**< 파일 내용 출력하기 - 코드 설명>**

- 헤더파일

표준 입출력, 입출력함수를 사용하기 위한 헤더파일 **<stdio.h>**

유닉스에서 사용하는 c 컴파일러 헤더파일 **<unistd.h>**

함수 조회와 설정을 할 수 있는 함수를 포함한 헤더파일 **<fcntl.h>**

문자열 변환 등 이용되지만 exit()함수를 사용하기 위해 포함한 헤더파일 **<stdlib.h>**

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

int argc : main()함수에 전달되는 데이터의 개수

char\* argv[] : main()함수에 전달되는 실제적인 데이터로 char형 포인터 배열로 구성

**int rfd, wfd, n;**

읽기 전용 파일 디스크립터 변수를 표현하기 위한 변수 rfd

쓰기 전용 파일 디스크립터 변수를 표현하기 위한 변수 wfd

글자 수를 의미하는 변수 n

**char buf[100];**

버퍼의 크기를 100으로 지정

**if (argc != 2 && argc != 3) {**

**printf("Error, write down 'file name'\n");**

**exit(1);**

**}**

main함수에 전달되는 데이터의 개수가 2개나 3개가 아닐 경우 파일을 입력 받지 못한 것과 같으므로 에러문 출력 및 현재 c프로그램 완전히 종료시킴.

**rfd = open(argv[1], O\_RDONLY);**

open() 함수를 이용하여 argv[1] 데이터 파일 열어 달라고 요청

O\_RDONLY : 파일을 읽기 전용으로 연다.

**if (rfd == -1) {**

**perror("Fail to open");**

**exit(1);**

**}**

현재 열려 있는 파일을 구분하는 정수 값인 파일 기술자는 0,표준 입력부터 시작한다.

-1일 경우에는 파일이 제대로 읽혀 지지 않는다는 의미와 같으므로 -1일 경우 에러문 출력

perror( ) : 존재하지 않는 파일을 읽기 모드로 열었을 경우 에러 메시지를 출력하는 함수

exit( ) : 현재의 c프로그램 자체를 완전히 종료 시키는 기능을 가지는 함수

- ()안에 1을 넣는 이유는 에러가 발생하여 종료할 때 대체로 1를 반환

main 함수에 전달되는 데이터의 개수를 조건문을 통하여 조건을 만족할 경우 실행

- 데이터의 개수가 2개일 경우

**if (argc == 2) {**

**while (read(rfd, buf, sizeof(buf)))**

**printf("파일 출력: %s\n", buf);**

**close(rfd);**

**}**

반복문 while() (조건)을 만족하는 ~동안

rfd파일을 버퍼의 크기만큼 버퍼에 읽어 들일 동안

read() : open함수로 열기로 한 파일의 내용을 읽어주는 함수

buf : 파일을 읽어 들일 버퍼

sizeof(buf) : 버퍼의 크기 sizeof()함수는 괄호 안 자료형의 크기를 출력하는 함수

printf(“파일 출력: %s\n”,buf)

즉, rfd파일을 버퍼의 크기만큼 버퍼에 읽어 들일 동안에, 저장된 버퍼의 내용을 출력한다.

모든 버퍼가 읽어 들여지면, close(rfd)를 통해 rfd파일을 닫아준다.

- 데이터의 개수가 3개일 경우

**if (argc == 3) {**

**wfd = open(argv[2], O\_CREAT | O\_WRONLY | O\_TRUNC, 0644);**

**if (wfd == -1) {**

**perror("Fail to open second file");**

**return 1;**

**}**

**while ((n = read(rfd, buf, sizeof(buf))) > 0) {**

**write(wfd, buf, n);**

**}**

**close(rfd);**

**close(wfd);**

**}**

**wfd = open(argv[2], O\_CREAT | O\_WRONLY | O\_TRUNC, 0644);**

open()함수를 이용하여 argv[2] 파일을 열어준다.

oflag와 숫자를 이용하여 파일의 접근 권한을 지정

O\_CREAT : 파일이 없으면 생성

O\_WRONLY : 파일을 쓰기 전용으로 연다.

O\_TRUNC : 파일을 생성할 때 이미 있는 파일이고, 쓰기 옵션으로 열었을 경우 내용을 모두 지우고 파일의 길이를 0으로 변경

0644 : 파일에 대한 소유자는 읽기,쓰기,그룹과 기타 사용자는 읽기 권한이 있음을 의미

**if (wfd == -1) {**

**perror("Fail to open second file");**

**return 1;**

**}**

조건문 if(조건) { 실행문 } : wfd 파일이 열리지 않을 경우

-1일 경우에는 파일이 제대로 읽혀 지지 않는 의미로 이것의 부정은 파일이 읽힌다는 의미기에 조건문을 통해 파일 열림의 유무를 판단한다.

perror() : 오류 상황을 표준 오류 스트림에 출력하기 위한 함수 -> 에러문 출력

return 1; : 1의 값을 반환함으로써 해당 함수 종료

**while ((n = read(rfd, buf, sizeof(buf))) > 0) {**

**write(wfd, buf, n);**

**}**

반복문 while() (조건)을 만족하는 ~동안

읽어 들인 버퍼의 수가 0보다 큰 동안에 write()함수를 통해 파일의 내용을 wfd 파일에 작성

read( ) : open()함수로 열기로 한 파일의 내용을 읽는 함수

rfd : 읽을 파일의 파일 디스크립터

buf : 파일을 읽어 들일 버퍼

sizeof(buf) : 버퍼의 크기 sizeof() 함수는 괄호 안에 자료형의 크기를 출력하는 함수

즉, rfd 파일을 버퍼의 크기만큼 버퍼에 읽어 들인다.

write() : 파일 디스크립터를 이용하여 참조한 파일에 데이터를 쓴다.

(wfd, buf, n) : buf가 가리키는 메모리에서 n으로 지정한 크기만큼 wfd 파일에 기록

**close(rfd);**

rfd 파일을 닫아준다.

**close(wfd);**

wfd 파일을 닫아준다.

**return 0;**

main 함수를 끝낸 후 벗어나기 위하여 0을 return한다.