gửi đi và bản tin phản hồi với kiểu dữ liệu double.

# Hàm thoiGianTreTrungBinh()

1. *Mô tả*

Hàm thoiGianTreTrungBinh() sẽ in ra thời gian trễ trung bình trong khoảng thời gian log dựa trên hàm strtod chuyển đổi một chuỗi thành một số kiểu double đối với các bản tin không lỗi.

Tham khảo:

* [[C++]. Chuyển Đổi Giữa Số Và String Trong C++ (28tech.com.vn)](https://blog.28tech.com.vn/c-chuyen-doi-giua-so-va-string-trong-c)

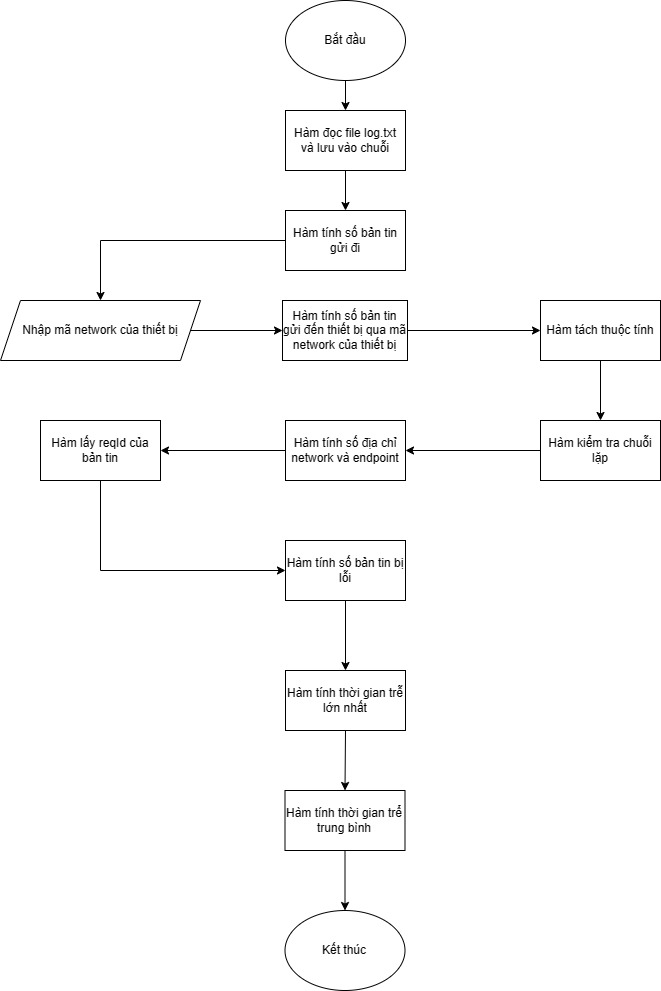
1. *Tham số truyền vào*

Không có tham số truyền vào.

1. *Giá trị trả về*

Không có giá trị trả về. Hàm này in ra thời gian trễ trung bình giữa bản tin gửi đi và bản tin phản hồi với kiểu dữ liệu double trong khoảng thời gian log.

# Luồng xử lý code



Các bước thực xử lý:

*Bước 1:* Đọc file log.txt và lưu chuỗi vào mảng fileStr

*Bước 2:* Tạo một mảng tạm và copy thông tin chuỗi trong mảng fileStr vào mảng tempLog (tránh các hàm sử dụng strtok làm thay đổi chuỗi gốc)

*Bước 3:* Tính số bản tin gửi đi

*Bước 4:* Tính số bản tin gửi đi theo mã network của thiết bị

*Bước 5:* Tính số công tắc có trao đổi thông tin với bộ điều khiển trung tâm

*Bước 6:* Tính số bản tin gửi đi bị lỗi

*Bước 7:* Tính độ trễ lớn nhất giữa bản tin gửi đi và bản tin phản hồi

*Bước 8:* Tính thời gian trễ trung bình trong khoảng thời gian log