

## Bài tập thực hành 8

1. Viết chương trình ngôn ngữ C tên `unixecho.c` giả lập lệnh:  
`echo [string] > example.txt`  
*Gợi ý:* Có thể sử dụng hàm `freopen` để định hướng lại lỗi ra.
2. Viết chương trình ngôn ngữ C tên `unixmv.c` giả lập lệnh “mv” của shell.  
*Gợi ý:* sử dụng hàm `rename`.
3. Viết chương trình ngôn ngữ C tên `unixrm.c` giả lập lệnh “rm” của shell. Chương trình nhận 1 tham số đầu vào là đường dẫn tới file/thư mục cần xóa.  
*Gợi ý:* Viết hàm `rm_file_or_directory` nhận tham số đầu vào là đường dẫn tới file/thư mục cần xóa thực hiện:
  - Kiểm tra tham số đầu vào là đường dẫn tới file hay thư mục.
  - Nếu là file, xóa file sử dụng hàm `unlink`
  - Nếu là thư mục, duyệt tới các file/thư mục con trong thư mục đó (có thể sử dụng các hàm `opendir`, `readdir`, `closedir`)
    - + Nếu là file con, xóa file con sử dụng hàm `unlink`
    - + Nếu là thư mục con, gọi đệ quy tới hàm `rm_file_or_directory`
  - Sau bước này, tất cả các file con trong thư mục đã được xóa, thư mục trở thành thư mục rỗng -> có thể xóa thư mục rỗng sử dụng hàm `rmdir`*Lưu ý:* Khi duyệt tới các file/thư mục con trong thư mục cần xóa sử dụng hàm `readdir`, kết quả trả về bao gồm cả `.` (thư mục hiện tại) và `..` (thư mục cha của thư mục hiện tại), cần bỏ qua trường hợp `.` và `..`.
4. Viết chương trình ngôn ngữ C tên `unixgrep.c` giả lập lệnh “grep” sao cho từ được lọc đặt trong `argv[1]`, đường dẫn tới file đặt trong `argv[2]`.  
*Gợi ý:* có thể sử dụng hàm `fgets` để đọc từng dòng của file, sau đó so sánh nội dung của từng dòng với từ được lọc sử dụng hàm `strstr`.
5. Viết chương trình ngôn ngữ C tên `grepdir.c` nhận 2 tham số đầu vào là từ được lọc và đường dẫn tới thư mục. Chương trình sẽ liệt kê tất cả các file trong thư mục chứa từ được lọc.  
*Gợi ý:* Duyệt thư mục đầu vào, với mỗi file trong thư mục, sử dụng lại code chương trình `unixgrep` ở bài 4 để lọc nội dung của file với từ được lọc. Nếu từ được lọc xuất hiện trong file thì in ra tên file.