# **기본 개념**

## **표준 입력과 표준 출력**

Java에서 입력을 받을 때 System.in 객체를 사용하고, 출력할 때 System.out 객체를 사용한다.

이 입력과 출력을 표준 입력, 표준 출력이라고 부른다.

## **redirection**

표준 입력이 들어오는 방향과 표준 출력이 출력되는 방향을, 유닉스에서는 쉽게 바꿀 수 있다.

표준 입출력 방향을 바꾸는 것을 redirection 이라고 한다.

명령 > 파일명 : 명령의 실행 결과 표준 출력을 파일에 저장한다.(redirection)

ls > list.txt : ls 명령의 출력이 화면에 출력되지 않고, list.txt 파일에 저장된다. list.txt 파일이 있었다면 덮어써지고, 없었다면 새로 생성된다.

### **pipe**

레고 블록을 조립하듯이 작은 명령들을 조합하여 큰 작업을 할 수 있다는 점이 유닉스의 장점이다. 작은 명령들을 조합할 때, 한 명령의 출력을 그 다음 명령의 입력으로 연결한다. 즉 한 명령이 출력한 내용이, 그 다음 명령의 입력으로 들어간다. 이렇게 입출력을 연결하는 것은 pipe 라고 한다.

명령1 | 명령2 : 명령1의 표준 출력이 명령2의 표준 입력으로 들어간다.

### **filter**

pipㄷ 방식으로 연결해서 사용하는 것이 주 목적인 명령들을 필터라고 부른다.

# **필터**

## **wc**

word count

파일의 줄 수, 단어 수, 크기를 출력하는 명령이다.

wc 파일명

|  |
| --- |
| $ wc pom.xml  74 84 2277 pom.xml |

74줄 84단어 2277byte 를 의미

### **-l 옵션**

줄 수만 출력한다.

### **-w 옵션**

단어 수만 출력한다.

### **-c 옵션**

byte 수만 출력한다.

|  |
| --- |
| $ wc -l pom.xml  74 pom.xml  $ wc -w pom.xml  84 pom.xml  $ wc -c pom.xml  2277 pom.xml |
| $ ls | wc -l  6 |

ls 명령의 표준 출력이 wc -l 명령의 표준 입력이 된다.

ls 명령은 현재 디렉토리의 파일 목록을 출력하고,

wc -l 명령은 표준 입력에서 줄 수를 세서 출력한다.

결국 현재 디렉토리의 파일 수가 출력된다.

## **find**

하위 디렉토리들에서 특정 파일을 찾는 명령이다.

### **name**

find src -name “\*.java”

src 디렉토리 아래에서 \*.java 파일을 찾아서 그 파일의 경로명을 출력한다.

|  |
| --- |
| $ find src -name "\*.java"  src/main/java/net/skhu/controller/DepartmentController.java  src/main/java/net/skhu/controller/FormController.java  src/main/java/net/skhu/controller/StudentController.java  src/main/java/net/skhu/dto/Department.java  src/main/java/net/skhu/dto/Student.java  src/main/java/net/skhu/mapper/DepartmentMapper.java  src/main/java/net/skhu/mapper/StudentMapper.java  src/main/java/net/skhu/Mybatis1Application.java  src/main/java/net/skhu/ServletInitializer.java  src/test/java/net/skhu/Mybatis1ApplicationTests.java |

find . -name “\*.java”

현재 디렉토리 아래에서 \*.java 파일을 찾아서 그 파일의 경로명을 출력한다.

|  |
| --- |
| $ find . -name "\*.java"  ./.mvn/wrapper/MavenWrapperDownloader.java  ./src/main/java/net/skhu/controller/DepartmentController.java  ./src/main/java/net/skhu/controller/FormController.java  ./src/main/java/net/skhu/controller/StudentController.java  ./src/main/java/net/skhu/dto/Department.java  ./src/main/java/net/skhu/dto/Student.java  ./src/main/java/net/skhu/mapper/DepartmentMapper.java  ./src/main/java/net/skhu/mapper/StudentMapper.java  ./src/main/java/net/skhu/Mybatis1Application.java  ./src/main/java/net/skhu/ServletInitializer.java  ./src/test/java/net/skhu/Mybatis1ApplicationTests.java |

find . -name “\*.java” -o -name “\*.jsp” : -o 옵션은 OR 연산자에 해당한다.

현재 디렉토리 아래에서 \*.java 파일이나 \*.jsp 파일을 찾아서 그 파일의 경로명을 출력한다.

|  |
| --- |
| $ find . -name "\*.java" -o -name "\*.jsp"  ./.mvn/wrapper/MavenWrapperDownloader.java  ./src/main/java/net/skhu/controller/DepartmentController.java  ./src/main/java/net/skhu/controller/FormController.java  ./src/main/java/net/skhu/controller/StudentController.java  ./src/main/java/net/skhu/dto/Department.java  ./src/main/java/net/skhu/dto/Student.java  ./src/main/java/net/skhu/mapper/DepartmentMapper.java  ./src/main/java/net/skhu/mapper/StudentMapper.java  ./src/main/java/net/skhu/Mybatis1Application.java  ./src/main/java/net/skhu/ServletInitializer.java  ./src/main/webapp/WEB-INF/views/department/edit.jsp  ./src/main/webapp/WEB-INF/views/department/list.jsp  ./src/main/webapp/WEB-INF/views/form/edit1.jsp  ./src/main/webapp/WEB-INF/views/form/edit2.jsp  ./src/main/webapp/WEB-INF/views/form/edit3.jsp  ./src/main/webapp/WEB-INF/views/student/edit.jsp  ./src/main/webapp/WEB-INF/views/student/list.jsp  ./src/test/java/net/skhu/Mybatis1ApplicationTests.java |

|  |
| --- |
| $ find src -name "\*.java"  src/main/java/net/skhu/controller/DepartmentController.java  src/main/java/net/skhu/controller/FormController.java  src/main/java/net/skhu/controller/StudentController.java  src/main/java/net/skhu/dto/Department.java  src/main/java/net/skhu/dto/Student.java  src/main/java/net/skhu/mapper/DepartmentMapper.java  src/main/java/net/skhu/mapper/StudentMapper.java  src/main/java/net/skhu/Mybatis1Application.java  src/main/java/net/skhu/ServletInitializer.java  src/test/java/net/skhu/Mybatis1ApplicationTests.java  $ find src -name "\*.java" | wc -l  10 |

### **size**

find . -size +10M

현재 디렉토리 아래에서 파일 크기가 10M 이상인 파일들을 찾아서 그 파일의 경로명을 출력한다.

find src -name “\*.java” -size -2k

src 디렉토리 아래에서 파일 이름이 \*.java 이고 파일 크기가 2k 이하인 파일들을 찾아서 그 파일의 경로명을 출력한다.

find src -name “\*.mp4” -size +1G

src 디렉토리 아래에서 파일의 이름이 \*.mp4 이고 파일의 크기가 1G 이상인 파일들을 찾아서 그 파일의 경로명을 출력한다.

### **directory**

find . -type d : 현재 디렉토리 아래에서 디렉토리들의 경로 명을 출력한다.

find . -type f : 현재 디렉토리 아래에서 보통 파일들의 경로명을 출력한다.

## **xargs 필터**

find src -name “\*.java” | wc -l

find src -name “\*.java” | xargs wc -lc

xargs가 없으면 파이프 이전 명령의 출력은 데이터이고, 그 출력된 데이터가 다음 명령의 입력이 된다.

xargs가 있으면 파이프 이전 명령의 출력은 파일명 목록이고, 그 파일들이 다음 명령의 대상 파일들이 된다.

find src -name “\*.java” | wc -l

find 명령의 표준 출력이 wc 명령의 표준 입력에 연결된다.

위 find 명령은 src 디렉토리 아래의 \*.java 파일들의 경로명을 출력하고 위 wc 명령은 표준 입력에서 입력된 내용의 줄 수를 세서 출력한다.

결국 src 폴더 아래의 \*.java 파일의 수를 출력한다.

|  |
| --- |
| $ find src -name "\*.java" | xargs wc -lc  60 1877 src/main/java/net/skhu/controller/DepartmentController.java  62 2068 src/main/java/net/skhu/controller/FormController.java  67 2197 src/main/java/net/skhu/controller/StudentController.java  26 526 src/main/java/net/skhu/dto/Department.java  72 1460 src/main/java/net/skhu/dto/Student.java  34 1071 src/main/java/net/skhu/mapper/DepartmentMapper.java  46 1645 src/main/java/net/skhu/mapper/StudentMapper.java  13 305 src/main/java/net/skhu/Mybatis1Application.java  13 400 src/main/java/net/skhu/ServletInitializer.java  16 327 src/test/java/net/skhu/Mybatis1ApplicationTests.java  409 11876 total |

find src – name “\*.java” | xargs wc -lc 위 find 명령은 src 디렉토리 아래의 \*.java 파일들의 경로명을 출력한다.

xargs wc -l -c 명령은 표준 입력으로부터 파일 경로명을 입력 받아서 그 파일의 줄 수와 크기를 출력한다.

|  |
| --- |
| $ find . -name "\*.class"  ./target/classes/net/skhu/controller/DepartmentController.class  ./target/classes/net/skhu/controller/FormController.class  ./target/classes/net/skhu/controller/StudentController.class  ./target/classes/net/skhu/dto/Department.class  ./target/classes/net/skhu/dto/Student.class  ./target/classes/net/skhu/mapper/DepartmentMapper.class  ./target/classes/net/skhu/mapper/StudentMapper.class  ./target/classes/net/skhu/Mybatis1Application.class  ./target/classes/net/skhu/ServletInitializer.class  ./target/test-classes/net/skhu/Mybatis1ApplicationTests.class  $ find . -name "\*.class" | xargs rm -v  removed './target/classes/net/skhu/controller/DepartmentController.class'  removed './target/classes/net/skhu/controller/FormController.class'  removed './target/classes/net/skhu/controller/StudentController.class'  removed './target/classes/net/skhu/dto/Department.class'  removed './target/classes/net/skhu/dto/Student.class'  removed './target/classes/net/skhu/mapper/DepartmentMapper.class'  removed './target/classes/net/skhu/mapper/StudentMapper.class'  removed './target/classes/net/skhu/Mybatis1Application.class'  removed './target/classes/net/skhu/ServletInitializer.class'  removed './target/test-classes/net/skhu/Mybatis1ApplicationTests.class' |

$find . -name “\*.class” | xargs rm -v

위 find 명령은 src 디렉토리 아래의 \*.java 파일들의 경로명을 출력한다.

xargs rm -v 명령은 표준 입력으로부터 파일 경로명을 입력 받아서 그 파일들을 삭제한다.

-v 옵션은 작업 내용을 자세히 출력하는 옵션이다. (verbose)

## **head 필터**

입력에서 선두 몇 줄만 출력한다.

head -n 10 test1.java

test1.java 파일에서 선두 10 줄만 출력한다.

ls -lS

현재 디렉토리의 파일 목록을, 크기 내림차순으로 정렬하여 출력한다.

ls -lS | head -n 5

현재 디렉토리의 파일 목록을, 크기 내림차순으로 정렬하여 선두 5줄만 출력한다.

|  |
| --- |
| $  cd /bin  $ ls -lS | head -n 5  total 94732  -rwxr-xr-x 1 SeungHyeon 197609 3562621 7월 28 09:20 msys-2.0.dll\*  -rwxr-xr-x 1 SeungHyeon 197609 3395427 7월 28 09:20 msys-perl5\_32.dll\*  -rwxr-xr-x 1 SeungHyeon 197609 2963968 7월 28 09:20 ex.exe\*  -rwxr-xr-x 1 SeungHyeon 197609 2963968 7월 28 09:20 rview.exe\* |

total 94732 줄을 포함하여 5줄이 출력되었다.

## **less**

표준 입력의 내용을 한 페이지씩 끊어서 출력해 주는 필터이다.

ls -l /bin 을 실행하면 한 화면을 훨씬 넘는 많은 출력을 나타낸다.

ls -l /bin | less 를 실행하면, ls -l /bin 출력을 less 필터가 한 화면씩 끊어서 출력한다.

### 단축키

|  |  |
| --- | --- |
| 단축키 | 설명 |
| h | 도움말 |
| q | 실행종료 |
| e | 한 줄 아래로 |
| y | 한 줄 위로 |
| f | 한 화면 아래로 |
| b | 한 화면 위로 |
| /패턴 | 정규식 패턴에 일치하는 문자열 찾기 (앞 방향으로 찾기). 패턴 입력후 엔터키를 눌러야 한다. |
| ?패턴 | 정규식 패턴에 일치하는 문자열 찾기 (뒤 방향으로 찾기). 패턴 입력후 엔터키를 눌러야 한다. |
| / | 앞 방향 찾기 계속 (/ 문자를 입력한 후 엔터키를 눌러야 한다) |
| ? | 뒷 방향 찾기 계쏙 (? 문자를 입력한 후 엔터키를 눌러야 한다) |

less 화면에서 위 단축키를 사용해서 탐색할 수 있다.

## **grep**

파일들에서 문자열 패턴을 찾는 명령으로 사용할 수도 있고,

표준 입력에서 문자열 패턴을 포함하는 줄만 출력하는 필터로 사용할 수도 있다.

grep main \*.java

현재 폴더의 \*.java 파일들에서 main 문자열이 들어있는 파일명을 출력한다.

|  |
| --- |
| $ grep main \*.java  DualPivotPartition1.java: public static void main(String[] args) {  DualPivotPartition2.java: public static void main(String[] args) {  DualPivotPartition3.java: public static void main(String[] args) {  DualPivotPartition4.java: public static void main(String[] args) {  partition1.java: public static void main(String[] args) {  partition2.java: public static void main(String[] args) {  QuickSortInt1.java: public static void main(String[] args) {  QuickSortInt2.java: public static void main(String[] args) {  QuickSortLoop1.java: public static void main(String[] args) {  QuickSortLoop2.java: public static void main(String[] args) {  QuickSortString1.java: public static void main(String[] args) {  QuikSortPerson1.java: public static void main(String[] args) { |

main 문자열이 포함된 파일명과 그 내용이 출력된다.

### **정규식 패턴으로 찾기**

grep “[Ss]ort” \*.java

현재 폴더의 \*.java 파일들에서 [Ss]ort 패턴의 문자열이 들어있는 파일명을 출력한다.

### **recursive**

grep -r --include “\*.java” main

현재 디렉토리 아래의 모든 디렉토리의 \*.java 파일들에서 main 문자열이 들어이는 파일명을 출력한다.

### **필터**

ls -l /bin | grep zip

ls -l /bin 명령은 /bin 폴더의 파일 목록을 출력한다.

/bin 폴더 아래에 유닉스 명령 실행 파일들이 들어있다.

ls -l /bin 명령의 출력 중에서 zip 문자열을 포함하는 줄만 출력된다.

# **압축 명령**

## **gzip**

gzip은 파일을 한 개씩 따로 압축하는 명령이다.

zip 압축 알고리즘은 완전히 무료가 아니기 때문에, Linux는 GNU 프로젝트에서 만든 압축 알고리즘 gzip을 사용한다.

### **압축하기**

gzip 파일명 : 파일을 압축하여 파일명.gz 파일을 생성하고 원본파일을 삭제한다.

gzip list.txt : list.txt.gzip 압축 파일이 생성되고, list.txt 파일은 삭제된다.

gzip a.txt b.txt : a.txt.gz b.txt.gz 파일이 생성되고, a.txt b.txt 파일은 삭제된다.

gzip \*.txt : \*.txt 패턴과 일치하는 파일들이 각각에 대해 파일명.txt.gz 파일이 생성되고 원본 파일은 삭제된다.

### **gzip 명령의 제약**

gzip은 파일을 한 개씩 따로 압축 파일을 만든다.

여러 파일을 묶어서 압축파일을 만들 수 없다.

디렉토리를 압축할 수 없다.

### **압축풀기**

gunzip 파일명.gz : 파일명.gz 파일의 압축을 풀어서 새 파일을 만들고, 파일명.gz 은 삭제한다.

gunzip list.txt.gz : 압축이 풀린 list.txt 파일이 생성되고 list.txt.gz 파일은 삭제된다.

gunzip \*.gz : \*.gz 패턴과 일치하는 파일들이 각각에 대해 파일명으로 생성되고 원본 파일은 삭제된다.

## **tar**

tar는 디렉토리나 여러 파일들을 묶어서 압축하는 명령이다.

gzip명령과 다르게 원본파일이 유지된다.

tar 명령어의 옵션

|  |  |
| --- | --- |
| 옵션 | 의미 |
| -c (create) | 파일을 tar로 묶음 |
| -p | 파일 권한을 저장 |
| -v (view) | 묵거나 파일을 풀 때 과정을 화면으로 출력 |
| -f (file) | 파일 이름을 지정 |
| -C | 경로지정 |
| -x (extract) | tar 압축 풀기 |
| -z (zip) | zip으로 압축 |

### **압축하기**

tar -cvzf 압축파일명.tgz 디렉토리

디렉토리를 압축하여 압축파일을 생성한다.

위 방법으로 생성된 압축파일명의 확장자는 .tgz 이거나 .tar.gz 인 것이 관례이다.

tar -cvzf 압축파일명.tgz 파일 1 파일2 파일3

여러 파일을 압축하여 압축파일 한 개를 생성한다.

tar -cvzf hw.tgz hw

hw 디렉토리를 압축하여 hw.tgz 파일을 생성한다.

tar -cvzf src.tgz \*.java

\*.java 패턴과 일치하는 파일들을 압축하여 src.tgz 파일을 생성한다.

### **압축파일 내용 보기**

tar -tvf 압축파일명.tgz

### **압축파일 풀기**

tar -xvf 압축파일명.tgz

### **특정 파일들만 압축하기**

|  |
| --- |
| $ find src -name "\*.java" | xargs tar -cvzf src.tgz  src/main/java/net/skhu/controller/DepartmentController.java  src/main/java/net/skhu/controller/FormController.java  src/main/java/net/skhu/controller/StudentController.java  src/main/java/net/skhu/dto/Department.java  src/main/java/net/skhu/dto/Student.java  src/main/java/net/skhu/mapper/DepartmentMapper.java  src/main/java/net/skhu/mapper/StudentMapper.java  src/main/java/net/skhu/Mybatis1Application.java  src/main/java/net/skhu/ServletInitializer.java  src/test/java/net/skhu/Mybatis1ApplicationTests.java |

위 find 명령은 src 디렉토리 아래의 \*.java 파일들의 경로명을 출력한다.

xargs tar -cvzf src.tgz 명령은 표준 입력으로부터 파일 경로명들을 입력받아서 그 파일들을 gzip 압축하여 src.tgz 파일을 생성한다.