

1. 리눅스 리다이렉션(Linux Redirection)

리다이렉션은 표준 스트림(standard stream)의 흐름을 바꿔줍니다. 리눅스에서 리다이렉션을 쓰고 싶으면 특수 기호 `>` 또는 `<`를 사용합니다. 더 상세한 설명을 하기 전에 기반 지식을 쌓고 갑시다.

일반적으로 커맨드(command)로 실행되는 프로세스(process)는 표준 입력 스트림(standard input stream), 표준 출력 스트림(standard output stream)과 오류 출력 스트림(standard error stream)을 가지고 있습니다. 총 하나의 입력 스트림과 두개의 출력 스트림을 가지고 있는것이죠. 기본적으로 이 모든 스트림은 일반적인 문자열(plain text)로 콘솔(console)에 출력하게 되어 있습니다.

그렇다면 다시 돌아가서 리다이렉션이란 스트림의 흐름을 바꿔주는 것이라고 앞서 말씀 드렸습니다. 결론부터 말하면 실행된 프로세스의 스트림을(콘솔이 아닌) 파일로 사용하고 싶다면 리다이렉션을 사용하면 됩니다. 코드로 살펴봅시다.

```
$ ls > ls.txt
```

`ls` 명령어의 출력 결과를 `ls.txt` 파일에 저장합니다. 정확하게 말해서 `ls`의 출력 스트림을 `ls.txt`이라는 파일로 쓰는것이죠. 따라서 `ls` 명령어의 결과는 콘솔이 아닌 `ls.txt`라는 파일에 기록됩니다. 반대의 경우도 가능합니다.

```
$ head < ls.txt
```

`head` 커맨드는 파일의 처음 부분을 설정한 라인(line)만큼 출력하는 명령어입니다. 기본적으로 10라인이 세팅 되어 있습니다. 위의 코드는 실행하면 `ls.txt`의 처음 10줄을 가져올 겁니다. 정확하게 말하면 `ls.txt`를 `head`의 입력 스트림으로 보낸 것입니다.

조금 더 응용하면 다음처럼 사용도 가능합니다.

```
$ head < ls.txt > ls2.txt
```

위의 코드는 다음과 같은 순서로 실행 됩니다.

- `ls.txt`의 내용을 `head` 명령어의 입력 스트림으로 전송
- `head` 명령어는 입력 받은 `ls.txt`의 내용에서 처음 10줄을 출력
- `head` 명령어의 출력 스트림을 `ls2.txt` 파일에 연결
- `head` 명령어의 출력 스트림은(콘솔이 아닌) `ls2.txt`에 연결 되어 있으므로 출력 결과를 `ls2.txt`에 저장

2. 리눅스 파이프(Linux Pipe)

리다이렉션은 프로세스의 입력이나 출력을 파일로 사용하는 것이라면 파이프(pipe)는 프로세스간 사용하는 것입니다. 리눅스에서는 특수 기호로 `|`를 사용합니다. 일반적으로 "**A | B**" 처럼 사용하는데 `|`를 기준으로 A에 있는 커맨드의 표준 출력을 B에 있는 커맨드의 표준 입력으로 사용합니다.


프로세스의 출력 스트림을 다른 프로세스의 입력 스트림으로 사용할 때 리다이렉션 기호를 쓰면 오류가 납니다. 아마도 일치하는 파일이 없거나 디렉토리가 없다고 오류 메시지가 나올겁니다. 이런 경우엔 파이프 기호를 씁시다.

다음과 같은 코드를 보면 자연스럽게 파이프를 사용하고 있었다는 것을 알 수 있습니다.

```
$ ls | grep ls.txt
```


`ls` 명령어의 출력 스트림을 `grep` 커맨드의 입력 스트림으로 보내고 `grep` 커맨드의 인자값으로 `ls.txt`를 준 것입니다. 현재 디렉토리에 `ls.txt` 파일이 있으면 결과를 콘솔에 출력할 겁니다. `grep` 명령어는 원래 다음처럼 사용합니다.

```
$ grep [-option] [pattern] [file or directory name]
```



Professional .NET Office API

.NET applications edit, convert, render, print Excel, Word, PowerPoint, Outlook and PDF.



`grep` 키워드 이후 옵션과 패턴, 그리고 파일이나 디렉토리 이름을 써주면 됩니다. 패턴은 정규식(Regular Expression)으로 사용 가능합니다. 이러한 일반적인 `grep` 사용법 외에 `grep`을 파이프로 입력 스트림을 주는 경우엔 필터처럼 동작합니다. 파일이나 디렉토리명을 써주지 않더라도 입력 스트림으로 받은 내용과 패턴을 바탕으로 `grep` 명령어를 시행합니다.

간단히 요약하면 프로세스간의 입력, 출력을 연결하고 싶으면 파이프 기호를 이용합시다.