컴퓨터학부 20202920 조민혁

1.ssu\_chmod\_1.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/stat.h>

int main(void){

struct stat statbuf;

char \*fname1 = "ssu\_file1";

char \*fname2 = "ssu\_file2";

if(stat(fname1,&statbuf) < 0){

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

}

if(chmod(fname1, (statbuf.st\_mode & ~S\_IXGRP) | S\_ISUID) < 0)

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

//fname1의 모드를 변경시켜준다.

if(chmod(fname2, S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IROTH | S\_IXOTH) < 0 )

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

//fname2의 모드를 변경시켜준다.

//두 가지의 방식으로 모드를 변경시킬 수 있다는 것을 알려주는 코드이다.

exit(0);

}

2. 실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

3.ssu\_chmod\_2.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/stat.h>

#define MODE\_EXEC (S\_IXUSR|S\_IXGRP|S\_IXOTH)

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

struct stat statbuf;

int i;

if(argc < 2){

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

exit(1);

} //인자가 적게 들어왔을 때 에러메세지를 출력하며 프로그램을 종료하는 것이다.

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

if(stat(argv[i], &statbuf) < 0){

fprintf(stderr , "%s : stat error\n",argv[i]);

continue;

}

statbuf.st\_mode |= MODE\_EXEC;

statbuf.st\_mode ^= (S\_IXGRP|S\_IXOTH);

if (chmod(argv[i], statbuf.st\_mode) < 0)

fprintf(stderr, "%s : chmod error\n",argv[i]);

else

printf("%s : file permission was changed.\n",argv[i]);

}//반복문을 통해 인자로 들어온 파일들의 mode를 변경시켜 줍니다.

exit(0);

}

4.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

5.ssu\_chown.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/stat.h>

int main(void){

struct stat statbuf;

char \*fname = "ssu\_myfile";

int fd;

if((fd = open(fname, O\_RDWR | O\_CREAT, 0600)) < 0){

fprintf(stderr , "open error for %s\n", fname);

exit(1);

}

//fname을 이름으로 갖는 파일을 없으면 생성하고 있는 경우 RDWR모드로 열어줍니다

close(fd);

stat(fname, &statbuf);

printf("# 1st stat call # UID:%d GID:%d\n",statbuf.st\_uid,statbuf.st\_gid);

if(chown(fname,501,300) < 0){

fprintf(stderr , "chown error for %s\n",fname);

exit(1);

}

//파일 사용자 id와 그룹 사용자 id를 변경해줍니다.

stat(fname,&statbuf);

printf("# 2nd stat call # UID:%d GID:%d\n",statbuf.st\_uid,statbuf.st\_gid);

if(unlink(fname) < 0){

fprintf(stderr, "unlink error for %s\n",fname);

exit(1);

}

//기존 디렉토리 항목의 링크를 제거해준다.

exit(0);

}

6.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

7.ssu\_link.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

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

{

if(argc < 3 ) {

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

exit(1);

}

if(link(argv[1],argv[2]) == -1){

fprintf(stderr, "link error for %s\n",argv[1]);

exit(1);

}

//인자로 입력된 두 개의 파일에 링크를 생성해준다.

exit(0);

}

8.ssu\_oslab의 소스코드

#include <stdio.h>

#include <stdlib.h>

int main(void){

printf("This is oslab file\n");

exit(0);

}

9.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

10.ssu\_unlink\_1.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

int main(void){

char \*fname = "ssu\_dump.txt";

if(open(fname, O\_RDWR) < 0) {

fprintf(stderr, "open error for %s\n",fname);

exit(1);

}

//파일을 열어줍니다.

if(unlink(fname) <0 ){

fprintf(stderr, "unlink error for %s\n", fname);

exit(1);

}

//링크를 제거해줍니다.

printf("File unlinked\n");

sleep(20);

printf("Done\n");

//sleep함수를 통해 기다려준 후 Done을 출력합니다.

exit(0);

}

11.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

12.ssu\_unlink\_2.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

int main(void){

char buf[64];

char \*fname = "ssu\_tempfile";

int fd;

int length;

if((fd = open(fname, O\_RDWR|O\_CREAT|O\_TRUNC, 0600)) < 0){

fprintf(stderr, "first open error for %s\n",fname);

exit(1);

}

//파일을 열어줍니다.

if(unlink(fname) < 0){

fprintf(stderr , "unlink error for %s\n", fname);

exit(1);

}

//파일의 링크를 해제하여 줍니다.

if(write(fd,"How are you?", 12) != 12){

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

exit(1);

}//파일에 write해줍니다.

lseek(fd ,0, 0);

if((length = read(fd, buf, sizeof(buf)))<0){

fprintf(stderr, "buf read error\n");

exit(1);

}

else

buf[length] = 0;

//read함수를 통해 파일에 적힌 내용을 읽어들입니다.

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

close(fd);

if((fd = open(fname, O\_RDWR)) < 0){

fprintf(stderr, "second open error for %s\n",fname);

exit(1);

}

else

close(fd);

//error가 발생하지 않으면 열은 후에 닫아준다.

exit(0);

}

13.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

14.ssu\_remove.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

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

if(argc != 3){

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

exit(1);

}

if(link(argv[1],argv[2]) < 0 ){

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

exit(1);

}//인자로 들어온 두 개의 파일을 link시켜준다.

else

printf("step1 passed.\n");

if(remove(argv[1])<0){

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

remove(argv[2]);

exit(1);

}//첫번째 인자로 들어온 파일이 remove에 실패한 경우 두번째 인자를 remove해준다.

else

printf("step2 passed.\n");

printf("Success!\n");

exit(0);

}

15.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

16.ssu\_rename.c의 소스코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

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

{

int fd;

if(argc!=3){

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

exit(1);

}

if((fd = open(argv[1], O\_RDONLY)) < 0 ){

fprintf(stderr, "first open error for %s\n",argv[1]);

exit(1);

}//첫번째 인자로 들어온 파일을 open 해준다.

else

close(fd);

if(rename(argv[1], argv[2]) < 0 ){

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

exit(1);

}

//첫번째 인자로 들어온 파일의 이름을 두 번째 인자의 값으로 바꿔준다.

if((fd = open(argv[1],O\_RDONLY)) < 0)

printf("second open for %s\n", argv[1]);

else {

fprintf(stderr, "it's very strange!\n");

exit(1);

}

//첫번째 인자 값으로 파일을 열어보고 안열리면 printf로 출력해준다.

if((fd = open(argv[2],O\_RDONLY))<0){

fprintf(stderr , "third open error for %s\n",argv[2]);

exit(1);

}

//두 번째 인자의 값으로 파일을 open합니다.

printf("Everything is good!\n");

exit(0);

}

17.실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명