컴퓨터학부 2020292 조민혁

1. 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/stat.h>

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

struct stat file\_info;

char \*str;

int i;

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

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

if(lstat(argv[i], &file\_info)<0){

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

continue;

} //lstat에러가 발생하면 메세지를 출력하고 for문을 한 번 건너뛴다.

if(S\_ISREG(file\_info.st\_mode))

str="regular";

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

str="directory";

else if(S\_ISCHR(file\_info.st\_mode))

str="character special";

else if(S\_ISBLK(file\_info.st\_mode))

str="block special";

else if(S\_ISFIFO(file\_info.st\_mode))

str="FIFO";

else if(S\_ISLNK(file\_info.st\_mode))

str="symbolic link";

else if(S\_ISSOCK(file\_info.st\_mode))

str="socket";

else

str="unknown mode";

}// 인자로 받은 name을 출력한다.file\_info의 st\_mode에 따라 다른 type을 출력한다.

exit(0);

}  
스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

#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);

}//인자가 적은 경우에 에러메세지를 출력한다.

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

if(access(argv[i],F\_OK)<0){

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

continue;

}

if(access(argv[i],R\_OK) ==0 )

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

if(access(argv[i],W\_OK)==0)

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

if(access(argv[i],X\_OK)==0)

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

}//입력하는 인자의 갯수만큼 반복하여 access 했을 때 파일 접근 권한에 대해서 printf해준다.

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

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

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

struct {

char \*text;

int mode;

}table[] = {

{"exists", 0},

{"execute", 1},

{"write", 2},

{"read", 4}

};//구조체를 선언하고 table이라는 배열을 만들었다.

int i;

if(argc < 2){

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

exit(1);

} //인자가 2개보다 작으면 에러 메세지를 출력한다.

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

if(access(argv[1], table[i].mode) != -1)

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

else

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

}//4번 반복하면서 argv[1]의 접근권하에 대하여 -ok가 붙어져서 출력되게 한다.

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

#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);//umask값을 0으로 설정한다.

if(creat(fname1,RW\_MODE)<0){

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

exit(1);

}//fname1의 값인 ssu\_file1을 이름으로 가진 파일을 만든다.

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

//umasek값을 다시 한 번 변경해준다.

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

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

exit(1);

}

//fname2인 ssu\_file2를 이름으로 파일을 생성한다

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명