컴퓨터학부 20202920 조민혁

1.ssu\_symlink.c의 소스코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

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

if(argc != 3){

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

exit(1);

}//인자가 3개가 아니면 에러메세지를 리턴한다.

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

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

exit(1);

}//인자로 들어온 것들을 이용하여 argv[2]를 argv[1]의 심볼릭 링크 파일을 생성한다.

else

printf("symlink: %s -> %s\n", argv[2], argv[1]);

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

2.ssu\_fopen.c의 소소코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

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

char \*mode = "r";

if(fopen(fname,mode) == NULL){

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

exit(1);

}//파일을 오픈시켜준다.

else

printf("Success!\nFilename: <%s>, mode: <%s>\n",fname,mode);

//열린 파일에 대한 이름과 모드를 출력합니다.

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

3.ssu\_freopen.c의 소스코드와 실행결과

#include <fcntl.h>

#include <stdio.h>

#include <stdlib.h>

int main(void){

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

int fd;

printf("First printf : Hello, OSLAB!!\n");

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

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

exit(1);

}//파일을 RDONLY로 열어준다.

if(freopen(fname,"w", stdout) != NULL)

printf("Second printf : Hello, OSLAB!!\n");

//ssu\_test.txt 파일로 stdout을 쓰기 모드로 다시 오픈하여 써준다.

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

4.ssu\_utime.c의 소스코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <utime.h>

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

struct utimbuf time\_buf;

struct stat statbuf;

int fd;

int i;

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

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

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

continue;

}//인자로 들어온 것에 대한 stat구조체를 얻기 위해 stat함수를 사용한다.

if((fd = open(argv[i], O\_RDWR | O\_TRUNC)) < 0){

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

continue;

}//해당 파일을 open 한다.

close(fd);

time\_buf.actime = statbuf.st\_atime;

time\_buf.modtime = statbuf.st\_mtime;

//tiem\_buf에 stat구조체의 atime 과 mtime을 대입시킨다.

if(utime(argv[i], &time\_buf) < 0){

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

continue;

}//utime함수를 사용하여 최종 접근 시간과 최정 변경 시간을 변경해준다.

}

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

5.ssu\_directory\_1.c의 소스코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <dirent.h>

#include <fcntl.h>

#include <string.h>

#include <sys/stat.h>

#define DIRECTORY\_SIZE MAXNAMLEN

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

struct dirent \*dentry;

struct stat statbuf;

char filename[DIRECTORY\_SIZE+1];

DIR \*dirp;

if (argc < 2){

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

exit(1);

}

//인자가 2개보다 작으면 오류를 출력한다.

if((dirp = opendir(argv[1]))==NULL || chdir(argv[1])== -1){

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

exit(1);

}

//dir를 열어주고 파일 이름만 넣어줄 때도 접근할 수 있게끔 작업디렉토리를 변경시켜준다.

while((dentry = readdir(dirp)) != NULL){

if( dentry->d\_ino == 0)

continue;

memcpy(filename, dentry->d\_name,DIRECTORY\_SIZE);

//메모리 값을 복사해서 넣어준다.

if(stat(filename, &statbuf) == -1){

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

break;

}//stat구조체를 얻기 위해 stat함수를 수행한다.

if((statbuf.st\_mode & S\_IFMT) == S\_IFREG)

printf("%-14s %ld\n",filename, statbuf.st\_size);

else

printf("%-14s\n",filename);

}

//mode에 대한 조건을 통해 if문으로 분기한다.

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

6.ssu\_directory\_2.c의 소스코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <dirent.h>

#include <limits.h>

#include <string.h>

#include <sys/stat.h>

#ifdef PATH\_MAX

static int pathmax = PATH\_MAX;

#else

static int pathmax = 0;

#endif

#define MAX\_PATH\_GUESSED 1024

#ifndef LINE\_MAX

#define LINE\_MAX 2048

#endif

char \*pathname;

char command[LINE\_MAX], grep\_cmd[LINE\_MAX];

int ssu\_do\_grep(void){

struct dirent \*dirp;

struct stat statbuf;

char \*ptr;

DIR \*dp;

if(lstat(pathname, &statbuf) < 0 ){

fprintf(stderr, "lstat error for %s\n",pathname);

return 0;

}//pathname에 대한 stat구조체를 얻기 위해서 lstat를 사용한다.

if(S\_ISDIR(statbuf.st\_mode) == 0){

sprintf(command, "%s %s", grep\_cmd, pathname);

printf("%s : \n",pathname);

system(command);

return 0;

}//디렉토리가 아닌 경우에 대한 예외처리를 작성해준다.

ptr = pathname + strlen(pathname);

\*ptr++ = '/';

\*ptr = '\0';

//ptr뒤에 /과 null을 붙혀준다.

if((dp = opendir(pathname)) == NULL){

fprintf(stderr, "opendir error for %s\n",pathname);

return 0;

}

//dir를 열어준다.

while((dirp = readdir(dp)) != NULL)

if(strcmp(dirp->d\_name,".")&& strcmp(dirp->d\_name,"..")){

strcpy(ptr,dirp->d\_name);

//dir 이름이 ..과 ..이 아닌 경우에 ptr에 이름을 넣어준다. 그 후에 ssu\_do\_grep함수를 사용해준다.

if(ssu\_do\_grep() < 0)

break;

}

ptr[-1] = 0;

closedir(dp);

return 0;

}

void ssu\_make\_grep(int argc, char \*argv[]){

int i;

strcpy(grep\_cmd, " grep");

//grep\_cmd에 grep이라는 문자열을 복사해서 넣어준다.

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

strcat(grep\_cmd, " ");

strcat(grep\_cmd, argv[i]);

}

}

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

if(argc < 2){

fprintf(stderr, "usage: %s <-CVbchilnsvwx> <-num> <-A num> <-B num> <-f file>\n"

" <-e> expr <directory>\n", argv[0]);

exit(1);

}//인자가 2개보다 작으면 에러메세지를 출력한다.

if(pathmax == 0){

if((pathmax = pathconf("/", \_PC\_PATH\_MAX)) < 0)

pathmax = MAX\_PATH\_GUESSED;

else

pathmax++;

}

if((pathname = (char \*) malloc(pathmax +1)) ==NULL){

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

exit(1);

} //pathname에 pathmax보다 크게 동적할당을 해준다.

strcpy(pathname, argv[argc-1]);

//pathname을 넣어준다.

ssu\_make\_grep(argc,argv);

//ssu\_make\_grep함수를 실행한다.

ssu\_do\_grep();

//ssu\_do\_grep함수를 실행한다.

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

7.ssu\_chdir.c의 소스코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

int main(void){

if(chdir("/etc") < 0){

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

exit(1);

}//현재 작업 디렉토리를 변경시켜줍니다.

printf("chdir to /etc succeeded.\n");

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

8.ssu\_getcwd.c의 소스코드와 실행결과

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#define PATH\_MAX 1024

int main(void)

{

char \*pathname;

if(chdir("/home/minhyuk") < 0){

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

exit(1);

}

//현재 작업디렉토리를 변경시켜줍니다.

pathname = malloc(PATH\_MAX);

if(getcwd(pathname, PATH\_MAX) == NULL){

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

exit(1);

}//현재 작업 중인 디렉토리를 확인합니다.

printf("current directory = %s\n", pathname);

exit(0);

}

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명