

[리눅스] 04. IO Redirection (입력 출력 리다이렉션)

04. IO Redirection (입력 출력 리다이렉션)

학습 목표

가. 리눅스의 IO Redirection에 대해 알아본다.

나. 표준 입출력을 통해 IO Redirection을 이해한다.

목차

04. IO Redirection (입력 출력 리다이렉션)

4-1 IO Redirection에 대해 알아보기(OUTPUT)

4-2 IO Redirection에 대해 알아보기(INPUT)

4-3 IO Redirection에 대해 알아보기(append)

4-4 SUMMARY

4-5 Quiz 및 실습

4-1 IO Redirection(입력 출력 리다이렉션)에 대해 알아보기

가. 리다이렉션이란?(Redirection)

리다이렉션은 단순히 스크립트 내에서 파일, 명령, 프로그램, 스크립트 또는 코드 블록의 출력을 캡처하여 다른 파일, 명령, 프로그램 또는 스크립트에 입력으로 보내는 것을 의미한다.

나. OUTPUT(출력)에 대해 알아보기

[예제]

```

1 data@data:~$ ls -l
2  합계  44
3  -rw-r--r--  1 data data 8980  2월  2 10:28 examples.desktop
4  drwxr-xr-x  2 data data 4096  2월  2 11:04 공개
5  drwxr-xr-x  2 data data 4096  2월  2 11:04 다운로드
6  drwxr-xr-x  2 data data 4096  2월  2 11:04 문서
7  drwxr-xr-x  2 data data 4096  2월  2 11:04 바탕화면
8  drwxr-xr-x  2 data data 4096  2월  2 11:04 비디오
9  drwxr-xr-x  2 data data 4096  2월  2 11:04 사진

```

```

10 | drwxr-xr-x 2 data data 4096 2월 2 11:04 음악
11 | drwxr-xr-x 2 data data 4096 2월 2 11:04 템플릿
12 |

```

'>'을 이용하여 ls -l의 결과를 result.txt로 보내어진다.
 결과적으로 result.txt에 저장이 된다.

```

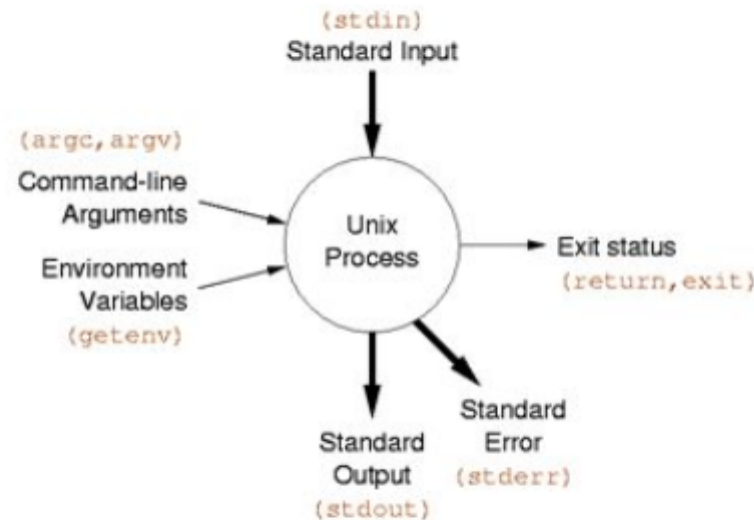
1 | data@data:~$ ls -l > result.txt
2 | data@data:~$ cat result.txt
3 | 합계 44
4 | -rw-r--r-- 1 data data 8980 2월 2 10:28 examples.desktop
5 | -rw-rw-r-- 1 data data 0 2월 4 22:02 result.txt
6 | drwxr-xr-x 2 data data 4096 2월 2 11:04 공개
7 | drwxr-xr-x 2 data data 4096 2월 2 11:04 다운로드
8 | drwxr-xr-x 2 data data 4096 2월 2 11:04 문서
9 | drwxr-xr-x 2 data data 4096 2월 2 11:04 바탕화면
10 | drwxr-xr-x 2 data data 4096 2월 2 11:04 비디오
11 | drwxr-xr-x 2 data data 4096 2월 2 11:04 사진
12 | drwxr-xr-x 2 data data 4096 2월 2 11:04 음악
13 | drwxr-xr-x 2 data data 4096 2월 2 11:04 템플릿

```

Review: UNIX Programs

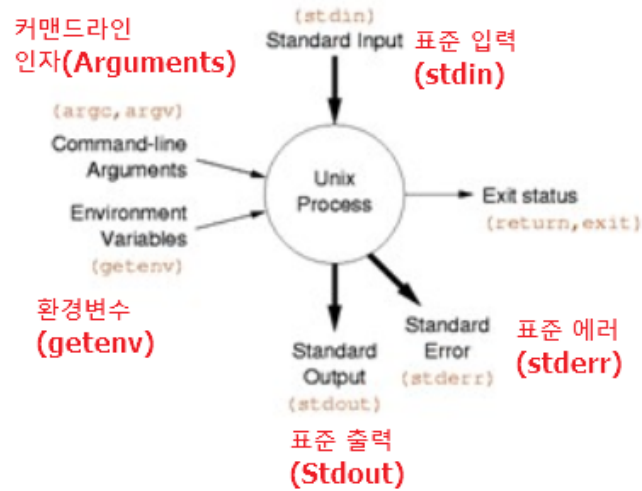
- **Means of input:**

- Program arguments [control information]
- Environment variables [state information]
- Standard input [data]



- **Means of output:**

- Return status code [control information]
- Standard out [data]
- Standard error [error messages]



[예제]

`ls -al` ### `ls`에 입력으로 들어오는 값이 `-al`이다. (`-al`은 command Line argument)

`ls -l` ### `ls -l`에 의해 나오게 되는 화면은 standard output(표준 출력) 이 된다.

`ls -l > result.txt` ### `ls -l`의 출력은 `result.txt`의 입력으로 되어 `result.txt`에 저장되게 된다.

다. Standard output and Standard Error

프로세스(중간의 원)은 2가지 결과값으로 출력하기로 함.

하나는 Standard output 와 Standard Error로 출력하기로 했다.

(예제)

`touch r1.txt`

`rm r1.txt` # 처음은 잘 삭제됨.

`rm r1.txt` # 결과 에러 발생

`rm r1.txt > result.txt`

생각외로 redirection으로 하고 있지 않다. standard error를 redirection하지 않고 있다.

자 그럼 다음과 같이 할 수 있다.

`rm r1.txt 1>result.txt` # 1은 standard output

```
rm r2.txt 2>error.log # 2은 standard Error
cat error.log
```

```
rm rename2.txt 1> result.txt 2> error.log
```

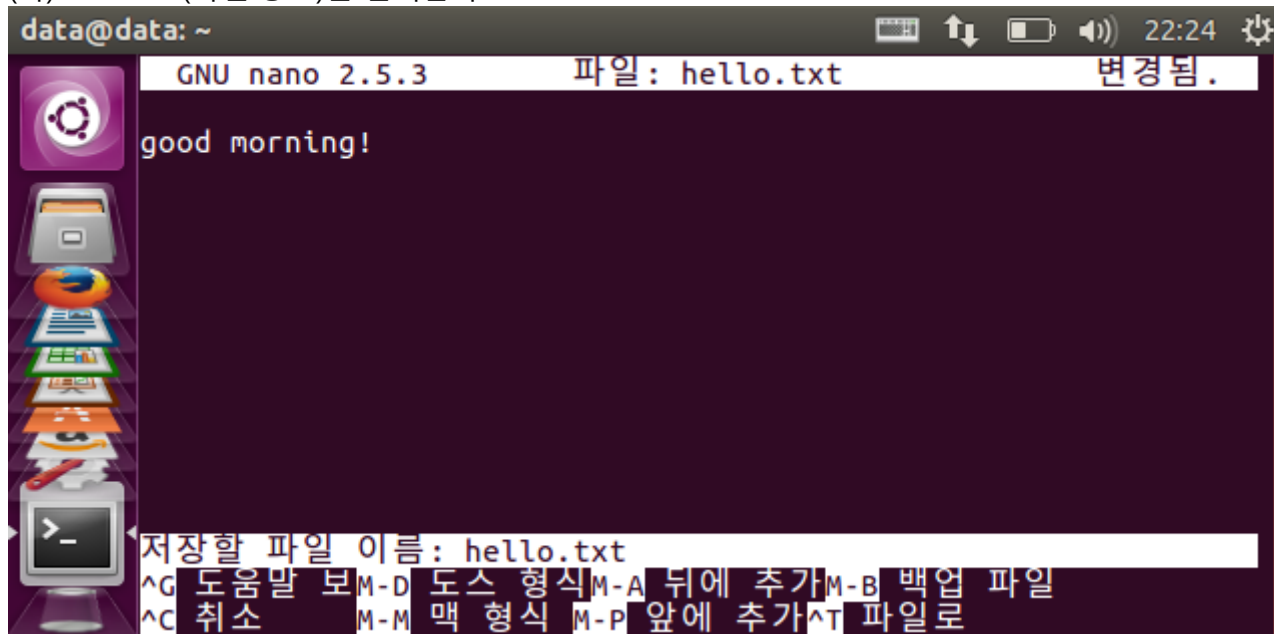
4-2 IO Redirection에 대해 알아보기(INPUT)

01 hello.txt 파일을 준비한다.

(가) data@data:~\$ nano hello.txt

(나) **good morning!**을 입력 후, **CTRL + O**(파일 저장)을 클릭한다.

(다) CTRL + X(파일 종료)을 선택한다.



확인

```
1 data@data:~$ cat hello.txt
2 good morning! cs
```

02 stdinput 실습해 보기

cat을 하여 키보드로부터 입력을 받을 수 있고, '<'을 이용하여 받을 수도 있다.

```
data@data:~$ cat
hi
hi
hello
hello
data@data:~$ cat < hello.txt
good morning!
```

03 일반 입력과 Standard Input의 실습해보기

(가) linux.txt데이터를 다운로드 받기

wget <https://www.dropbox.com/s/8xrybxjes55cbd4/linux.txt?dl=0>

(나) 아래 실행을 하여 확인해 보기

```
data@data:~$ head -n1 linux.txt
data@data:~$ head -n1 < linux.txt
data@data:~$ head -n1 < linux.txt > resultl.txt
data@data:~$ cat resultl.txt
```

4-3 IO Redirection에 대해 알아보기(append)

[예제]

```
ls -al > r1.txt
ls -al >> r1.txt
ls -al >> r1.txt
cat r1.txt
```

4-4 SUMMARY

'>'에 대해

'<'에 대해

'>>'에 대해

4-5 Quiz 및 실습

(1) rm rename2.txt 1> result.txt 2> error.log 에 대해 무엇을 하는 것인지 적어보자.

(2) '>' 과 '>>'의 차이는 무엇인지 적어보자.

(3) `ls -l`의 결과를 `l1.txt`에 저장하자. 그리고 `rm ttt.txt`의 '`1>`', '`2>`'의 결과를 각각 `l2.txt`와 `err.log`을 저장해 보자.

`cat l2.txt`의 결과와 `err.log`의 결과를 `l1.txt`에 추가해보자.