20192403 컴퓨터학부 박상철

write\_1.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#define BUFFER\_SIZE 1024

/\*

입력 받은 만큼 표준 출력으로 출력하는 코드

\*/

int main(void){

char buf[BUFFER\_SIZE];

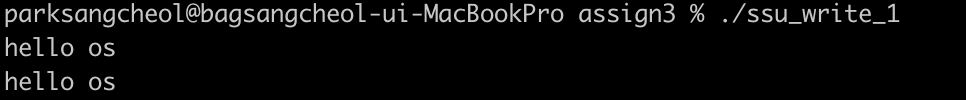
int length;

length=read(0,buf,BUFFER\_SIZE);

write(1,buf,length); // 리턴 값 : 성공 시 기록된 바이트 수, 에러 시 -1

exit(0);

}



write\_2.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#define S\_MODE 0644

#define BUFFER\_SIZE 1024

/\*

인자로 들어온 파일을 열어서 두번 째 인자로 들어온 파일에 작성하는 코드

\*/

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

char buf[BUFFER\_SIZE];

int fd1,fd2;

int length;

if(argc!=3){ // 인자가 3개가 아니면 에러

fprintf(stderr,"Usage : %s filein fileout\n",argv[0]);

exit(1);

}

if((fd1=open(argv[1],O\_RDONLY))<0){ // 첫 인자로 들어온 파일 읽기 모드로 오픈

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

exit(1);

}

if((fd2=open(argv[2],O\_WRONLY|O\_CREAT|O\_TRUNC,S\_MODE))<0){ // 두번 째 인자로 들어온 파일 쓰기 모드로 오픈

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

exit(1);

}

while((length=read(fd1,buf,BUFFER\_SIZE))>0) // 첫 인자의 파일 내용을 최대 1024까지 읽고 두 번째 인자로 들어온 파일에 쓰기

write(fd2,buf,length);

exit(0);

}

텍스트이(가) 표시된 사진

자동 생성된 설명

write\_3.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include "ssu\_employee.h"

/\*

.을 입력 받을 때까지 정보를 입력받아 구조체에 담아서 파일에 작성하는 코드

\*/

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

struct ssu\_employee record;

int fd;

if(argc<2){

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

exit(1);

}

if((fd=open(argv[1], O\_WRONLY | O\_CREAT | O\_EXCL, 0640))<0){ // 첫 번째 인자로 들어온 파일 쓰기 모드로 오픈

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

exit(1);

}

while(1){

printf("Enter employee name <SPACE> salary: ");

scanf("%s",record.name); // 구조체 이름 변수 입력

if(record.name[0]=='.') // 이름으로 들어 온 값이 .이면 종료

break;

scanf("%d",&record.salary); // 구조체 임금 변수 입력

record.pid = getpid();

write(fd,(char \*)&record, sizeof(record)); // 파일에 구조체 정보 작성

}

close(fd);

exit(0);

}

텍스트이(가) 표시된 사진

자동 생성된 설명

dup.c

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <fcntl.h>

#include <unistd.h>

#define BUFFER\_SIZE 1024

/\*

읽기 전용으로 파일을 열고 파일 디스크립터 복사해서 각 디스크립터로 출력

\*/

int main(void){

char buf[BUFFER\_SIZE];

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

int fd1,fd2;

int count;

if((fd1=open(fname,O\_RDONLY,0644))<0){

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

exit(1);

}

fd2=dup(fd1); // 리턴 값 : 성공 시 새로운 파일 디스크립터, 에러 시 -1 리턴

count=read(fd1,buf,12); // fd1으로 12 바이트 읽기

buf[count]=0;

printf("fd1's printf : %s\n",buf);

lseek(fd1,1,SEEK\_CUR);

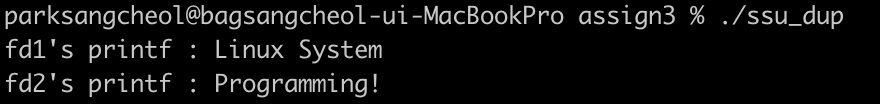
count=read(fd2,buf,12); // fd2로 뒤에 12 바이트 읽기

buf[count]=0;

printf("fd2's printf : %s\n",buf);

exit(0);

}



dup2\_1.c

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

#include <fcntl.h>

/\*

파일 디스크립터에 값을 표준 출력으로 복사하는 코드

\*/

int main(void){

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

int fd;

if((fd=creat(fname,0666))<0){ // 읽기 전용으로 오픈

printf("creat error for %s\n",fname);

exit(1);

}

printf("First printf is on the screen.\n");

dup2(fd,1); // fd 값을 1(표준 출력으로 복사)

printf("Second printf is in this file.\n"); // fd 파일이 표준 출력이 되어 이 글이 파일 안으로 들어감

exit(0);

}

텍스트이(가) 표시된 사진

자동 생성된 설명

dup2\_2.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#define BUFFER\_SIZE 1024

int main(void){

char buf[BUFFER\_SIZE];

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

int fd;

int length;

if((fd=open(fname,O\_RDONLY,0644))<0){

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

exit(1);

}

if(dup2(1,4)!=4){ // 1번 파일 디스크립터(표준 출력을) 4번으로 복사 -> 4번 파일 디스크립터를 쵸준 출력 처럼 쓸 수 있음

fprintf(stderr,"dup2 call failed\n");

exit(1);

}

while(1){

length=read(fd,buf,BUFFER\_SIZE);

if(length<=0)

break;

write(4,buf,length);

}

exit(0);

}

텍스트이(가) 표시된 사진

자동 생성된 설명