Bài tập dài nhà số 3

Họ và tên sinh viên:

Mã số sinh viên: **Lớp:**

Bài tập dài này yêu cầu thiết kế và xây dựng phần mềm hiển thị file .hex, được format theo dạng intel hex file

1. Giới thiệu

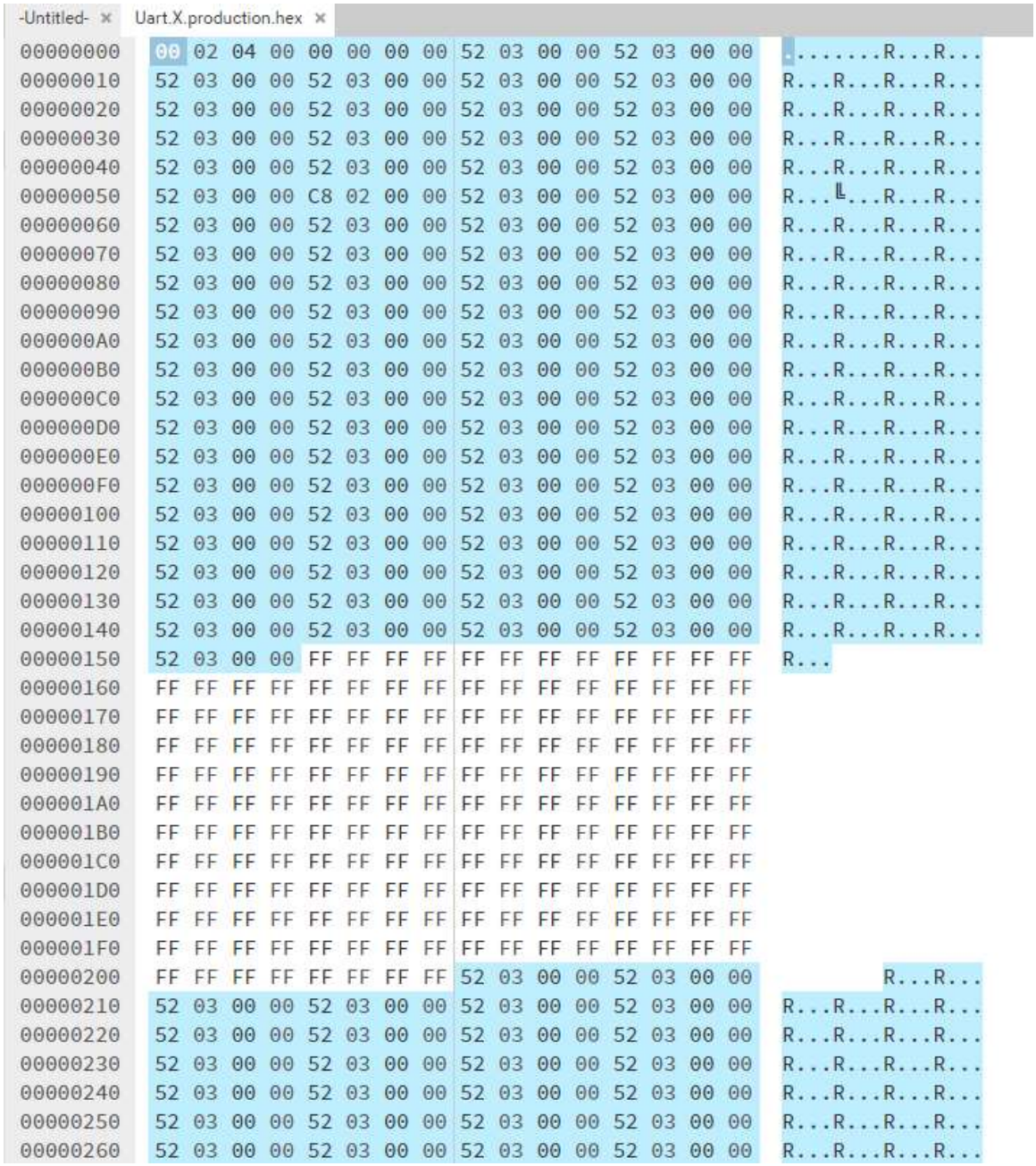
Intel hex file là một chuẩn format file hex thường dùng trong việc lưu trữ dữ liệu nhị phân trong các phần mềm làm việc với Vi điều khiển. Cấu trúc của Intel Hex file có thể được tham khảo ở trang sau: http://www.keil.com/support/docs/1584/.

Trong các chương trình sử dụng trong vi điều khiển, chương trình dịch (compiler) thường dịch ra file hex (hoặc file .bin), các file này cho phép nói với vi điều khiển là các đoạn mã hex của chương trình sẽ nằm đâu trong bộ nhớ. Để đảm bảo tính chính xác của dữ liệu, Intel Hex luôn đưa ra kiểm tra check sum và nhiệm vụ của lập trình viên là đảm bảo luôn kiểm tra check sum cho mỗi dòng dữ liệu được đọc vào. Theo tiêu chuẩn cũ thì kích thước của file lớn nhất là 64Kbytes, tức là khoảng 655336 ô trong bộ nhớ.

2. Yêu cầu

Chương trình sẽ mở một file vào và tìm hiểu xem file đấy có phải là file hex không ? Sau đó chương trình sẽ đọc từng dòng của file và cuối cùng hiển thị file đấy lên trên màn hình. Định dạng của hiển thị như hình vẽ sau, tham khảo file ví dụ với trang Web: https://hexed.it/, file (Uart.X.production.hex)

1. Vùng đầu tiên là địa chỉ của ô nhớ, lần lược đánh số từ 0000 -FFFF
2. Vùng thứ hai là nội dung ô nhớ, dưới dạng hex (các ô nhớ không có dữ liệu sẽ được mặc định là FF
3. Vùng thứ 3 là giá trị ASCII tương ứng với giá trị hex
4. Vùng nhớ cố định là 64Kbytes, lớn hơn vùng nhớ này sẽ không cần hiển thị.
5. Lưu ý là chương trình chỉ cần hiển thị dạng text đơn giản (không cần phải format màu như hình ví dụ).
6. Các file .hex mẫu được đính kèm với yêu cầu này.



3. Yêu cầu kỹ thuật

Chương trình sẽ bao gồm các chức năng sau:

- Chương trình sẽ chạy từ dòng lệnh (command – line), với tên file chạy là HexDisplay, sau đấy tên của file đầu vào
- Chương trình sẽ kiểm tra dạng xem có phải là file hex không, nếu không phải thì sẽ in ra thông báo lỗi

Sau khi đọc xong dữ liệu vào bộ nhớ, chương trình sẽ in nội dung màn hình, lưu ý mỗi trang màn hình sẽ có độ dài là 25 dòng, nếu chương trình dài hơn 25 dòng thì phần mềm hỏi người sử dụng có tiếp tục hiển thị nữa không, nếu có thì chương trình sẽ tự động xóa đi 25 dòng này và hiển thị 25 dòng tiếp theo, điều này tiếp tục cho khi đến

hết số dòng cần hiển thị. Lệnh xóa màn hình là lệnh *system("cls"); khai báo ở thư viện stdlib.h*

Xử lý lỗi

1. Usage: HexDisplay <input file>

Nếu một file mà không mở được thì sẽ có thông báo
Error: FILENAME could not be opened.
với tên file là file không mở được

Nếu trong chương trình có bất cứ dòng nào mà không đúng format của file Intel hex, thì chương trình sẽ thông báo lỗi tại dòng đó và kết thúc.

5 Thiết kế

Phương pháp thiết kế truyền thống (top – down). Bạn sẽ phải vẽ sơ đồ thiết kế hình cây và đưa ra giải thích về sự chuyển đổi thông tin giữa các phần của chương trình.

6 Code

Lưu ý, để dễ dàng cho việc chấm, mỗi hàm cần có lời giải thích ngắn gọn, đầu vào là gì, đầu ra là gì . Bạn nên đưa ra giải thích trong hàm khi có thể. Lưu ý việc sử dụng tên hàm, tên biến cho dễ hiểu và thống nhất. Các hàm phải cách nhau bằng hàng trống, tránh sử dụng các giá trị hằng.

7. Gởi ý
Sử dụng các hàm sau khi làm việc với file fgetc(), fgets(), ungetc(), fputc(), fprintf(), and rewind().
Sự dụng vùng nhớ đệm, ví dụ buffer[65536];
Sử dụng hàm đọc file để đọc dữ liệu vào bộ nhớ đệm
Sau khi đọc xong mới tiến hành gửi dữ liệu này ra màn hình.

MÔ TẢ CẤU TRÚC CÁC DÒNG CỦA INTEL HEX FILE

🔷 CẤU TRÚC TỪNG DÒNG

Mỗi dòng (gọi là **record**) trong file có cấu trúc sau:

:LLAAAATT[DD...]CC

Trong đó:

Thành Phần	Độ dài	Mô tả
:	1 ký tự	Bắt đầu của dòng
LL	2 hex	**Byte Count** - số byte dữ liệu trong dòng (tối đa 255)
AAAA	4 hex	**Address** - địa chỉ offset trong bộ nhớ (16-bit)
TT	2 hex	**Record Type** - loại bản ghi (xem bên dưới)
[DD...]	N x 2 hex	**Data** - dữ liệu (nếu có), N = LL
CC	2 hex	**Checksum** - dùng để kiểm tra lỗi dòng

🔶 CÁC LOẠI RECORD TYPE (TT)

TT	Ý nghĩa
00	**Data Record**: chứa dữ liệu tải vào bộ nhớ
01	**End Of File Record**: kết thúc file

🔷 VÍ DỤ

:10010000214601360121470136007EFE09D2190140

Giải thích:

- : → bắt đầu dòng
- 10 → 16 byte dữ liệu
- 0100 → offset địa chỉ 0x0100
- 00 → loại bản ghi là Data Record
- 21460136012147013600 7EFE09D21901 → 16 byte dữ liệu

- 40 → checksum

🔶 CHECKSUM TÍNH THẾ NÀO?

Checksum = **bổ sung 2** (two's complement) của tổng tất cả các byte từ LL đến hết DD...

Ví dụ dòng trên:

10 + 01 + 00 + 00 + 21 + 46 + 01 + 36 + ... + 01 = XX

Checksum = (256 - (XX mod 256)) & 0xFF = 0x40