

# 입출력 리디렉션

2020년 12월 20일 일요일 오후 8:44

- 콘솔 프로그래밍의 어려움
  - 손가락이 아픔
    - 맨날 stdin 받을 때마다 타이핑 쳐야하고 EOF 넣기도 귀찮다
  - 눈이 아픔
    - stdin 뒤에 stdout 출력하면 뒤죽박죽 나오고
    - stdout과 stderr 출력했는데 섞여서 나온다
  - 이것들을 해결해주는 것이 있다
    - 바로 '입출력 리디렉션(IO redirection)

## ■ 입출력 리디렉션(IO redirection)

- 입력이나 출력이 들어오고 나가는 방향을 다른 데로 돌려줌
  - 리디렉션은 방향을 틀어준다는 의미
- 입력 리디렉션
  - 텍스트 파일을 열어 stdin에 대신 타이핑 쳐주는 기능
- 출력 리디렉션
  - stdout에 출력되는 것을 화면에 보여주는 대신 텍스트 파일에 저장
  - 마찬가지로 stderr에 출력되는 것도 별도의 텍스트 파일에 저장해줌
- 입출력 리디렉션은 C의 기능이 아님
  - 커멘드 라인 또는 shell의 기능이다
- fopen( )하고는 다른 것
  - 리디렉션 != fopen( ) 이다!!
    - 파일을 여는 것과 리디렉션 하는 것은 다른 것

### • 사용 방법

- stdin
  - < 를 사용
- stdout
  - > 를 사용
- stderr
  - 2> 를 사용

```
> a.exe < input.txt > output.txt 2> error.txt
```

### • 장점

- 손가락이 안 아프다
  - 여태까지 다양한 거 테스트할 때 한땀한땀 콘솔창에서 타이핑한 것들 다 input.txt에 저장
  - 그리고 a < input.txt 하면 끝
- 출력과 오류 메시지가 확실히 구분됨
  - 출력과 오류 메시지를 각각 다른 파일에 저장할 수 있다
- EOF를 안 쳐도 됨
  - 기존에 키보드에서 직접 입력할 때는 EOF + 엔터를 쳐줘야 했음
    - 프로그램이 키보드 입력의 끝을 알 수 없으니 사용자가 직접 넣어줌
    - 그리고 EOF 뒤에 새 줄 문자가 따라와서 다시 함수를 호출하면 그 뉴 라인이 읽혀질 수 있다
  - 파일로 입력 리디렉션을 대신할 경우 자동으로 파일을 다 읽는 순간 EOF가 입력 됨

### • 사용 코드

- 빈 input.txt를 읽을 때

```
/* 코드 생략 */
```

```
if (fgets(filename, FILE_LENGTH, stdin) == NULL) {  
    fprintf(stderr, "no input\n");  
    return;  
}
```

```
/* 코드 생략 */
```

```
> a.exe < input.txt > output.txt 2> error.txt  
>
```



input.txt



output.txt

no input

error.txt

- input.txt의 숫자를 출력하기

```
int score;  
  
while (scanf("%d", &score) == 1) {  
    printf("%d ", score);  
}
```

```
> a.exe < input.txt > output.txt 2> error.txt  
>
```

65 38 75 91 24  
100 27 88 75 43

input.txt

65 38 75 91 24 100 27 88 75 43

output.txt



error.txt

- \n이 scanf에서는 공백 문자로 인식되어 저장되지 않음