#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <dirent.h> // 디렉토리의 구조를 dirent로 정의하고 있는 헤더파일

#include <sys/types.h> // 어느 곳에서든지 사용되는 다양한 데이터 유형을 나타내는 헤더파일

int main(int argc, char\* argv[]) {

int argc : main()함수에 전달되는 데이터의 개수

char\* argv[] :main()함수에 전달되는 실제적인 데이터로 char형 포인터 배열로 구성

struct dirent\* dent;

디렉토리 파일에 속하는 파일들을 관리하는 dirent 구조체로 정의

if (argc != 2) {

printf("Error, 확인할 디렉토리명을 입력해주세요.\n");

exit(1);}

argc != 2

입력받은 데이터의 개수가 2개가 아닐 경우 디렉토리를 입력받지 못한 것과 같음

exit(1)

1를 반환하여 현재 프로그램 완전히 종료

if (argc == 2) {

입력받은 데이터의 개수가 2개일 경우 디렉토리를 입력받은 것과 같음

DIR\* dp = opendir(argv[1]);

디렉터리 포인터 DIR\* dp 선언

opendir()

디렉토리를 열어주는 open함수로 argv[1]로 입력받은 디렉토리를 열어준다.

if (dp == NULL) {

perror("Fail to open directory");

exit(1);}

디렉토리가 비어있을 경우 디렉토리를 여는 것에 실패했다는 오류문 출력

exit(1) : 1를 반환하여 현재 프로그램을 완전히 종료

closedir(dp);

opendir함수를 이용하여 열어준 디렉토리 파일을 closedir()함수를 이용하여 닫아준다.

if (rmdir(argv[1]) == -1) {

printf("%s 디렉토리는 비어있지 않습니다.\n", argv[1]);

}rmdir()

비어있는 디렉토리일 경우 해당 디렉토리를 삭제해준다.

디렉토리 내에 파일이 존재하거나 사용중이면 삭제할 수 없다.

정상 동작할 경우 0 반환, 에러 발생 시 -1를 반환한다.

rmdir(argv[1]) : 삭제할 디렉토리의 경로 argv[1]로 설정

else{

printf("%s 디렉토리가 비어있어 해당 디렉토리를 삭제합니다.\n", argv[1]);

}

argv[1]가 비어있을 경우 0을 반환하며 해당 디렉토리를 삭제한다.

return 0;

main 함수를 끝낸 후 벗어나기 위하여 0을 return 한다.