컴퓨터학부 20201841 박세연

<filemacro>

1. 소스코드

**[ssu\_file\_marco.c]**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/stat.h>

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

struct stat file\_info; // 파일의 정보를 저장할 stat 구조체 선언

char \*str;

int i;

for(i = 1; i < argc; i++){

printf("name = %s, type = ", argv[i]);

if (lstat(argv[i], &file\_info) < 0){ // lstat 함수로 파일의 정보를 file\_info에 받음

fprintf(stderr, "lstat error\n"); // 에러처리

continue;

}

// file\_info의 st\_mode는 파일의 형식을 가지고 있음

// 파일의 종류를 확인하는 매크로를 통해 파일의 종류 출력

if (S\_ISREG(file\_info.st\_mode)) // 일반 파일인 경우

str = "regular";

else if (S\_ISDIR(file\_info.st\_mode)) // 디렉터리인 경우

str = "directory";

else if (S\_ISCHR(file\_info.st\_mode)) // 문자 특수 파일인 경우

str = "character special";

else if (S\_ISBLK(file\_info.st\_mode)) // 블럭 특수 파일인 경우

str = "block special";

else if (S\_ISFIFO(file\_info.st\_mode)) // 프로세스간 통신에 쓰이는 파일인 경우

str = "FIFO";

else if (S\_ISLNK(file\_info.st\_mode)) // 심볼릭링크인 경우(다른 파일을 가리키는 파일)

str = "symbolic link";

else if (S\_ISSOCK(file\_info.st\_mode)) // 프로세스간 네트워크 통신에 쓰이는 파일인 경우

str = "socket";

else

str = "unknown mode";

printf("%s\n", str);

}

exit(0);

}

1. 실행결과

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

<access1>

1. 소스코드

**[ssu\_access\_1.c]**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

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

int i;

if (argc < 2) {

fprintf(stderr, "usage: %s <file1> <file2>.. <fileN>\n", argv[0]);

exit(1);

}

for (i = 1; i < argc; i++) {

if (access(argv[i], F\_OK) < 0) { // access 함수로 F\_OK를 통해 파일의 존재여부를 확인

fprintf(stderr, "%s doesn't exist.\n", argv[i]); // 존재하지 않으면 에러처리

continue;

}

if (access(argv[i], R\_OK) == 0) // access 함수로 R\_OK를 통해 읽기 권한 여부를 확인

printf("User can read %s\n" ,argv[i]);

if (access(argv[i], W\_OK) == 0) // access 함수로 W\_OK를 통해 쓰기 권한 여부를 확인

printf("User can write %s\n", argv[i]);

if (access(argv[i], X\_OK) == 0) // access 함수로 X\_OK를 통해 실행 권한 여부를 확인

printf("User can execute %s\n", argv[i]);

}

exit(0);

}

2. 실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<access2>

1. 소스코드

**[ssu\_access\_2.c]**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#define TABLE\_SIZE (sizeof(table)/sizeof(\*table))

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

{

struct {

char \*text;

int mode;

} table [] = {

{"exists", 0},

{"execute", 1},

{"write", 2},

{"read", 4}

};

int i;

if (argc < 2) {

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

exit(1);

}

for (i = 0; i < TABLE\_SIZE; i++){

if (access(argv[1], table[i].mode) != -1) // 차례대로 파일의 존재 여부, 실행 권한, 쓰기 권한, 읽기 권한을 access 함수로 확인함

printf("%s -ok\n", table[i].text); // 권한이 존재하면 ok 출력

else

printf("%s\n", table[i].text);

}

exit(0);

}

2. 실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<umask>

1. 소스코드

**[ssu\_umask.c]**

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <sys/stat.h>

#define RW\_MODE (S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IWGRP | S\_IROTH | S\_IWOTH)

int main(void){

char \*fname1 = "ssu\_file1";

char \*fname2 = "ssu\_file2";

umask(0);

if (creat(fname1, RW\_MODE) < 0) { // 위에서 선언해준 RW\_MODE로 파일을 생성 (rw-rw-rw-)

fprintf(stderr, "creat error for %s\n", fname1);

exit(1);

}

umask(S\_IRGRP | S\_IWGRP | S\_IROTH | S\_IWOTH);

// umask 함수를 통해 파일 생성 시에 기본 권한을 제한하는 마스크를 일시적으로 변경함

// 현재 넘긴 인자로는 Gropu의 읽기, 쓰기 권한과

// Other의 읽기, 쓰기 권한이 제한됨

if (creat(fname2, RW\_MODE) < 0){

fprintf(stderr, "creat error for %s\n", fname2);

exit(1);

}

exit(0);

}

2. 실행결과

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명