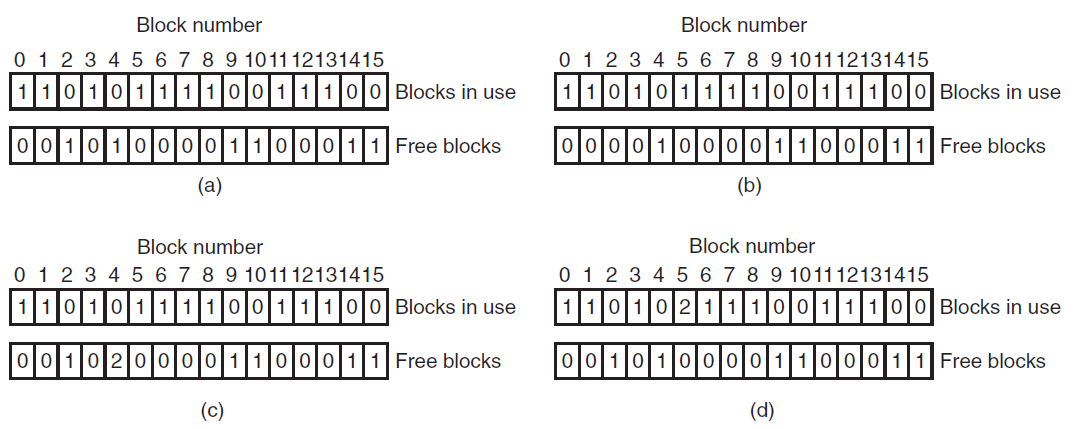
Mini project:

Viết lệnh checkdisk cho phép kiểm tra các lỗi khối đĩa trên hệ thống tệp iFS. Khi chạy lệnh sẽ đọc bảng inodes và trạng thái các khối đĩa rỗi vào một mảng 2 chiều như trong hình. Nếu tổng trên cột i khác 1 thì đã có lỗi xảy ra.



**Yêu cầu**

1. Chương trình cần phát hiện được các loại lỗi, thông báo ra màn hình.
2. Cài đặt chức năng sửa lỗi đối với khối đĩa bị thất lạc (không thuộc tệp nào và cũng không nằm trong danh sách các khối rỗi)
3. Khuyến khích cài thêm chức năng sửa các lỗi khác, nếu có thể
4. Soạn thảo chương trình trong tệp *checkdisk.c* cho sẵn và nộp tệp này. Không thay đổi tên tệp chương trình. Sinh viên được yêu cầu bắt buộc mô tả chi tiết thuật toán được cài đặt trong phần comment đầu file và viết comments chú thích cho từng lệnh.
5. Hạn nộp chương trình 12:00PM 25/12/2022. Form chỉ cho nộp 1 lần, cần kiểm tra cẩn thận trước khi nộp
6. Copy bài hoặc cho bạn khác copy sẽ nhận điểm 0

**Hướng dẫn viết chương trình**

Lưu ý: Tải lại toàn bộ file nguồn bản mới nhất từ thư mục chia sẻ chung.

Để tạo mẫu lỗi đĩa cho việc kiểm thử, chạy lệnh *badformat* trong hệ vỏ *fsshell*.

Hệ vỏ fsshell đã được cài đặt để nhận lệnh *checkdisk* và gọi hàm *fs\_checkDisk()*. Sinh viên chỉ cần viết đoạn mã cho hàm *fs\_checkDisk()* trong module *checkdisk.c* cho sẵn.

Bảng inodes trên đĩa đã được hệ vỏ fsshell tải vào mảng *inodeTable[]* khai báo trong module *fs.c*. Cấu trúc *inode\_t* được khai báo trong file *fs.h*.

Danh sách các khối đĩa rỗi được cài đặt trong hệ thống tệp iFS như sau

* Biến bản ghi *superBlock* trong module fs.c chứa các thông tin của bảng super block
* Trường *superBloc.freeBlockList* chỉ tới khối đĩa rỗi đầu tiên
* Các khối đĩa rỗi được móc nối với nhau thành danh sách móc nối. Hai bytes đầu của mỗi khối chứa số hiệu của khối đĩa rỗi tiếp theo
* Xem đoạn mã *// create free block chain* trong hàm fs\_format để hiểu về cách tạo móc nối các khối đĩa rỗi

Các hàm khác có liên quan trong module *diskemu.c*

* Hàm *writeBlock(i, &buf)* ghi nội dung bộ đệm buf xuống khối đĩa i
* Hàm *readBlock(i, &buf)* đọc nội dung khối đĩa i vào bộ đệm buf

Các hàm khác có liên quan trong module *fs.c*

* Hàm *getFreeBlock()* cấp phát 1 khối đĩa rỗi từ danh sách các khối rỗi
* Hàm *addFreeBlock(i)* bổ sung khối đĩa i vào danh sách các khối rỗi

Lưu ý, module *fs.c* gồm 2 nhóm hàm

* Các hàm được coi là public để sử dụng lại trong các module khác sẽ được đặt tên bắt đầu bằng tiền tố *fs\_*, ví dụ fs\_open(), fs\_close vv.
* Các hàm tiện ích chỉ được sử dụng trong nội tại module *fs.c* không có tiền tố *fs\_* và được khai báo *static* (hàm dành riêng trong module). Ví dụ các hàm getFreeBlock()