컴퓨터학부 20201841 박세연

<dup1>

1. 소스코드

**[ssu\_dup.c]**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#define BUFFER\_SIZE 1024

int main(void){

char buf[BUFFER\_SIZE];

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

int count;

int fd1, fd2;

// open 함수를 이용하여 fd1에 파일 디스크립터 반환받음

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

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

exit(1);

}

fd2 = dup(fd1); // fd1의 파일 디스크립터를 복사하여 fd2에 저장

count = read(fd1, buf, 12);

// read 함수를 이용하여 fd1을 buf로 12byte 읽어와서 읽은 바이트 수를 count 변수에 저장

buf[count] = 0;

// 읽은 문자들을 가지고 있는 buf 배열의 count 인덱스에 해당하는 문자에 0을 추가하여 문자열의 끝을 표현

printf("fd1's printf : %s\n", buf); // 읽어온 buf를 확인하기 위해 출력

lseek(fd1, 1, SEEK\_CUR); // fd1의 파일 커서 위치를 현재 위치한 커서에서 1칸 이동

count = read(fd2, buf, 12);

// read 함수를 이용하여 fd2를 buf로 12byte 읽어와서 읽은 바이트 수를 count 변수에 저장

buf[count] = 0;

// 읽은 문자들을 가지고 있는 buf 배열의 count 인덱스에 해당하는 문자에 0을 추가하여 문자열의 끝을 표현

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

// 읽어온 buf를 확인하기 위해 출력

// 출력 결과 fd2에서 읽은 데이터는 fd1에서 읽은 데이터의 뒷 부분이 나옴

// 같은 프로세스에서 파일 디스크립터의 번호가 다를 뿐,

// dup()를 호출하면 원본이 되는 파일 디스크립터와 오프셋을 공유함을 알 수 있음

exit(0);

}

1. 실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<dup2 예제 1>

1. 소스코드

**[ssu\_dup2\_1.c]**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

int main(void){

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

int fd;

// creat로 파일을 생성하고 파일 디스크립트를 받아 fd에 저장

// 일반적으로 파일 디스크립터 0은 표준 입력, 1은 표준 출력, 2는 표준 에러로

// 그 다음인 3번부터 일반 파일에 대한 디스크립터 번호가 부여됨

if ((fd = creat(fname, 0666)) < 0) {

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

exit(1);

}

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

dup2(fd, 1); // ssu\_test.txt의 디스크립터(fd)를 표준 출력 1번으로 복사 (표준 출력 없어짐)

// 여기서부터는 표준 출력이 ssu\_test.txt로 바뀜

printf("Second printf is in this file.\n");

// 표준출력이 ssu\_test.txt로 바뀌면서 ssu\_test.txt에서 해당 문자열이 출력됨

exit(0);

}

2. 실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

<dup2 예제 2>

1. 소스코드

**[ssu\_dup2\_2.c]**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.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);

}

// 1(표준 출력)을 4로 복사하여 4도 표준 출력을 함

if (dup2(1,4) != 4) {

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

exit(1);

}

while(1){

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

// read 함수로 파일의 내용을 읽고 읽은 바이트 수 만큼 length에 저장

if (length <= 0) // 읽지 못한 경우 에러 처리

break;

write(4, buf, length);

// 표준 출력 1번을 4번으로 복사했기 때문에

// 4번 파일 디스크립터를 인자로하여 write()하면 표준 출력과 같기 때문에 파일의 내용이 출력됨

}

exit(0);

}

2. 실행결과

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명