컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include <dirent.h>

#include <fcntl.h>

#include <string.h>

#include <sys/stat.h>

#include "ssu\_runtime.h"

#define DIRECTORY\_SIZE MAXNAMLEN

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

{

struct dirent \*dentry;

struct stat statbuf;

char filename[DIRECTORY\_SIZE];

DIR \*dirp;

gettimeofday(&begin\_t, NULL);

if (argc < 2) {

fprintf(stderr, "usage : %s <directory>\n", argv[0]);

exit(1);

}

if ((dirp = opendir(argv[1])) == NULL || chdir(argv[1]) == -1) { // 첫번째 인자로 받은 폴더를 열고 그 파일로 폴더 변경

fprintf(stderr, "opendir, chdir error for %s\n", argv[1]); // 둘중 하나라도 실패 시 출력 후

exit(1); // 종료

}

while ((dentry = readdir(dirp)) != NULL) { //열려진 폴더안의 읽어들일 파일이 없을때까지

if (dentry->d\_ino == 0) { // 디렉토리의 아이노드가 0이라면

continue; // 다음 반복문 수행

}

memcpy(filename, dentry->d\_name, DIRECTORY\_SIZE); // filename변수에 파일명을 비트단위 복사

if ((stat(filename, &statbuf)) == -1) { // 그 파일의 stat구조체를 저장

fprintf(stderr, "stat error for %s\n", filename); // 실패 시 출력 후

break; // 반복문 탈출

}

if ((statbuf.st\_mode & S\_IFMT) == S\_IFREG) { // 파일이 레귤러파일이면

printf("%-14s %lld\n", filename, statbuf.st\_size); // 파일명과 사이즈 출력

}

else {

printf("%-14s\n", filename); // 아니면 파일명만 출력

}

}

gettimeofday(&end\_t, NULL);

ssu\_runtime(&begin\_t, &end\_t);

exit(0);

}

1. 실행 결과

